

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2004-527041

(P2004-527041A)

(43) 公表日 平成16年9月2日(2004.9.2)

(51) Int.Cl.⁷

G06F 13/00

G06F 17/21

F I

G06F 13/00

G06F 17/21

G06F 17/21

550L

501T

596Z

テーマコード (参考)

5B009

審査請求 有 予備審査請求 有 (全 156 頁)

(21) 出願番号 特願2002-575832 (P2002-575832)
 (86) (22) 出願日 平成14年3月25日 (2002.3.25)
 (85) 翻訳文提出日 平成15年9月22日 (2003.9.22)
 (86) 国際出願番号 PCT/US2002/009268
 (87) 国際公開番号 W02002/077855
 (87) 国際公開日 平成14年10月3日 (2002.10.3)
 (31) 優先権主張番号 60/278, 137
 (32) 優先日 平成13年3月23日 (2001.3.23)
 (33) 優先権主張国 米国 (US)

(71) 出願人 503346197
 アリザン コーポレイション
 アメリカ合衆国 ジョージア 30328
 , アトランタ, グレンリッジ ドライ
 ブ 5775
 (74) 代理人 100078282
 弁理士 山本 秀策
 (74) 代理人 100062409
 弁理士 安村 高明
 (74) 代理人 100113413
 弁理士 森下 夏樹

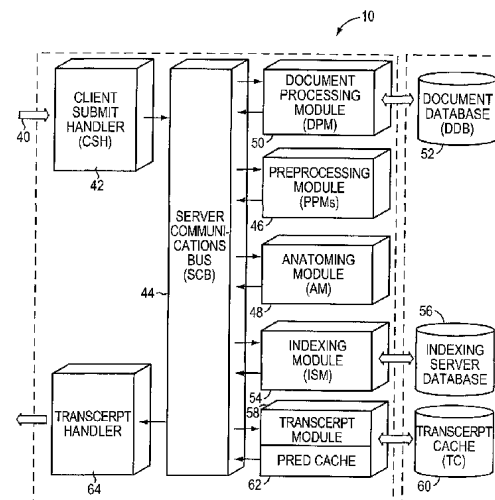
最終頁に続く

(54) 【発明の名称】 携帯用計算デバイスへの無線通信媒体を介するコンテンツ送達のためのシステムおよび方法

(57) 【要約】

共通のマークアップ言語を用いてドキュメントオブジェクトモデル内の電子ドキュメントを解析、統括および格納するシステム。ドキュメントオブジェクトモデルは、サーバ(10、14、50、52)に格納され、コンテンツは、ユニバーサルコンテンツシステムフォーマット(162)で無線デバイス(12)に送達される。ドキュメントオブジェクトモデル、および、無線デバイス上に、対応するリーダーを有するユニバーサルコンテンツストリームを利用することによって、電子ドキュメントは、基本的なコンテンツ、および、送信に必要なフォーマット情報のみに低減される。

【選択図】 図3



【特許請求の範囲】

【請求項 1】

携帯用計算デバイスにコンテンツを送達する方法であって、
該携帯用計算デバイスから遠隔のドキュメントに関連するコンテンツのために、該携帯用計算デバイスからの要求を受信するステップと、
該要求されたコンテンツに関連するドキュメントの 1 つ以上の部分を識別するステップと、
ドキュメントオブジェクトモデル (DOM) 格納装置に格納されたドキュメントの要素からの、該識別された 1 つ以上の部分を再構成するステップと、
該携帯用計算デバイスに表示するために、該再構成されて識別された 1 つ以上の部分を該携帯用デバイスに伝送するステップと
を包含する、方法。 10

【請求項 2】

前記ドキュメントの 1 つ以上の部分を識別するステップは、該ドキュメントの 1 つ以上の部分を識別するために、ルックアップまたはナビゲーションツリーを利用するステップを包含する、請求項 1 に記載の方法。

【請求項 3】

前記ドキュメントの 1 つ以上の部分を識別するステップは、前記要求されたコンテンツに最も関連するドキュメントの部分を決定する 1 つ以上の確率アルゴリズムを利用するステップを包含する、請求項 1 に記載の方法。 20

【請求項 4】

前記識別された 1 つ以上の部分を再構成するステップは、前記携帯用計算デバイスに適合する所定のフォーマットにある該識別された 1 つ以上の部分を再構成するステップを包含する、請求項 1 に記載の方法。

【請求項 5】

遠隔の携帯用計算デバイスにあるドキュメントコンテンツを発行する方法であって、
発行されるべきドキュメントを受信するステップであって、該ドキュメントは、視覚的なドキュメント表示およびドキュメントコンテンツを含む、ステップと、
該視覚的ドキュメント表示と該ドキュメントコンテンツとを分離するステップと、
該ドキュメントコンテンツをインデックス付けするステップと、 30
ナビゲーションツリーマッピングされるナビゲーションリンクを構築するステップと、
ドキュメントオブジェクトモデル構造として該ドキュメントコンテンツを格納するステップと
を包含する、方法。

【請求項 6】

前記ドキュメント用のメタデータヘッダを生成するステップをさらに包含する、請求項 5 に記載の方法。

【請求項 7】

携帯用計算デバイスにドキュメントコンテンツを送達する無線デバイスのための方法であって、
該ドキュメントコンテンツのためのナビゲーションリンクを設立するドキュメントの視覚的表示に基づいて解析的に処理することによって、存在するナビゲーション要素、または、何も存在しない場所を識別することにより、構造化または非構造化ドキュメントのコンテンツを解析するステップと、
該ドキュメントにあるコンテンツのために該携帯用計算デバイスからの要求を受信するステップと、
該要求されたコンテンツを発見して、伝送するために該ナビゲーションリンクを用いるステップと、
該携帯用計算デバイスで該コンテンツを受信して、該デバイスで該コンテンツを再構成するステップと 40
50

を包含する、方法。

【請求項 8】

前記解析中にコンテンツテーブルを生成するステップであって、前記携帯用計算デバイスは、前記コンテンツを要求するために前記コンテンツテーブルを利用する、ステップをさらに包含する、請求項 7 に記載の方法。

【請求項 9】

ドキュメントオブジェクトモデルにある前記コンテンツを格納して、該ドキュメントオブジェクトモデルから前記要求されたコンテンツを得るステップを包含する、請求項 7 に記載の方法。

【請求項 10】

ユニバーサルコンテンツストリームにある該コンテンツを前記携帯用計算デバイスに伝送するステップを包含する、請求項 7 に記載の方法。

【請求項 11】

前記ドキュメントを解析するステップは、該ドキュメントをアナトミゼーション (anatomizing) するステップを包含する、請求項 7 に記載の方法。

【請求項 12】

前記ドキュメントにおけるコンテンツの抜粋を前記携帯用計算デバイスに伝送するステップを包含する、請求項 7 に記載の方法。

【請求項 13】

前記コンテンツの要求に関連した該コンテンツの抜粋、および、以前に送られた該コンテンツの抜粋の予測キャッシングするステップを包含する、請求項 7 に記載の方法。

【請求項 14】

携帯用計算デバイスへの無線ドキュメントコンテンツ送達のためのシステムであって、通信バスであって、該通信バスに接続された入力および出力を有する、通信バスと、該バスおよびドキュメントデータベースに接続されたドキュメント処理モジュールと、該バスに接続された前処理モジュールと、該バスに接続されたアナトミゼーションモジュールと、該バスおよびインデックス付けデータベースに接続されたインデックス付けモジュールと、

該バスおよびトランスセラプトキャッシュに接続されたトランスセラプトモジュールとを含む、システム。

【請求項 15】

前記トランスセラプトモジュールおよび前記バスに接続された予測キャッシュを含む、請求項 14 に記載のシステム。

【発明の詳細な説明】

【背景技術】

【0001】

(関連出願の相互参照)

本出願は、米国特許法第 119 条により、2001 年 3 月 23 日に提出され、「SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE」と称された米国特許出願第 60/278,137 号の優先権を主張する。同出願は、本明細書中で参照として援用される。

【0002】

(発明の背景)

(発明の分野)

本発明は、概して、ドキュメントの無線コンテンツ送達に関し、より詳細には、無線デバイス上で再構築され、かつ、読み出されることによって要求されるドキュメントコンテンツを統括することに関する。

【0003】

10

20

30

40

50

(従来技術の説明)

コンピュータの使用が急速に広まったために、電子ドキュメントが比較的容易にコンピュータ間で格納かつ通信される。フルデスクトップシステムおよびラップトップシステムは、インターネット等の様々な通信媒体を介してこれらのドキュメントを容易に転送する。

【0004】

しかし、無線通信媒体によってこれらのドキュメントを利用し、移動無線デバイスを利用することも望まれる。これは、全体として一連の異なる問題を提示する。なぜなら、利用可能な帯域幅は、4 ~ 48 kbps のオーダーであり得、短期間では、56 kbps よりも増えるようには期待されない。

【0005】

通常、無線デバイスは、バッテリーの寿命を最大化し、同時に重量およびデバイスの物理的なサイズを最小化するように試みる。通常、そのデバイスはまた、遅いプロセッサおよび最小量のメモリを含む。同時に、デバイスは、異なる、一般的に制限された、オペレーティングシステム、スクリーン解像度および制限されたグラフィックレンダリング能力を有する。広く多様なフォーマットを有する大きなドキュメントは、従って、これらのデバイスをそれらの標準フォーマットで用いるためにはかなり不適切または不可能である。

【発明の開示】

【発明が解決しようとする課題】

【0006】

このように、無線デバイスがドキュメントのサイズまたはドキュメントに関係なく電子ドキュメントへのアクセスを有することを可能とするシステムを提供することが望まれる。

【課題を解決するための手段】

【0007】

(本発明の要旨)

本発明は、共通のマークアップ言語を用いてドキュメントオブジェクトモデルの電子ドキュメントを解析、統括および格納するシステムを提供する。ドキュメントオブジェクトモデルは、サーバに格納され、コンテンツは、ユニバーサルコンテンツストリームフォーマットで無線デバイスに送達される。ドキュメントオブジェクトモデルおよび無線デバイスに対応するリーダを有するユニバーサルコンテンツストリームを利用することによって、電子ドキュメントは、基本的なコンテンツおよび無線伝送に必要なフォーマット情報のみに減少される。

【0008】

上記の関連する利点を達成することによって、本発明は、添付の図面で示されたフォームで具現化され得る。しかし、図面は、例として示すのみであり、本発明の範囲を逸脱することなく示された明細書の構成に変更があってもよいことは、事実として注目すべきである。

【0009】

様々な他の機能、特徴および本発明の付属の利点は、これらが添付の図面と共に説明された場合に完全に認識される。この図面において、同様な参照文字は、いくつかのビュー全体で、同じまたは類似の部分を示す。

【発明を実施するための最良の形態】

【0010】

(発明の詳細な説明)

本発明が利用され得る1つ以上の無線デバイスまたはクライアント12と通信するデータ処理システムまたはサーバ10が図1に示される。データ処理システム10は、通常、中央処理ユニット、内部マス格納装置およびランダムアクセスメモリ等の従来のコンポーネントを含む。これらは、本発明で利用される。データ処理システム10への連絡、および、データ処理システム10からの連絡は、通常、従来の態様では、ファイアウォール/プロキシサーバ12として周知の安全なソフトウェアまたはハードウェア処理システム/ユニットを介して往来し、本発明が開発されるネットワークへの認証され、かつ、有効なア

10

20

30

40

50

クセスを保証する。データ処理システムまたはサーバ10で実行する本発明は、通常、インターネット16を介して上りデータを受信し、無線デバイスまたはクライアントに常駐する本発明の一部へと下りデータを伝送する。本発明の現在の実施形態は、データ処理システム10として、DELL、COMPAQ、SUNまたは他のWindows(R)オペレーティングシステム リリース NT 4 / 2000 / XP以上、または、Sun Solaris オペレーティングシステム リリース 8以上で動作するサーバハードウェアの他の認識されている製造業者を利用し得る。

【0011】

図2は、データ処理システムまたはサーバ10、および、無線デバイスまたはクライアント12をより詳細な概略を示す。サーバ10は、中央処理ユニット(CPU)20、ランダムアクセスメモリ(RAM)22および内部マス格納デバイス24を含み、全ては、バス26で周知のデータチャネルによって相互接続されている。データ処理システム10は、インターネット16を介して無線デバイスクライアント12と通信する。無線デバイスクライアント12処理システムは、小さなフットプリントデバイス(例えば、Motorola Dragon Ball CPUまたはIntel Strong ARM CPU)として設計された中央処理ユニット(CPU)28を含む。この中央処理ユニット(CPU)28は、小さなフットプリント無線デバイスオペレーティングシステム(例えば、Palm OSまたはPocket PC OS)を格納する読み出し専用メモリ(RAM)30と、内部フラッシュラムメモリ格納装置32と接続され、無線ラジオ/モデム34を利用してインターネット16と通信する。これら全ては、データバス36を用いて共に接続されている。

【0012】

データ処理システム10は、好ましくは、C++およびJAVA(R)等の典型的なプログラミング言語を利用するオペレーティングシステムによって提供された機能を用いて実行するように書かれたデータ命令を利用可能なオペレーティングシステムを伴うことによって本発明の本実施形態の機能を実行かつ行うように構成される。

【0013】

図3は、本発明に関して、本発明のサーバ側処理モジュールに対する処理フローのブロックダイアグラムを示す。定義によるモジュールは、オプションまたは必須の個々のデータ処理である。これらは、他のモジュールと結合され、本発明のデータ処理システム10を形成する。構造化または非構造化電子ドキュメント、あるいは、データストリーム40は、無線または有線クライアント側12から提出され、バイナリデータとしてネットワークからサーバ10に入る。クライアント提出ハンドラ42は、クライアントリクエストを受信し、これが有効なリクエストであることを検証し、サーバ通信バス44を介してクライアントリクエストに基づいて適切なルーティングコマンドによって受信されたデータを通過させる。クライアント提出ハンドラ42は、ルーティングコマンドを読み出し、ルーティングコマンドによって示された必要な順序で、必要な処理モジュールに受信されたデータを送信する。第1のルーティングコマンドが前処理モジュール(PPM)46に対するリクエストである場合、受信データは、ドキュメントまたはデータ最適化のために通過される。ドキュメントタイプがイメージを描くテキストの内容を含み、絵を表すバイナリイメージ、または、線または円の描画要素を用いて図を描く走査要素を含む走査された要素である場合、前処理モジュール46の主な目的は、本発明による利用のためのドキュメントの質を改善することである。

【0014】

イメージを描くテキストのコンテンツを含むスキャンされたドキュメントに対して、視覚的文字認識(OCR)として公知の技術は、本発明によって処理用にテキストの文字を抽出する目的に用いられる。任意のOCRスタイルソフトウェアパッケージ(例えば、ScanSoft CorporationによるOmniPage)がこの目的に対して本発明に適用され得る。絵を示すイメージに対して、本発明は、イメージの質(例えば、シャープにする、ぼかす、ガウスぼかし、モーションぼかし、輝度、明暗、ガンマ修正、変

10

20

30

40

50

換、色相および彩度、ヒストグラム平均化、エンボス加工、モザイク、メジアン、アベレージ）を改善するための周知のイメージアルゴリズムを適用してもよい。線または円描画要素を用いて絵を描くスキャンされたドキュメントに対して、ラスタートゥベクトル（R2V）ソフトウェアは、描画要素を抽出し、かつ、スキャンされた図からの情報を位置付ける目的のために用いられ、本発明の無線/有線クライアントにあるベクトル描画データとして提出されたイメージを再構築する。任意のR2Vスタイルのソフトウェアパッケージ（例えば、SoftTelecからのVP HybridCAD）がこの目的のために本発明によって利用され得る。

【0015】

サーバ通信バス44は、次いで、アナトマイゼーション（Anatomization）モジュール（AM）48上にデータ40をルーティングする。このアナトマイゼーションモジュール（AM）48は、ドキュメントのアナトミー（anatomy）を構築する責任を負い、図17に記載されたドキュメントオブジェクトモデル（DOM）を形成し（populate）、従属するプロセスがSCB44を用いて、ドキュメント処理モジュール（CPM）50にデータをルーティングする。このドキュメント処理モジュール（CPM）50では、図13に記載されたエンジンは、ディスティラー（distiller）モジュールを用いて構造化または非構造化ドキュメントをロードする。SCB44のルーティングコマンドがDPM50によってパース（parse）されるドキュメントデータを維持する（保存する）コマンドを含む場合、ドキュメントデータは、ドキュメントデータベース（DDb）52に保存される。ドキュメントデータベース52は、データベース（例えば、Microsoft CorporationからのMicrosoft SQLサーバ）を共通して参照する通常のソフトウェアパッケージに対するコレクタである。DPM50によってドキュメントの処理が首尾良く完了すると、SCB44は、アナトマイゼーションモジュール48に逆にコマンドを発して、ドキュメントオブジェクトモデルは、ドキュメントのコンテンツおよびアナトミーによって更新される。インデックス付けリクエストがSCB44のコマンドスタックの一部である場合、インデックス付けモジュール（ISM）54が呼び出され、ドキュメントのアナトマイゼーションモジュール48から利用可能なテキストコンテンツは、周知のソフトウェア製品（例えば、MicrosoftからのIndexing Server）を用いてインデックス付けされる。ドキュメントオブジェクトモジュールに関連する参照による結果は、インデックス付けサーバデータベース56に格納される。

【0016】

ISM54によって完了すると、SCB44コマンドスタックは、トランスセラプト（Transcript）モジュール（TM）58（Transcript=transmit（伝送する） excerpt（抜粋する））を呼び出し、コンテンツの最初の抜粋は、アナトマイゼーションモジュール48を呼び出して、ドキュメントオブジェクトモジュールで利用可能なコンテンツに基づいてドキュメントのコンテンツまたは要旨の最初のテーブルを要求することによって伝送のために準備される。AM48によるトランスセラプトをリターンすると、最初のトランスセラプトのコピーは、繰り返してアクセスする目的のためにトランスセラプトキャッシュ（TC）60に置かれる。SCB44が予測キャッシングモジュール62のリクエストを含む場合、AM48は、ドキュメントからの関連するコンテンツデータの将来のクライアントリクエストを予測するためにクライアントセッション期間繰り返して呼び出される。予測キャッシングモジュール62は、次いで、クライアントセッション期間に、トランスセラプトモジュール58からのトランスセラプト結果によってトランスセラプトキャッシュ60を形成する（populate）。初期のトランスセラプトは、次いで、TM58から完了すると、SCB44を介してトランスセラプトハンドラ-64にリターンされる。トランスセラプトハンドラ-64では、リターンデータ用の正しいクライアントデバイス12が識別され、トランスセラプトは、ネットワーク16に送られ、無線/有線クライアント12に戻される。

【0017】

クライアント提出ハンドラー 42 が以前に提出されたドキュメントからより多くのコンテンツのためのクライアントリクエストを受信する場合、クライアント提出ハンドラー 42 は、サーバ通信バス 44 を介するリクエストを通過させる。サーバ通信バス 44 は、順に、そのリクエストをトランスセラプトモジュール 58 に渡す。トランスセラプトモジュール 58 は、次いで、トランスセラプトキャッシュ 60 を調査して、予測キャッシングモジュール 62 によって生成されたトランスセラプトが伝達のために利用可能であるかどうかを観察する。利用可能ではない場合、アナトミゼーションモジュール 48 が呼び出され、新規のトランスセラプトは、システム 10 のためのドキュメント提出用に記述された手順を追跡するように準備される。完了すると、トランスセラプトは、トランスセラプトキャッシュ 60 に置かれ、トランスセラプト 64 は、再び呼び出され、初期のドキュメント提出として記載されたように、クライアント無線 / 有線デバイス 12 にトランスセラプトを送達する。

10

【0018】

図 4 は、本発明によって刊行するために提出されたドキュメントまたはコンテンツリクエスト 70 のパッケージングを示す。CML ベースのメタデータ記述 72 は、パッケージングのカプセルであり、適切なサーバコマンド（例えば、(publish) 刊行する (query) またはクエリする）、情報（例えば、デバイスタイプ、デバイススクリーンサイズ、自由格納スペース、推定された帯域幅転送速度等）を提供する要求デバイス 12 の無線デバイスの性質、および、ドキュメント 70 の情報（例えば、ドキュメント名、ドキュメントの日付、ドキュメントタイプまたは任意の他の捕捉情報）を含む。元のドキュメントまたはコンテンツリクエスト 70 の固有のドキュメント ID 74 は、ドキュメント提出リクエストの他の一部を作る。固有のドキュメント ID がリクエストの一部として送られる場合、その固有のドキュメント ID は、ドキュメントが既に発行され、サーバがトランスセラプトキャッシュ 60 を参照して、固有のドキュメント ID に基づくドキュメントデータベース 52 からのドキュメント取り出していることを示す。

20

【0019】

図 5 は、データ処理システムサーバ 10 への無線クライアント 12 提出要求のためのフローチャートプロセス 80 を示す。データを有する入来要求 82 が到着する。プロセス 80 の第 1 のステップは、入来要求 82 が新規の刊行要求 84 であるかどうかを構築することである。入来要求が固有のドキュメント ID 74 を有する場合、入来要求は、新規の刊行要求ではなく（いいえ）、ドキュメントは、既にサーバデータ処理システム 10 に常駐しており、従って、トランスセラプト処理ステップ 96 が直接行われる。元のドキュメントが提出される場合、メタデータ 72 は、前処理 86 を実行するクライアント要求が存在するかどうかを参照するように処理され、前処理モジュール 46 は、ステップ 88 で呼び出される。そうではない場合、アナトミゼーション処理 90 が実行される。アナトミゼーション処理モジュール 48 からの処理が完了すると、メタデータ 72 は、アナトマイズされたコンテンツに対してインデックス付け 92 を実行するクライアント要求が存在するかどうかを参照するように処理されう。これがトゥルー (true) である場合、インデックス処理モジュール 54 が呼び出され、コンテンツは、ステップ 94 でインデックス付けされる。ステップ 92 および / またはステップ 94 が完了すると、トランスセラプトモジュール 58 は、初期のコンテンツの結果を要求している無線クライアント 12 に戻して送達する目的のためにステップ 96 で呼び出される。

30

40

【0020】

図 6 は、サーバ通信バス 44 (SCB) のサーバ側のデータ処理システム 10 のデータフローを示すフローチャート 100 である。入来要求および / またはデータは、SCB による受信され 102、SCB は、入来要求が SCB 44 にとって有効な要求 104 であるかどうかを決定するように見る。要求が有効である場合（はい）、コマンドおよび関連データは、SCB によって取り出される 106。関連データは、元のドキュメントまたは元のドキュメントに固有の ID である。SCB コマンドが未解決 (pending) であるので、データは、適切な処理モジュールにルーティングされる 108。このモジュールにお

50

いて、S C B コマンドの任意の 1 つまたは組み合わせは、システムモジュール (ドキュメント処理モジュール 50、アナトミゼーションモジュール 48、前処理モジュール 46、インデックス付けモジュール、トランスセラプトモジュール 58) のうちの任意を含むサーバ側のデータ処理システム 10 によって実行される。それぞれのデータ処理システムモジュール 46、48、50、54、58 によって処理するデータに対して S C B 44 が同期リクエスト要求と非同期リクエスト要求との両方を処理するように設計されるので、S C B 44 がステップ 110 で処理されるさらなるコマンドおよびデータが存在するかどうかをチェックする場合、要求は、完了してもよいし、しなくてもよい。より多くのコマンドまたはデータが存在する場合 (はい)、コマンドスタックが空になるまで、処理は、ステップ 106 で開始し続ける。それ以上のデータまたはコマンドがステップ 110 で止まっている場合、S C B 44 は、ステップ 112 のサーバ側データ処理モジュールから要求している無線 / 有線クライアント 12 への適切な応答および結果のデータをリターンする。有効なコマンドがステップ 104 で S C B によって検出されない場合 (いいえ)、適切な失敗コードがステップ 112 で割り当てられ、結果のリターンまたは応答は、要求している無線または有線クライアント 12 に送られる。

10

【0021】

図 7 は、前処理モジュール 76 のためのサーバ側のデータ処理システムを説明するフローチャートを示す。上りデータ 122 が S C B 44 で受信される。有効ステップ 124 は、上りデータ 122 は、有効な前処理コマンドかどうかを参照するように実行される。コマンドが有効な前処理コマンドとして認識されない場合 (いいえ)、結果のエラーコードは、ステップ 126 で E _ F A I L に設定され、エラーコードは、ステップ 128 で S C B の下りデータとしてリターンされる。コマンドが有効な処理コマンド (はい) であることをステップ 124 が決定する場合、ステップ 130 は、前処理コマンドによって提出されたドキュメントタイプが有効なサポートされたドキュメントであるかどうかを参照することをチェックする。答えが否定である場合 (いいえ)、ステップ 126 は、再び結果のエラーコードを E _ F A I L に設定して、エラーコードは、ステップ 128 で S C B で下りデータとしてリターンされる。これがステップ 130 で有効なドキュメントタイプである (はい) と決定される場合、処理モジュール 46 は、ステップ 132 で呼び出される。うまくいったステップ 130 の一部として決定されたドキュメントタイプに基づいて、適切な前処理サブモジュール 134、136 または 138 が呼び出される。ステップ 134 は、(通常 O C R と呼ばれる) 光学的文字認識を実行して、走査されたドキュメントをテキストのコンテキストに変換する。本発明は、S c a n s o f t C o r p o r a t i o n による O m n i P a g e 等のソフトウェアベンダーから O C R 用に販売されているソフトウェアを用いるように設計されている。

20

30

【0022】

ステップ 136 は、ステップ 130 で決定された有効なドキュメントタイプが保存された形式 (例えば、B M P (ビットマップ) または J P E G (J o i n t P h o t o g r a p h i c E x p e r t s G r o u p (共同写真専門化グループ))) でイメージまたは写真データである場合に、ステップ 132 によって呼び出される。これらの形式では、周知の数学の公式が用いられ得、画像写真データの視覚的な局面 (例えば、シャープにする、ぼかす、ガウスぼかし、モーションぼかし、輝度、明暗、ガンマ修正、変換、色相および彩度、ヒストグラム平均化、エンボス加工、モザイク、メジアン、アベレージ) を改善し得る。ステップ 138 は、ステップ 132 で決定された有効なドキュメントタイプがスキャンされた絵であり、ラインまたは円の描画要素を用いて絵画を描く場合に、ステップ 132 によって呼び出される。ステップ 138 は、次いで、絵の要素を抽出して、走査された絵画からの情報を位置付けるために、販売されているラスターツーベクトル (R 2 V) ソフトウェアを用いて、無線 / 有線クライアント 12 のベクトル描画データとして提出された画像を再構成または再生成する。S o f t T e l e c による V P H y b r i d C A D 等の任意の R 2 V スタイルソフトウェアパッケージは、この目的のために本発明によって利用され得る。ステップ 132 によって任意のサブプロセス 134、136、13

40

50

8が完了すると、ステップ140は、プロセスが成功したと決定されるかどうかをチェックする。ステップ140が成功である場合（はい）、前処理されたデータは、ステップ128でSCBに逆にリターンされる。ステップ132からの結果が成功ではないとステップ140が決定する場合（いいえ）、結果のエラーコードがステップ126においてE__FAILに設定され、エラーコードは、ステップ128においてSCBに対する下りデータとしてリターンされる。

【0023】

図8は、アナトミゼーション処理48のためのサーバ側データ処理システム10の処理フローを示すフローチャート150である。SCBからの上りデータは、ステップ152において受信され、ステップ154へと渡される。ステップ154において、ドキュメントデータが有効であるかどうか決定される。ドキュメントデータが有効ではない場合（いいえ）、ステップ156は、E__FAILに結果のエラーコードを設定し、DOMへのポインタは、ゼロまたはNILに等しく設定され、ステップ158においてエラーコードをSCBの下りデータとしてリターンする。ドキュメントデータがステップ154において有効である（はい）と決定された場合、受信されたドキュメントデータは、ステップ160においてパッシングされ、次いで、ドキュメントオブジェクトモデル（DOM）は、ステップ162において構築される。DOMがステップ162によって構成された後、ステップ164は、構築されたDOMを維持または格納するためのコマンド要求が存在するかどうかをチェックする。ステップ164がトゥルーである（はい）場合、DOMデータは、ステップ168まで継続している前にステップ166においてドキュメントデータベース52に格納される。ステップ164がフォルス（false）である（いいえ）場合、ステップ168は、デコレータを適用するコマンドが存在するかどうかを参照することをチェックする。デコレータは、ステップ162によって構築されたドキュメントオブジェクトモデルを横断して、コンテンツまたはナビゲーション情報（例えば、コンテンツテーブル（TOC）、ブックマーク、ハイパーリンク、ファインドテキスト（Find Text）等）を抽出するコマンドである。ステップ168が適用するデコレータコマンドを全く有さない（いいえ）場合、ステップ170は、結果のコードをS__OKに設定し、ステップ158において下りデータをリターンする前にDOMポインタを設定する。デコレータが適用される（はい）ことをステップ168が決定する場合、ステップ172が呼び出され、デコレータは、ステップ162で構築されたDOMを横断する。ステップ172が完了すると、結果のコードは、S__OKに設定され、DOMポインタは、ステップ158で下りデータをリターンする前にステップ172において設定される。

【0024】

図9は、サーバ側のデータ処理システム10のためのインデックス付け処理54のフローチャート180を示す。SCB44からの上りデータがステップ182に到着する。ステップ184は、データにインデックス付けするリクエストがあるかどうかをチェックする。ステップ184が偽（いいえ）と判定されると、ステップ186は、結果コードをE__FAILに設定して、その結果は、ステップ188でSCBに下りデータとしてリターンされる。ステップ184が真（はい）であると、システムは、ステップ190でドキュメントオブジェクトモデル（DOM）をトラバース開始する。ステップ192で、単一のDOMエレメントのコンテンツが取り出され、システム10は、ステップ194でインデックスを構築開始する。ステップ194で、システムは、Microsoftインデックス付けサーバ等の購入可能なソフトウェアインデックス付けパッケージを通常用いる。取り出されたDOMエレメントのコンテンツがステップ194でインデックス付けされた後、ステップ196は、DOMトラバースが完了するかどうかを観察するようにチェックする。答えがいいえであると、ステップ192および194は、DOMトラバースがステップ196で完了する（はい）であるまで再び処理される。インデックスは、次いでステップ198でインデックスデータベース56に格納され、結果のコードは、その結果がステップ188でSCBへの下りデータとしてリターンされるまで、ステップ200でS__OKに設定される。インデックスDB56は、ステップ198で利用され、MS SQL

サーバまたはOracle等の任意の販売用データベース商品であり得る。

【0025】

図10は、サーバ側のデータ処理システム10用のトランスセラプションモジュール58のプロセスフローチャート210である。上りデータは、SCBからステップ212に到着して、チェックされ、この上りデータがトランスセラプトリクエストかどうかをステップ214で調べられる。それがトランスセラプトリクエストではない場合（いいえ）、ステップ216は、結果のコードをE_FAILに設定して、下りデータとしてその結果をステップ218のSCBに戻す。データがトランスセラプトリクエストである（はい）と、ステップ214が判定する場合、ドキュメントオブジェクトモデルのIDを得るためにステップ220が呼び出される。DOM IDは、トランスセラプトキャッシュ60か、ドキュメントデータベース52のいずれかから正しいドキュメントを識別するために用いられる。ステップ222は、データがキャッシュ60で利用可能かどうかを調べるためにチェックする。答えがはいである場合、ステップ224は、トランスセラプトキャッシュ60からデータを得て、ステップ226は、その結果をS_OKに設定して、ステップ218で下りデータをSCBにリターンする。キャッシュにデータが無い場合（いいえ）、ステップ222は、メモリでロードされたDOMがあるか、および、メモリでロードされたDOMがDOM IDにマッチしているかどうかを調べる。

10

【0026】

システム10がキャッシングを用いているので、システム10は、特定数のドキュメント（例えば、最後に処理された32個のドキュメント）のためのDOMおよびDOM IDを保持し、DOM IDがキャッシュにあるDOMコンテンツのいくつかとマッチする場合、DOMが既にキャッシュに無いのなら、DOM IDがメモリに戻され、それは、処理された最後のドキュメントであったことを意味する。ステップ228におけるテストは、メモリにDOMが存在しない、または、DOM IDがマッチしない（いいえ）（DOMがキャッシュにないことを意味する）ことを判定する場合、ステップ230が呼び出され、ステップ232に進む前にドキュメントデータベース52からDOMを取り出す。システム10は、DOMがデータベースに保存されるか、DOMを最初に生成するために用いられる元のドキュメントを再処理せずすぐに取り出すためにファイリング可能にする。メモリにDOMがあり、ドキュメントIDがマッチする（はい）ことをステップ228が判定する場合、ステップ232は、トランスセラプトを得るために呼び出される。ステップ234は、上りデータとしてステップ212でSCBから受信したSDBコマンドが予想したキャッシングフラッグセットを有するかどうかを検証するために観察する。フラッグが設定される場合（はい）、ステップ236は、ステップ238に進む前に非同期の予想キャッシングプロセス62を開始する。フラッグが設定されないことをステップ234が判定する場合（いいえ）、ステップ238が呼び出され、ステップ232で呼び出されたトランスセラプトデータは、キャッシュ60に託される。ステップ240は、次いで、コミットが完了したかどうかを調べるために繰り返してテストし、テスト結果が否となる場合にステップ238に戻る。ステップ240がコミット完了のために真をリターンする場合、ステップ224が呼び出され、トランスセラプトデータは、キャッシュから取り出される。ステップ226は、結果をS_OKに設定して、ステップ218で下りデータをSCBにリターンする。

20

30

40

【0027】

図11は、無線デバイスまたはパーソナルデジタルアシスタント(PDA)に常駐する本発明の無線クライアント処理システムの詳細図250である。リーダ252は、関数を無線PDA/デバイスオペレーティングシステム270に呼び出す複数の別々のソフトウェア関数エレメントおよびライブラリ254~266を含むソフトウェアアプリケーションである。オペレーティングシステム270は、オペレーティングシステム270を形成する複数の別々のソフトウェア関数を含む。無線PDA/デバイス用の周知のオペレーティングシステム(OS)は、Palm CorporationのPalm OS、Microsoft CorporationのWindows(R) CE、Sun Cor

50

porationのJ2ME、SymbianのEPOCおよびQualcommのBREWである。本発明のクライアント側のソフトウェア関数は、通常UCSと呼ばれる、定義されたユニバーサルコンテンツストリームを読み出すように設定されている。リーダ252は、テキスト、ベクトルおよびイメージベースのデータをデバイス12にレンダリングするように用いられるグラフィック命令のセットを提供するグラフィック関数ライブラリ266を有する。これを達成するために、ライブラリ266は、デバイスOS270からの利用可能なグラフィックオペレーションを用いる。コア関数ライブラリ264は、メモリアクセスおよびハンドリング272、ファイル入力/出力および/またはデータベースアクセス274、およびインターネット16からデータを取り出すための無線データアクセス276用にデバイスOS270から導かれた関数のセットを提供する。次いで、リーダ252は、ビューモード254、256、258および260で定義されたグラフィックユーザインタフェースGUIを提供可能なコア関数ライブラリ264およびグラフィック関数ライブラリ266を呼び出すアプリケーションコマンドレイヤー262を有する。

10

【0028】

UCSからロードされたデータに基づいて、イメージビュー254は、イメージまたは画像に基づくデータを表示し、シートビュー256は、スプレッドシートスタイルグリッドのスプレッドシートデータを表示し、ドキュメントビュー258は、リッチ(rich)かつ直感にテキストのコンテンツを表示し、ファイルビュー260は、デバイス12にUCSファイルをロードかつ保存するためにユーザインタフェースを表示する。リーダ252において層状に設計されているので、コア関数ライブラリ264およびグラフィック関数ライブラリ266のみが、クライアント側の処理システムが置かれる無線/pdaデバイス12の各々において異なる。これにより、本発明がPalm OS、Windows(R) CE、J2ME、EPOCおよびBREW等(ここで、2大プログラミング言語は、プログラミングの分野における当業者において周知のJava(R)およびC++である)の異なるオペレーティングシステムに素早く移動またはポートされることが可能となる。USCバイナリファイルは、インターネット16を介してサーバ側のデータ処理システム10からデバイス12に送達され、無線クライアントリーダ252は、コア関数ライブラリ264において無線データアクセス(ロード)276用の関数を用いることによってこのデータを取り出す。コア関数ライブラリ264では、順に、無線デバイスPDA/デバイスOS270のリモートHTTP/TCP/IPアクセス276用にライブラリを呼び出す。

20

30

【0029】

図12は、無線PDA/デバイス12に常駐するクライアント側の処理アプリケーションの詳細なフローチャート280である。ステップ282において、インボックス(Inbox)アプリケーション等のデバイス12上の別のアプリケーションにより起動されるか、ユーザがリーダ用のアプリケーションアイコンを選択して、そのアプリケーションを呼び出すかのいずれかによって、リーダ252が呼び出される。ステップ284において、渡されたユニバーサルコンテンツストリーム(UCS)は、必要があれば、ローカルまたはリモートであることを検証される。ローカルである場合、UCSは、既にデバイス12に格納されており、ステップ286は、ファイルまたはデータベース288からローカルUCSストリームを読み出す。UCSがステップ287でリモートである(デバイス12に常駐していない)と判定される場合、ステップ290は、リモートUCSストリームを読み出すために呼び出される。次いで、ステップ292は、インターネット16を用いてサーバ側のデータ処理システム10に常駐するリモートUCSを得る。ステップ286か290かのいずれかが完了すると、取り出されたデータがパース(parse)されるステップ294まで実行が続く。次いで、ステップ296は、データをデバイスメモリにロードして、ステップ298は、リモートまたはローカルUCSデータのパーシングが完了するかどうかを調べるためにテストする。ステップ298がローカルデータ用に完了しない場合(いいえ)、ステップ294は、ステップ298がローカルUCSファイル用に真

40

50

である（はい）まで、繰り返して呼び出される。リモートUCSパーシング用のステップ298が、完了しない場合（いいえ）、クライアントへの任意のさらなるトランスセラプト用にサーバ10に以前として常駐していると記述された任意のUCSデータのために、ステップ300は、リモートリンクリストを生成する。次いで、ステップ302は、リモートリンクによってUCSメモリを更新して、リモートパーシングステップ298が完了する（はい）までステップ294へと続く。

【0030】

UCSデータがパースされ、メモリにロードされた後、ステップ304は、UCSデータのビュタイプまたはビュタイプの組み合わせを決定する。呼び出されたイメージビュー254、ドキュメントビュー258またはシートビューモード256のいずれかのビュタイプに応じて、UCSデータは、ステップ306で表示される、または、レンダリングされる。ステップ306でUCSデータを表示した後、クライアントデータ処理アプリケーションは、ステップ308でユーザの入力を待機する。ユーザがメニューオプション、またはトラック/スクロールホイール、スタイラス(stylus)またはキーパッドを用いてロードされたドキュメント内でナビゲートする場合、ステップ308は、ユーザが脱出オプションを選択することによってリーダ252を終了させるまで、繰り返して呼び出され、ステップ310は、モバイルリーダを脱出して、ステップ282の前の呼び出しアプリケーションにリターンする。ユーザがさらなるUCSデータのためにステップ308でより多くのリクエストを選択するまたは呼び出す場合、ステップ290は、ユーザがステップ308でアプリケーションを終了させるまで、上述されたような同じ以下のプロセスフローで再び呼び出され、ステップ310は、モバイルリーダを脱出して、そのステップの前に呼び出しアプリケーションにリターンする。

【0031】

図13は、ドキュメント処理モジュール50のためにロードされた仮定のドキュメントと共にドキュメントエンジン構造を示す図320である。ドキュメントエンジンの構成は、以下のモジュール（エンジンモジュール322、ディスティラーモジュール324とロードされたドキュメント326とデコレータモジュール（図示せず）とのセット）としてカテゴリー化された各コンポーネントの集合である。ディスティラーモジュール324の機能は、主に、異なるドキュメントタイプを取り扱うことである。このモジュールのセットは、特に公知のドキュメントフォーマットをパースし、ドキュメント構造を構築し、所望の場合、ドキュメント構造を公知のドキュメントフォーマットに戻すことに固執する役目を有する。新規のディスティラーモジュール324が動作時間に追加され、または、取り除かれて、柔軟な環境のために必要なものを収容するという意味で、ディスティラーモジュール324のセットは開かれている。エンジンモジュール322の機能は、主に、動作しているディスティラーモジュール324、所望の場合、ロードされたドキュメント326をホストして、これらのために均一の環境を提供することである。クライアントのために共通の挙動を提供する（例えば、ドキュメントをローディングすること）ことも用いられる。複数のドキュメント構造モジュールは、ロードされたドキュメントの構築しているピースである。これらのピースは、（ロードされたドキュメントを示し、他の全てのノードを含む）ドキュメントノード328、（ある程度の詳細なドキュメントを有するコンテンツを示す）コンテンツノード330、（他のノード用のコンテンツを示す）コンテンツノードの対332、334、（テキストコンテンツを示す）テキストコンポーネント336、（イメージコンテンツを示す）イメージコンポーネント338、（ベクトルコンテンツを示す）ベクトルコンポーネント340、および、（スプレッドシート等のテーブルデータを示す）テーブルコンポーネント342を含む。ドキュメント構造326は、ツリー状の構造（ノードを共有する可能性を有するので、より適切な記述は、グラフ状である）である。この事実によって、固執したドキュメント構造によって制約されることなく所望の任意の態様でユーザがドキュメントを実用的に構築することを可能にする。

【0032】

各ノードが（子供および/親に対する）方法の様な集合をインプリメントするという事実

によってこのことが達成される。レンダリングの様な挙動は、それ自体ノードに取りつけられ得るか、外部のコンポーネントとして提供され得るかである。このことは、ドキュメント構造の標準的なイタレータ (i t e r a t o r) のセットの存在によって容易にされる。意図する動作に必要なこと (速度、大きさ、配置等) を考慮することによって、内部の挙動と外部の挙動との間で選択が成され得る。デコレータモジュールの機能 (図 1 3 に示されない) は、主に、ドキュメント構造に対して選択されてカスタマイズされたドキュメントの動作を実行することである。コンテンツ、ブックマークまたはハイパーリンクのリスト、検索動作等のテーブルを抽出することが例として含まれる。ここでは、トランスセラプション技術が用いられる。これらは、それぞれ独自のフレームワークを有する分離したモジュールであり、これらのモジュールは、他のエンジンコンポーネントと完全に独立して動作する。実際、エンジンは、インストールされているデコレータモジュール無しで動作し得る。これらは、さらなる機能を加える。デコレータモジュールフレームワークは、 (それ自体でコンポーネントである) デコレータマネージャ、および、特定の動作を実行するインストールされたモジュールのセットで構成される。各モジュールは、フレームワークに統合して、デコレータマネージャによって認識されるために、予め決定された方法で登録する必要がある。デコレータは、ドキュメントオブジェクトモデル (D O M) をナビゲートする単一の目的のためにソフトウェア開発の分野で当業者によって共通して参照されるコンポーネントであり、特定のコンテンツおよび / またはナビゲート情報を取り出す。言い換えると、デコレータコンポーネントがコマンドを構成することが記述され得る。いかに多くのデコレータが D O M をナビゲートするために用いられ得るかということに対して制限はない。通常のデコレータ (コマンド) は、ドキュメントのブックマークを取り出し、ドキュメントのハイパーリンクを取りだし、ドキュメントのコンテンツテーブルを取り出し、T O C が利用可能ではない場合にドキュメントの要旨を構築し、特定のページを取り出し、ドキュメント内のテキストコンテンツを検索し、特定のフォントスタイルまたは情報を取り出し、ドキュメント内のイメージまたはイラストレーションを取り出す。

10

20

30

40

50

【 0 0 3 3 】

図 1 4 は、ドキュメントエンジン 3 2 0 内の親ノード 3 5 2 および子供ノード 3 5 4 の関係 3 5 0 を示す。親 3 5 2 と子供 3 5 4 との間の実際のリンクは、2 つの間のポイントよりも若干複雑である。それらの間において、リンクの属性を保持する役割を果たす、いわゆるスペシフィアー (s p e c i f i e r) 3 5 6 オブジェクトが存在する。関連している親または子のオブジェクト I D S に対する情報は、通常リンクの属性によるものである。このことは、グラフ状の構造であり、単一の親が多数の子供を有し得、1 つの子供も多数の親オブジェクトを有し得るためである。

構造化または非構造化電子ドキュメントの形式か、絵またはイラストレーション等のグラフィックイメージ、および、ベクトル形式か A から B への線を描く方法に関する記述として周知の形式で示された描画データによって通常表されたテキストおよび非テキストのコンテンツを具現化するデータストリームの形式で入力を提供することが、本発明の動作である。電子ドキュメントは、ビジネスまたは個人的な通信レター、製品情報および製品の説明書、部品のマニュアル、会社または個人の経営情報等が最良として記述されているが、これらに制限されない。図 1 5 および 1 6 は、それぞれ構造化ドキュメントおよび非構造化ドキュメントを表す。

【 0 0 3 4 】

図 1 5 は、通常非構造化ドキュメント 3 6 0 がいかに見え得るかを示す。非構造化によって、ドキュメントが最小のナビゲーション情報および構造化エレメント (例えば、ヘッダ、フッタ、テーブルおよびフォントスタイル) を有することが意図される。これらを使用することによって、発明のクライアント側のデータ処理システム 2 5 0 に戻って部分的なドキュメントコンテンツを送達し、再構築する目的のために、このようなフォーマットおよびナビゲーション情報を直接抽出することができる。サーバ側 1 0 データ処理能力は、従って、非構造化ドキュメントデータを解析することによって、ナビゲーションおよび

再構築情報等を構築するために用いられる必要がある。

【0035】

フォントサイズ、フォントタイプ、フォントスタイル、および、フォントポジショニングの変化を検出することによって解析が行われる。これを解析することによって、全てのテキストコンテンツエレメント362は、以下の基準によってヘッダとしてマークされる。用いられる最大のフォントサイズ=14、フォントスタイル=B o l d、フォントタイプ=A r i a l。これは、テキストパラグラフ定義364によって強化される。テキストパラグラフ定義364は、全て、フォントサイズ=10、フォントスタイルは無く、フォントタイプ=T i m e sである。この情報に基づいてヘッダ362を識別化することによって、本発明は、非構造化ドキュメントに対しては存在しないコンテンツテーブルの形式でナビゲーションのヘルプを構築可能である。組み込まれたイメージ366は、画像挿入ポイント座標x, yおよび全てのピクセルにおける幅および高さ情報を有する。次いで、本発明のシステム10は、イメージ366をクライアント側のアプリケーション処理システム12にレンダリングする場合に、イメージ366へリンクを提供し得、マークされたエレメント368は、テキストパラグラフ364内のフォントスタイルの変化が原因となつて、ヘッダとしてもマークされるが、エレメント368がテキストパラグラフ364内に位置すると、他のヘッダ362よりも低い重要度でランク付けされる。エレメント368がナビゲーションの重要さを有していると判断されるので、コンテンツテーブルの形式でナビゲーションエイド(a i d e)を形成する場合に、エレメント368は、次いで、サーバ側10のアプリケーション処理によって用いられる。同様に、タイプ370および372のエレメントは、本発明のサーバ側の処理能力によって検出される。なぜなら、これらのエレメントのフォント位置が緊密であり、加えて、フォントスタイルB o l dおよびI t a l i c、および、スペーシング用のタブ(T A B)文字を使用しているからである。これら全ては、このセクションがドキュメント360に対して重要であり、テーブルコンテンツを構成する際にこのセクションがナビゲーションエイドポイントとしてマークされ得ることを示している。最終的に、ページエレメント374は、ドキュメントページ1にある全ての非構造化コンテンツに対する公知のナビゲーションインデックス参照と関連するページ番号の記述を定義する。

【0036】

図16は、テーブルコンテンツの形式のドキュメントナビゲーション構造を識別する目的で、および、クライアント側の処理アプリケーション12でコンテンツの抜粋を再構成することを意図してコンテンツの抜粋を定義する目的でサーバ側データ処理システム10によって処理され得る非構造化ドキュメント380を示す。コンテンツテーブル(T O C)382は、構造化ドキュメントに存在するエレメントとして定義される。ヘッディング1およびヘッディング2として参照されるコンテンツエレメント384および386がパラグラフヘッダ388として定義され、ドキュメント自体によってT O Cエンティティ382の構築のときにナビゲーションエレメントとして用いられる。従って、サーバ側のデータ処理アプリケーション10は、非構造化ドキュメントのために必要とされる処理を必要とすることなく、これらのエレメントを直接抽出することができる。さらに、システム10は、構造化ドキュメントをパースして、ブックマーク1へのハイパーリンク390を識別する。ハイパーリンクは、構造化ドキュメント内のオブジェクトまたはエンティティ間のナビゲーションリンクである。この構造化ドキュメントは、それらが接続され、関連していること、および、ハイパーリンクと相互作用することによって、ユーザがドキュメントの異なる部分にナビゲートできることを示す。システム10は、ハイパーリンクエンティティ390の記録を作り、その記録をナビゲーションのために用いる。ブックマークエンティティ392は、構造化ドキュメント380自体内に定義されている別のエレメントである。この構造化ドキュメント自体は、ナビゲーションアンカー(a n c h o r)ポイントを提供し、システム10は、ナビゲーションハイパーリンクエンティティ390の実行を可能にする目的のためにこのナビゲーションアンカーポイントを記録する。ハイパーリンクエンティティ392、394、396は、外部のハイパーリンクであることによつ

てハイパーリンクエンティティ 390 とは全く異なる。外部ハイパーリンクは、他のコンテンツ、または、外部アプリケーション、または、構造化ドキュメント 380 自体に定義されるプロセスに対するリンクオブジェクトとして定義される。

【0037】

1つのハイパーリンク 392 は、ウェブサイトアドレス `www.arizana.com`、または、`www.air2web.com` への第 2 のハイパーリンク 394 へと示すアドレスによってインターネットブラウザを立ち上げ、一方で、リンク 396 は、`sales@arizana.com` へと設定された「To」アドレスによって e メールアプリケーションプログラムを立ち上げる。サーバ側のデータ処理システム 10 は、構造化ドキュメントを処理する一方で、ナビゲーションまたは可能なアプリケーションを立ち上げる目的で、本発明を用いるソフトウェアアプリケーションのために構築されたドキュメントオブジェクトモデルにこれらのエレメントを記録する。構造化ドキュメント 380 のイメージエレメント 398 は、イメージ/ピクチャに対する情報、位置 x, y および幅、高さ情報を提供する。システム 10 の発明は、DOM に記録し、従って、視覚的にかつ内容としても正しいコンテキストで無線デバイス 12 にあるこのエレメントを再構成する能力をクライアントコンポーネント 12 に与える。構造化ドキュメント 380 によって定義されたテーブルエレメント 400 は、テーブル状の態様で表示かつ組織化されたコンテンツを含む。本発明のサーバ側のデータ処理機能は、この情報を読み出し、ドキュメントエンジンテーブルコンポーネントを利用して DOM を構築する際にそれをマッピングする。このドキュメントエンジンテーブルコンポーネントによって、本発明のクライアント側の処理システムは、無線デバイス / `pda` 12 にテーブル状の構造で DOM を再構築可能である。

【0038】

構造化ドキュメントのグラフエレメント 402 は、バーチャート等のグラフタイプフォーマットの情報を含み、Microsoft Corporation による MS Word のようなワードプロセッサ等の構造化ドキュメントを読み出すアプリケーションによってチャートを描く際に、テーブルエレメント 400 に提供されたデータにリンクする。サーバ側のデータ処理能力は、無線デバイス / `pda` 12 においてクライアント側のデータ処理システムにグラフを再構築させることを意図して構造化ドキュメントを処理する際に、グラフリンクおよびレンダリング情報を取り出し、記録する。ヘッダエレメント 404 は、構造化ドキュメントにある各ページに対するページヘッダ情報を定義して、一方で、フッタエレメント 406 は、構造化ドキュメント 380 の各ページに対して再利用されるページフッタ情報を定義する。サーバ側のデータ処理能力は、エレメント 404 および 406 を記録し、適切なエンティティを DOM に構築する。ページエンティティ 408 も、ナビゲーションインデックスを提供し、一方で、構造化ドキュメント 380 をナビゲートする。なぜなら、各ページエンティティは、最初のページで開始し、最後のページで終わるページ範囲内の各ページをユーザが参照可能とする連続的なエントリであるからである。図 16 によって表された構造化ドキュメントも、メタデータ 410 を参照するドキュメントに関する非視覚的な記述情報を埋め込んでいる。本発明のサーバ側のデータ処理能力は、この情報を抽出可能であり、それを DOM に格納可能である。この DOM では、その情報は、後に、システムに 10 によって用いられ得、ドキュメント情報（構造化ドキュメントに格納されたメタデータによって定義された、ドキュメント名、著者、日付、生成時間等）を提供し得る。

【0039】

図 17 は、本発明のサーバ側のデータ処理システム 10 によって構築されたドキュメントオブジェクトモデル 420 を示す。イラストレーション 420 は、ロードされたドキュメントの一般的な構造 422 を示す。DOM のエントリポイントは、コンテンツノード 424 である。複数のコンテナ 426 の各々は、異なるタイプの多数のコンポーネントを有し得る。本発明のサーバ側のデータ処理アプリケーションは、定義された 4 つのコンポーネントタイプを有する。それらは、テーブルコンポーネント 430、テキストコンポーネント 431、イメージコンポーネント 432 およびベクトルコンポーネント（図示せず）で

ある。各コンポーネントも、参照コマンドの形式で、同じコンテナ 4 2 6、または、異なるコンテナの下で他のコンポーネントによって参照され得る。

【0040】

図 1 8 は、ドキュメントオブジェクトモデル (DOM) 4 2 0 を構築した後、および、図 1 6 によって示された構造化ドキュメント 3 8 0 をロードした後にサーバ側のデータ処理システム 1 0 によって用いられたテキストコンポーネント 4 3 0 を示す。各テキストコンポーネント 4 3 0 は、1 つ以上のセクションコマンド 4 4 0 を有し得、任意の標準的な構造化または非構造化ドキュメントとして通常見られるために、複数のパラグラフ 4 4 2 は、1 つの単一セクション 4 4 0 に含まれ得る。パラグラフオブジェクト 4 4 2 は、定義された 1 つの属性 4 4 4 を有する。この 1 つの属性 4 4 4 は、コンテンツハイパーリンクのテーブルに対する参照である。ハイパーリンクスタートオブジェクト 4 4 6 自体は、その属性オブジェクト 4 4 8 によって定義される。この属性オブジェクト 4 4 8 は、ハイパーリンク ID およびリンクの送信先を記述する。それから、テキストセグメントオブジェクト 4 5 0 は、テキストオブジェクト 4 5 0 に対するテキストコンテンツ自体を保持する記述属性 4 5 2 を有する。パラグラフオブジェクト 4 4 2 の最後に、ハイパーリンクエンドオブジェクト 4 5 4 は、ハイパーリンクペア ID を含む属性オブジェクト 4 5 6 によって構築される。

10

【0041】

図 1 9 A は、ロードされた構造化ドキュメント 3 8 0 のドキュメントオブジェクトモデル 4 6 0、および、サーバ側のデータ処理システムによって用いられたパラグラフに対する DOM 構成を示す。パラグラフオブジェクト 4 6 2 は、パラグラフ 4 6 2 の関連するスタイルおよびテキストのコンテンツを記述する属性オブジェクト 4 6 4 を有する。フォントオブジェクト 4 6 6 は、フォントコマンドを含み、関連する属性オブジェクト 4 6 8 は、フォント記述およびフォントスタイル情報を含む。パラグラフオブジェクト 4 6 2 にリンクされた次のコマンドオブジェクトは、ブックマークオブジェクト 4 7 0、および、その記述属性オブジェクト 4 7 2 である。この記述属性オブジェクト 4 7 2 は、ブックマーク ID を含む。次に、パラグラフオブジェクト 4 6 2 は、テキストオブジェクト 4 7 4 にリンクされ、このテキストオブジェクト 4 7 4 は、テキストコンテンツ自体の価値を有する属性オブジェクト 4 7 6 に関連する。図 1 9 B は、従って、図 1 9 A によって開始された DOM パラグラフ構成 4 6 0 の続きである。パラグラフオブジェクト 4 6 2 にリンクされた次のコマンドオブジェクトは、テキストコマンドオブジェクト 4 7 8、および、テキスト値を含む関連する属性オブジェクト 4 8 0 である。次は、新規のフォントコマンドオブジェクト 4 8 2 であり、関連する属性オブジェクト 4 8 4 は、フォントタイプおよびスタイルの定義を含む。次いで、パラグラフオブジェクト 4 6 2 にリンクされた以下のオブジェクトは、それぞれ、テキストコマンドオブジェクトおよびフォントコマンドオブジェクト、および、関連する属性 4 8 6 ~ 4 9 6 であり、パラグラフ構成が参照コマンドオブジェクト 4 9 8、および、その参照値を含む関連属性オブジェクト 5 0 0 で終わる。

20

30

【0042】

図 2 0 は、ドキュメントオブジェクトモデル (DOM) を構築した後、および、図 1 6 によって示された構造化ドキュメント 3 8 0 をローディングした後にサーバ側のデータ処理システム 1 0 によって用いられたテーブルコンポーネント 4 2 8 を示す。各テーブルコンポーネント 4 2 8 は、多数の列オブジェクト 5 1 0、5 1 2 を有し得る。ここで、1 つ以上のセルが含まれてもよい。図 2 0 は、サンプル構造化ドキュメント 3 8 0 にあるサンプルテーブルの最初の列を示す。テーブル列コマンドオブジェクト 5 1 0 は、セルコマンドオブジェクト 5 1 4 にリンクする。このセルコマンドオブジェクト 5 1 4 は、セルのテキストコンテンツ値を含む関連する属性オブジェクト 5 1 6 を有する。さらなるセルコマンドオブジェクト 5 1 8、5 2 0、5 2 2 および 5 2 4 は、次いで、これ以上のデータが列として利用可能ではなくなるまで、テーブル列コマンドオブジェクト 5 1 0 にリンクされる。

40

【0043】

50

図 2 1 A ~ 2 1 F は、インターネット 1 6 を介してデータが無線で送達することに対して効果的かつ最適化された方法でデータを送信して、クライアント側のデータ処理システム 1 2 によって処理されるために、本発明で用いられたユニバーサルコンテンツストリーム (U C S) の形式で出力されたコンテンツにおけるサーバ側のデータ処理システム 1 0 の記述を示す。図 2 1 A ~ 図 2 1 F は、2 つのテーブルを有する。各テーブルは、図 1 6 および 1 5 でそれぞれ表される構造化ドキュメント 3 8 0 および非構造化ドキュメント 3 6 0 を生成するコマンドを示す U C S データファイルを出力する際のサーバ側の処理能力を示す。

【 0 0 4 4 】

図 2 2 を参照して、U C S フォーマット 5 3 0 のレイアウトが示される。U C S (ユニバーサルコンテンツストリーム) フォーマット 5 3 0 は、サーバ 1 0 とクライアント技術またはデバイス 1 2 との間のメカニズムおよびデータ送信層を提供する。このフォーマットの主な目的は、利用可能なバンド幅域、最適な方法で送信されている必要とされる視覚的イメージのみを可能とすることである。フォーマット 5 3 0 は、グラフィックイメージ / ピクチャ、2 D / 3 D 描画、ならびに、ワードプロセッサおよびスプレッドシートのコンテンツ等の様々なテキストデータの転送を支援する。

【 0 0 4 5 】

U C S は、トランスセラプト技術および任意の利用可能なバンド帯を利用するトランスセラプトコマンドおよびデータ転送に加えて、情報送信のために様々な異なる標準フォーマットを含み得る。U C S 内に最適な標準フォーマットを埋め込むことは、トランスセラプトサポートを必要としないクライアント技術を容易に統合することへとつながる。イメージ / ラスターデータ等のフォーマットに対して、J P E G 2 0 0 0 等の現在の / 将来の基準は、サーバ 1 0 からクライアント 1 2 へのデータロスのない圧縮に対して十分なサポートを提供する。

【 0 0 4 6 】

サーバ 1 0 と他のクライアントとの両方によってアクセスされ得る業界標準フォーマットの例は、以下の 4 つのタイプである。

- ・テキスト、メタデータおよびコンテンツ情報用の X M L
- ・ベクトル情報用の S V G
- ・イメージ用の J P E G 2 0 0 0
- ・ O p e n G L 3 D 表示

30

他の基準フォーマットまたは専有フォーマットは、コンポーネントベースのサーバ 1 0 のプラグ可能 (p l u g a b l e) およびスケール可能 (s c a l a b l e) なアーキテクチャによって容易に追加されるか、取り除かれるかされ得る。

【 0 0 4 7 】

U C S フォーマット 5 3 0 は、オプションの業界標準フォーマットと共に U C S フォーマット 5 3 0 を作る 3 つの異なるデータ層を含む。この 3 つの専有層は、T C L 層、U C S テキストコンテンツ層 5 3 4、および、U C S イメージまたはベクトルデータ層 5 3 6 である。

【 0 0 4 8 】

T C L (トランスセラプトコマンド言語) 層 5 3 2 は、サーバ 1 0 とクライアント 1 2 との間の双方向通信のためのデータチャネルを提供する。クライアント 1 2 は、リクエストをサーバ側のクライアントハンドラに送信し、逆に、トランスセラプトデータストリームで応答を受信する。T C L は、以下のタイプのコマンドを取り扱う。すなわち、視覚的イメージ用のトランスセラプトリクエスト、V Q (ビューおよびクエリ) コマンド、ディレクトリ情報、刊行されたドキュメントの検索 / 取り出し、および、任意の必要とされた第 3 パーティ情報同期である。T C L 言語自体は、また、容易に適応するために、および、さらに拡張するために、X M L を利用して構築される。

40

【 0 0 4 9 】

U C S ベクトルストリーム 5 3 6 は、高度に最適化されたベクトルフォーマットであり、

50

貧弱なクライアントソフトウェアにベクトル描画能力を提供する。これにより、この貧弱なクライアントソフトウェアは、小さなPDAデバイスまたはハンドヘルドデバイスに対してさえもより高度で精密なベクトル表示を提供することができる。ベクトルストリーム転送およびレンダリングは、TCL層532を通して制御される。このTCL層532は、トランスセラプトハンドラが、クライアント12によって要求されたクリップされたビュー可能な領域に対してベクトル表示を生成し、そのリクエストを生成した制限されたクライアント表示領域のために、そのベクトル表示を最適な方法でレンダリングするのみであることを保証する。UCSフォーマット530は、新規の専有層またはフォーマットが容易に追加され得る状態で設計される。イメージ、ベクトルおよびテキストのトランスセラプションのために用いられたフォーマットのうちのいくつかも、第3パーティによって統合される必要があるならば、取り換えられるか、追加されてもよい。

10

【0050】

ユニバーサルコンテンツストリーム(UCS)530は、バイナリファイルフォーマット規格である。この規格は、元のドキュメントコンテンツ(このドキュメントコンテンツは、テキストコンテンツ、イメージデータ、ベクトルデータおよび処理されたドキュメントのためのナビゲーション情報を含むが、これらに制限されない)の送達を可能にする。UCSフォーマットは、利用可能なバンド帯域を考慮すると、無線デバイス12の現在の制限を克服する際に非常に効果的である。今日一般的に、このバンド帯域は、理論的にはさらに高速にすることができるが、9.6から通常48までの範囲である。UCSは、様々な方法で制限された帯域幅の問題を取り扱う。

20

【0051】

UCSフォーマット530によって、元のドキュメントファイルサイズは、90%まで減少可能となるが、クライアント側の処理能力を用いて無線デバイスにあるドキュメントを再構築するための全ての本質的なコンテンツおよび必要なフォーマット情報を保持している。このサイズの減少は、元のドキュメントアプリケーションによって用いられる本質的ではないフォーマット情報(例えば、デスクトップコンピュータ上でリッチ表示されたMS Word)を取り去ることによって達成される。本質的ではないフォーマット情報は、異なるフォントスタイル、フォントカラー、および、改訂情報等を用いることであり得る。これらは、通常、無線デバイス12上に値を有していない。なぜなら、デバイススクリーンは、モノクロームまたはグレースケールであり得、そのデバイスは、通常、1つのフォントタイプのみであるからである。UCSファイルフォーマットはまた、クライアント側で大きさをさらに減少するために、サーバ側でUCSファイルを圧縮して、解凍することを可能にする。圧縮には、LZW等の圧縮技術形式で共通に利用可能な技術が利用され得る。さらに、(伝送用のコンテンツを除いた)トランスセラプトを生成する際のサーバ側の処理能力によって、繰り返しのプロセスでサーバからクライアントへの個々の伝送に対して複数のUCSファイルの生成を可能にする。このプロセスでは、無線デバイスに送信された生じたUCSファイルは、ユーザからの特定のコンテンツクエリに対する結果である。サーバからのコンテンツを要求する、無線デバイスでのユーザからの繰り返しクエリプロセスは、例えば、デバイスに送達される各10個のUCSファイルを送達することにつながり得る。UCSファイルフォーマットは、一貫したビューイングまたは読み出しを通して元のドキュメントがメモリにある各ファイルによって再構成され得るように、これらの各UCSファイルがどのように共にリンクするかに関する情報を含む。UCSフォーマットはまた、ドキュメントオブジェクトモデル(DOM)Idを含み、サーバ10の処理能力が要求しているデバイス12へ正しいDOM情報を送達可能であることを保証する。さらに、UCSファイルフォーマットはまた、クライアントがドキュメント内からより多くのコンテンツを要求することを可能にするインターネットハイパーリンクを含む。コマンドコードはまた、UCSフォーマットに組み込まれ得、サーバ側のデータ処理能力によって特定の処理の実行を容易にし得る。これは、通常、特定のコンテンツに対するリクエスト(例えば、DOM等内に格納されたナビゲーション情報のルックアップ)である。

30

40

50

【 0 0 5 2 】

(用語の定義)

1. アナトミゼーション - 全てのコンテンツおよびナビゲーションドキュメント階層構造を含む均一のドキュメントオブジェクトモデルを構築する目的で、構造化または非構造化ドキュメント、あるいは、データソースを (p a r s i n g) パーシングするおよび解析する特定の技術プロセス。

2. ブックマーク - ドキュメント内にある位置 X から位置 & へのナビゲーション参照に関連するナビゲーションリンク。

3. DOM - ドキュメントオブジェクトモデル (D O M)、すなわち、ワールドワイドウェブコンソーシアムによって開発されているプログラミングインタフェース規格であり、オブジェクトとして X M L ベースのドキュメントを生成かつ構造化する方法を定義する。

4. 外部ファイル - 関連する外部データ、または、元のドキュメントの外部にあるドキュメントファイルに対する参照。

5. 外部オブジェクト - 元のドキュメントの外部にある関連する外部データオブジェクトに対する参照。

6. フォントカラー - フォントの色。

7. フォント位置 - レンダリング x , y ピクセルまたは文字の位置情報。

8. フォントサイズ - フォント文字のサイズをポイントで示すフォントの大きさ。

9. フォントスタイル - 普通、イタリック、ボールド、下線等のフォントのスタイル。

10. フォントタイプ - ドキュメントで用いられるフォントタイプの名前 (例えば、タイムズローマン)。

11. フッタ - コンテンツを含むドキュメントにあるそれぞれのまたは全てのページの底部にある脚注。

12. ヘッダ - コンテンツを含むドキュメントにあるそれぞれのまたは全てのページの上部にあるヘッダ。

13. ハイパーリンク - ブックマークまたは外部データ、あるいは、コンテンツソースを参照するドキュメントコンテンツの外部または内部にあるナビゲーションリンク。

14. ページ - 定型ページまたはプリンタページの中断ページの文字によって通常分離されたコンテンツ情報を含むドキュメントにあるそれぞれの (a g e) 年代を定義するもの。

15. パラグラフ - 中断ラインによって分離された結合性のある本体でグループ化されたテキスト線の集合。

16. 予測キャッシング - 無線 / 有線デバイスへの必要とされた送達 / 伝送の前にサーバにある該当または関連するコンテンツをキャッシュして、帯域幅およびネットワーク待ち時間を最小化するように設計された特定の技術プロセス。

17. コンテンツテーブル - ドキュメントコンテンツのナビゲーションを通知して、支援するためにドキュメントコンテンツに階層構造のナビゲーション構造を提示するもの。

18. タイトルスタイル - ヘッディング 1、ヘッディング 2、ヘッディング 3 のようにテーブルコンテンツの構成を考慮した上でのドキュメントの中央生成アプリケーションにおける予め定義されたスタイル。

19. トランスセラプション - 要求されたコンテンツの抜粋のみを送達して、これをクライアントに伝送する特定の技術プロセス。

20. U S C - 最小化されたデータサイズのために最適化されたコンテンツデータフォーマットするユニバーサルコンテンツストリームであって、無線 / 有線デバイスへの送達のためのコンテンツの文脈、ナビゲーションおよび提示の情報を保存する。

21. X M L - 共通の情報フォーマットを生成する柔軟な方法であり、ワールドワイドウェブ、イントラネットおよびその他にあるフォーマットとデータとの両方を共有する拡張マークアップ言語。

【図面の簡単な説明】

【 0 0 5 3 】

10

20

30

40

50

【図 1】図 1 は、データ処理システムまたはサーバを示す。ここで、情報は、インターネットを介するクライアントの無線デバイスと通信するために用いられる。

【図 2】図 2 は、図 1 によって示されたデータ処理システムおよび無線デバイスのブロックダイアグラムを示す。

【図 3】図 3 は、本発明の個々の処理モジュールを詳述する図 1 のサーバに常駐する本発明のブロックダイアグラムである。

【図 4】図 4 は、本発明に提出されたドキュメントのデータパッケージングを示す。

【図 5】図 5 は、サーバ上のデータ処理システムによって処理された提出要求の処理フローを示すフローチャートである。

【図 6】図 6 は、サーバデータ処理システムのサーバ通信バス (S C B) モジュールによって処理された入来リクエストの処理フローを詳述するフローチャートである。 10

【図 7】図 7 は、光学的文字認識 (O C R)、画像拡張およびベクトル変換に対するラスタ (R 2 V) 用の外部処理を呼び出す前処理モジュールの処理フローを詳述するフローチャートである。

【図 8】図 8 は、ドキュメントオブジェクトモデル (O D M) を構築する際のドキュメントアナトミー (a n a t o m y) を構成するために用いられるデータ処理システムアナトミゼーションモジュールの処理フローを詳述するフローチャートである。

【図 9】図 9 は、提出されたドキュメントのコンテンツにインデックス付けするインデックスモジュールの処理フローを詳述するフローチャートである。

【図 10】図 10 は、無線 / 有線クライアントへの該当するコンテンツの抜粋を送達する本発明のトランスセラプションの処理フローを詳述するフローチャートである。 20

【図 11】図 11 は、個々のデータ処理モジュールを詳述する無線デバイスまたはクライアント上に常駐する本発明を示すブロックダイアグラムである。

【図 12】図 12 は、ユニバーサルコンテンツストリーム (U C S) の形式におけるトランスセラプトされたデータを受信する無線デバイス上に常駐する本発明の処理フローを詳述するフローチャートである。

【図 13】図 13 は、ロードされた構造化ドキュメントまたは非構造化ドキュメント、あるいは、データストリームを有する本発明の処理エンジンのアーキテクチャを示す図である。

【図 14】図 14 は、ロードされたドキュメントに対する構成されたドキュメントオブジェクトモデル (D O M) の本発明における親および子のオブジェクトの関係を示す図である。 30

【図 15】図 15 は、本発明が用いられ得るサンプルの非構造化ドキュメント図である。

【図 16】図 16 は、本発明が用いられ得るサンプルの構造化ドキュメントの図である。

【図 17】図 17 は、ロードされた構造化ドキュメントに対して本発明によって利用される構成されたドキュメントオブジェクトモデル (D O M) の図である。

【図 18】図 18 は、テキストコンテンツ用の本発明のドキュメントオブジェクトモデルを示す図である。

【図 19 A】図 19 A は、テキストコンテンツのパラグラフを構成する際の本発明のドキュメントを示す図である。 40

【図 19 B】図 19 B は、図 19 A におけるパラグラフコマンドを示す構成である。

【図 20】図 20 は、スプレッドシート (s p r e a d s h e e t) ドキュメントタイプで通常見られるテーブルデータを含むロードされたドキュメント用のドキュメントオブジェクトモデルを示す図である。

【図 21 A】図 21 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線 / 有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。

【図 21 B】図 21 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線 / 有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。 50

【図 2 1 C】図 2 1 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線／有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。

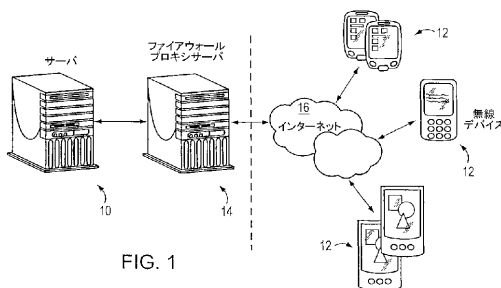
【図 2 1 D】図 2 1 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線／有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。

【図 2 1 E】図 2 1 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線／有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。

【図 2 1 F】図 2 1 A は、それぞれ構造化および非構造化ドキュメント用のユニバーサルコンテンツストリームデータ構造の形式での本発明から無線／有線デバイスへのトランスセラプト応答のコマンドコンテンツを詳述する表である。

【図 2 2】図 2 2 は、本発明の通信フォーマット用のレイアウトの一実施形態である。

【図 1】



【図 2】

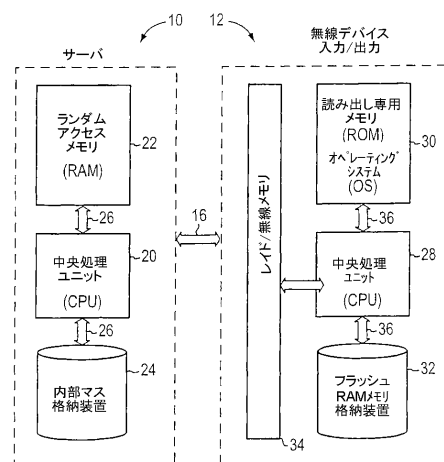


FIG. 2

【図3】

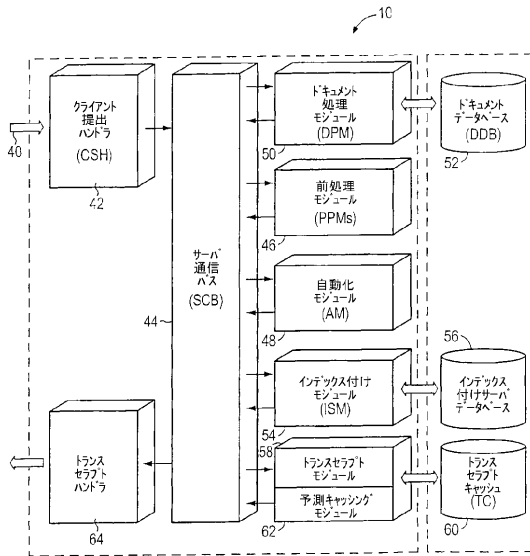


FIG. 3

【図4】

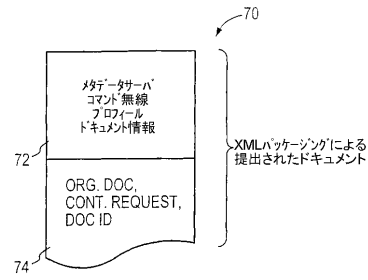


FIG. 4

【図5】

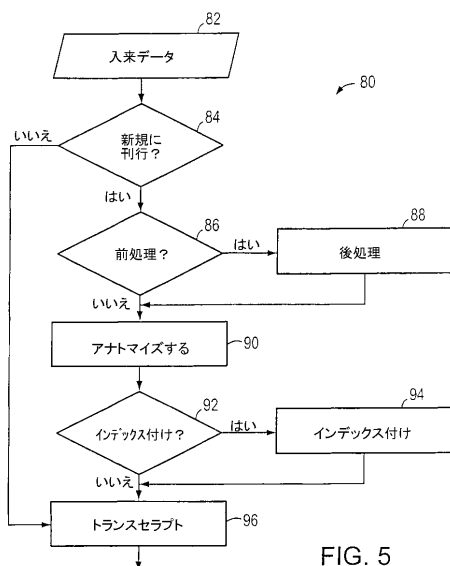


FIG. 5

【図6】

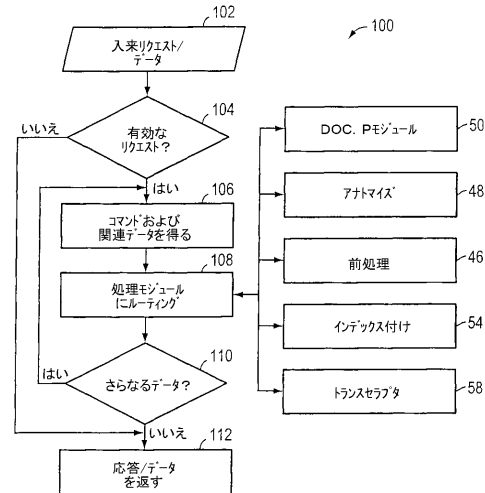


FIG. 6

【 図 7 】

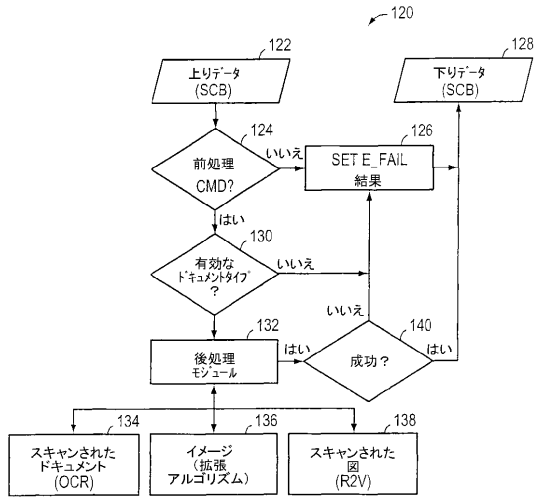


FIG. 7

【 図 8 】

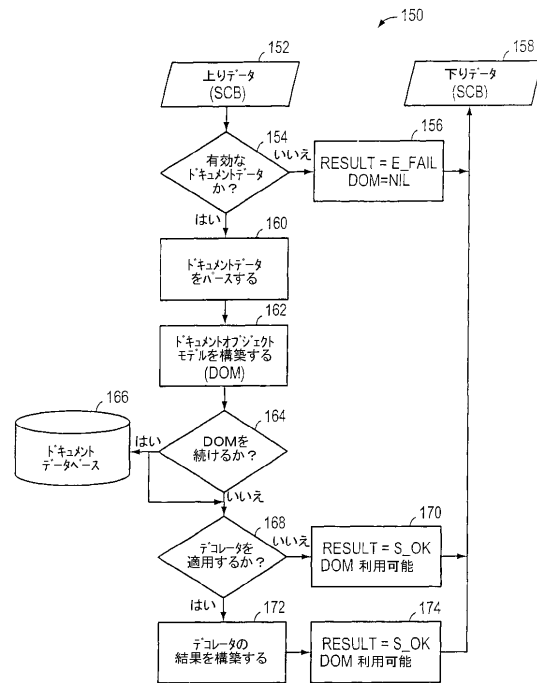


FIG. 8

【 図 9 】

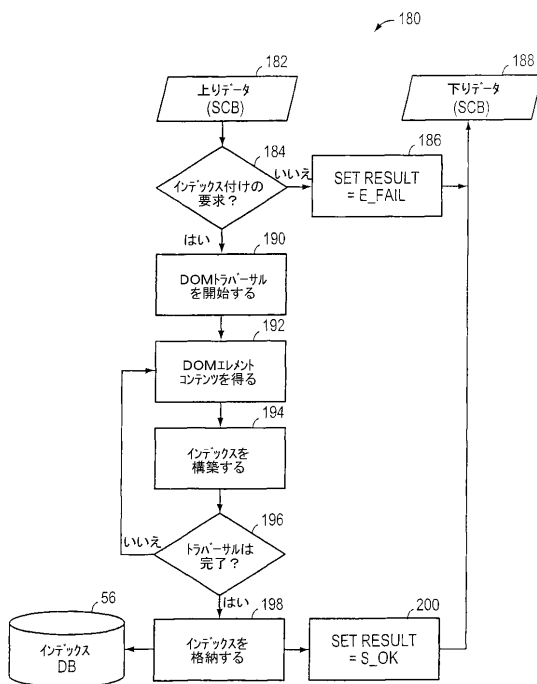


FIG. 9

【 図 10 】

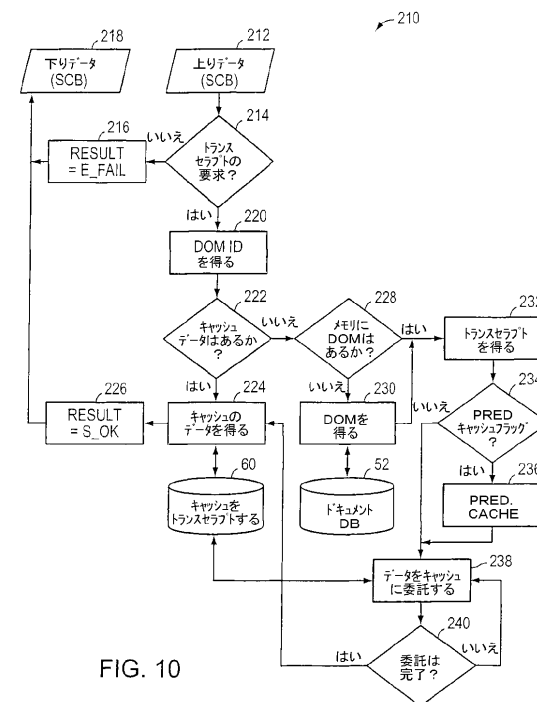


FIG. 10

【図 1 1】

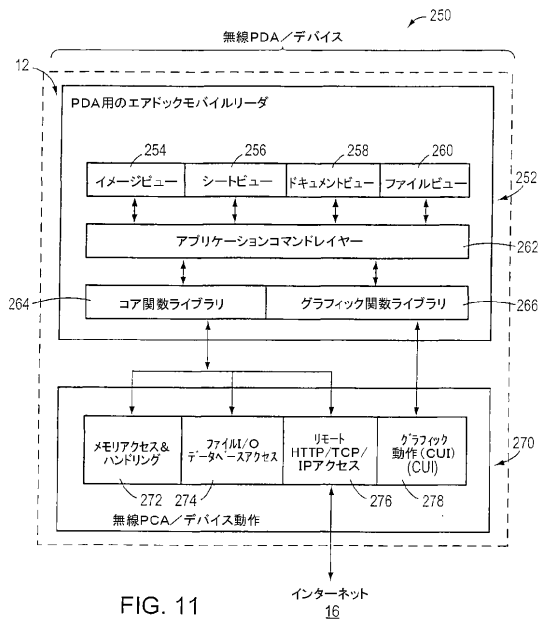


FIG. 11

【図 1 2】

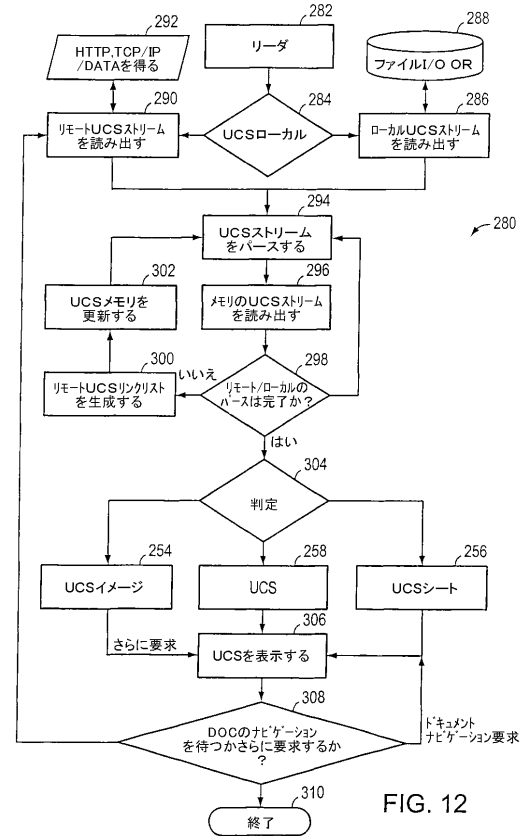


FIG. 12

【図 1 3】

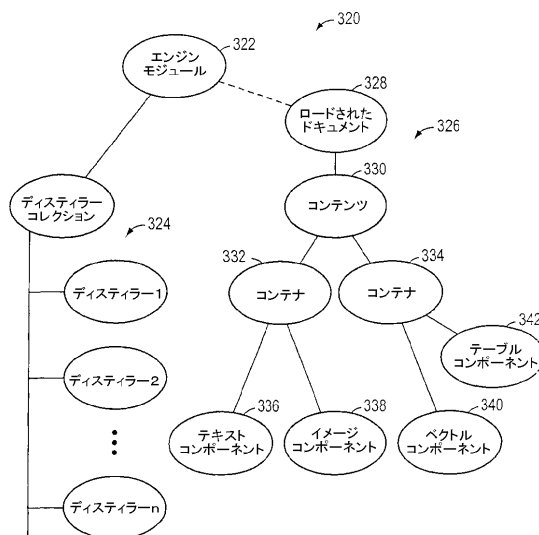


FIG. 13

【図 1 4】

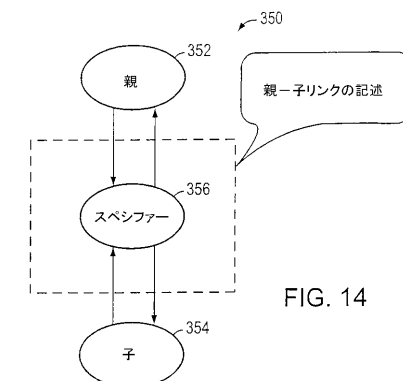


FIG. 14

【図 15】

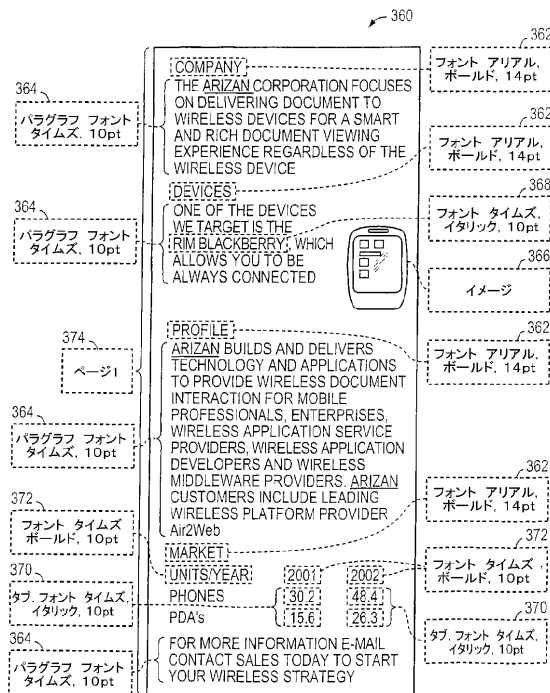


FIG. 15

【図 16】

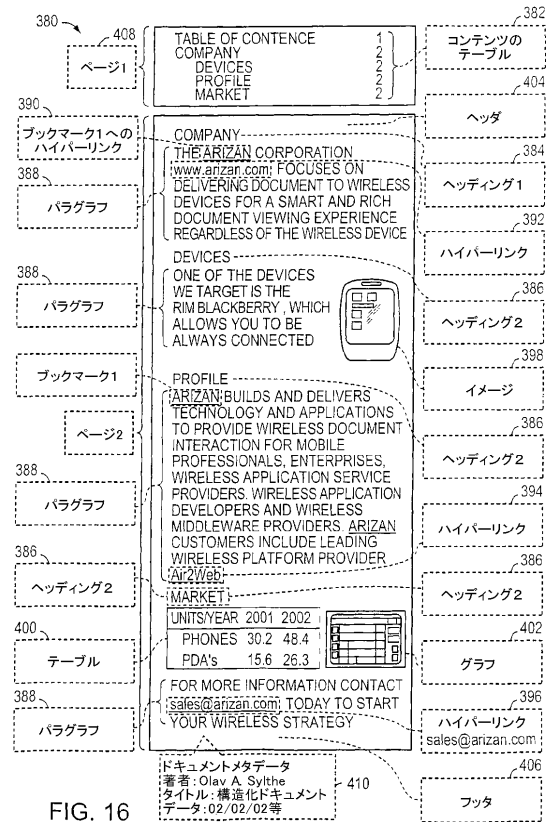


FIG. 16

【図 17】

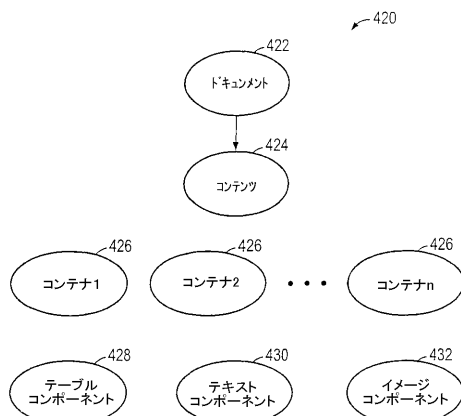


FIG. 17

【図 18】

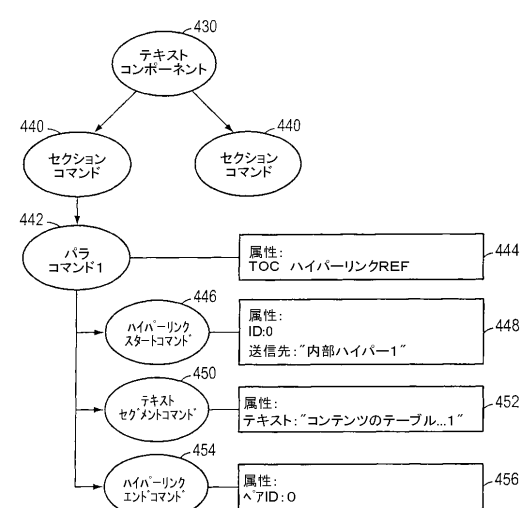


FIG. 18

【図 19 A】

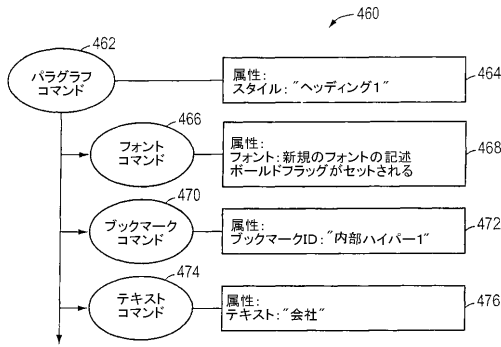


FIG. 19A

【図 19 B】

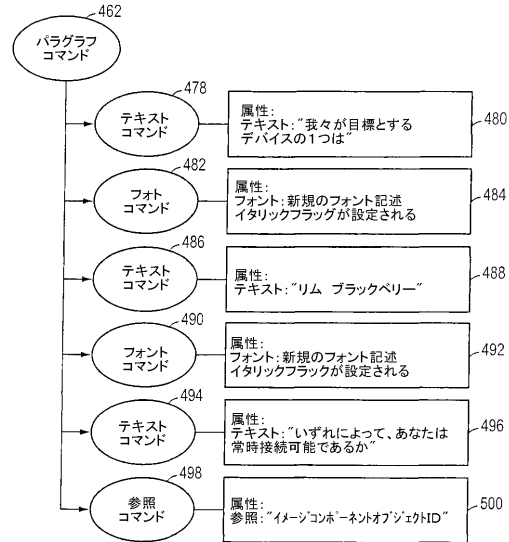


FIG. 19B

【図 20】

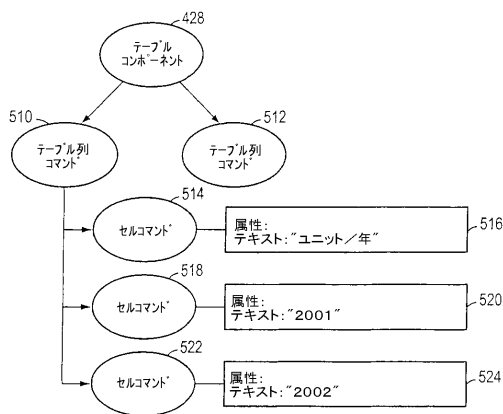


FIG. 20

【図 21 A】

記録 ID	構築されたDOC	記録 ID	記録ID構築されていないDOC
1	UCS SIGNATURE	1	UCS SIGNATURE
2	DOCUMENT HEADER	2	DOCUMENT HEADER
3	DOCUMENT CONTAINER	3	SECTION CONTAINER
4	PARAGRAPH	4	PARAGRAPH
5	PARAGRAPH INFORMATION	5	PARAGRAPH INFORMATION
6	BOOKMARK	6	BOOKMARK
7	SIMPLE FONT CHANGE	7	SIMPLE FONT CHANGE
8	TEXT SEGMENT	8	TEXT SEGMENT
9	PARAGRAPH	9	PARAGRAPH
10	PARAGRAPH INFORMATION	10	PARAGRAPH INFORMATION
11	PARAGRAPH	11	PARAGRAPH
12	PARAGRAPH INFORMATION	12	PARAGRAPH INFORMATION
13	HYPERLINK BEGIN	13	HYPERLINK BEGIN
14	SIMPLE FONT CHANGE	14	SIMPLE FONT CHANGE
15	TEXT SEGMENT	15	TEXT SEGMENT
16	SIMPLE FONT CHANGE	16	SIMPLE FONT CHANGE
17	TEXT SEGMENT	17	TEXT SEGMENT
18	TEXT SEGMENT	18	TEXT SEGMENT
19	HYPERLINK END	19	HYPERLINK END
20	PARAGRAPH	20	REFERENCE 1 - TEXT BOX
21	PARAGRAPH INFORMATION	21	PARAGRAPH
22	HYPERLINK BEGIN	22	PARAGRAPH INFORMATION
23	SIMPLE FONT CHANGE	23	HYPERLINK BEGIN
24	TEXT SEGMENT	24	SIMPLE FONT CHANGE
25	SIMPLE FONT CHANGE	25	TEXT SEGMENT
26	TEXT SEGMENT	26	SIMPLE FONT CHANGE
27	TEXT SEGMENT	27	TEXT SEGMENT
28	HYPERLINK END	28	TEXT SEGMENT
29	PARAGRAPH	29	HYPERLINK END
30	PARAGRAPH INFORMATION	30	PARAGRAPH
31	BOOKMARK	31	PARAGRAPH INFORMATION
32	HYPERLINK BEGIN	32	BOOKMARK
33	SIMPLE FONT CHANGE	33	HYPERLINK BEGIN
34	TEXT SEGMENT	34	SIMPLE FONT CHANGE
35	SIMPLE FONT CHANGE	35	TEXT SEGMENT
36	TEXT SEGMENT	36	SIMPLE FONT CHANGE
37	TEXT SEGMENT	37	TEXT SEGMENT
38	HYPERLINK END	38	TEXT SEGMENT
39	PARAGRAPH	39	HYPERLINK END
40	PARAGRAPH INFORMATION	40	REFERENCE 2 - IMAGE

FIG. 21A

【図 22】

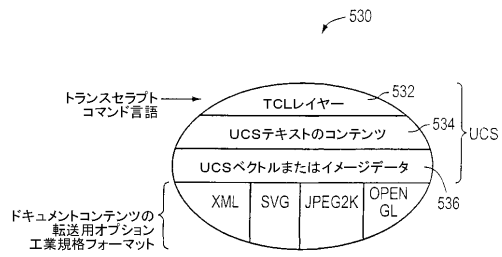


FIG. 22

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
3 October 2002 (03.10.2002)

PCT

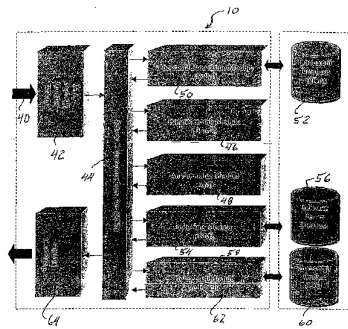
(10) International Publication Number
WO 02/077855 A1(51) International Patent Classification: **G06F 17/21**

(21) International Application Number: PCT/US02/09268

(22) International Filing Date: 25 March 2002 (25.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/278,137 23 March 2001 (23.03.2001) US(71) Applicant: **ARIZAN CORPORATION** [US/US]; 430
10th Street, N.W., Suite S-105, Atlanta, GA 30318 (US).(72) Inventor: **SYLTHE, Olav, A.**; 4867 Ashford Dunwoody
Road #6208, Atlanta, GA 30338 (US).(74) Agent: **WINBURN, John, T.**; Morris Manning & Martin,
L.P., 3343 Peachtree Road, N.E., Atlanta, GA 30326-1044
(US).(81) Designated States (national): **AF, AG, AI, AM, AT, AU,**
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,**CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,**
GM, GR, GU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,
LI, LU, LV, MA, MD, MG, MK, MN, MW,
MX, MY, NZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK,
SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.(84) Designated States (regional): **ARIPO** patent (GH, GM,
KI, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZM, ZW),
Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR,
GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), **OAPI** patent
(BJ, BJ, CI, CG, CL, CM, GA, GN, GQ, GW, ML, MR,
NI, SN, TD, TG).**Declaration under Rule 4.17:**
*as to the applicant's entitlement to claim the priority of the
earlier application (Rule 4.17(ii)) for all designations***Published:**
*with international search report**For two-letter codes and other abbreviations, refer to the "Guidance
Notes on Codes and Abbreviations" appearing at the beginning
of each regular issue of the PCT Gazette.*(54) Title: **SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO
A PORTABLE COMPUTING DEVICE**

(57) Abstract: A system which analyzes, organizes and stores electronic documents in a document object model using a common markup language. The document object model is stored on a server (10,14,50,52) and the content is delivered to the wireless device (12) in a universal content system format (162). By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for transmission.

WO 02/077855 A1

WO 02/077855

PCT/US02/09268

**SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS
COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE****CROSS REFERENCE TO RELATED APPLICATION**

5 This application claims priority under 35 U.S.C. 119 to the benefit of the filing date of
the U.S. patent application serial no. 60/278,137, which was filed on March 23, 2001,
entitled "SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS
COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE," which is
10 incorporated herein by reference.

BACKGROUND OF THE INVENTION**Field of the Invention**

The present invention relates generally to wireless content delivery of documents and
more specifically relates to organizing document content to be requested by reconstructed and
15 read on a wireless device.

Description of the Prior Art

With the rapidly expanding use of computers, electronic documents are stored and
communicated between computers with relative ease. Full desktop systems and laptop
20 systems readily transfer these documents over various communication media, such as the
Internet.

It is also desirable, however, to utilize these documents with the wireless
communications medium and utilizing mobile wireless devices. This presents a whole series
of different problems, since the bandwidth available can be on the order of 4 to 48 kbps,
25 which is not expected to increase past 56 kbps in the near term.

The wireless devices typically attempt to maximize battery life, while at the same
time minimizing weight and the device physical size. Typically, the devices also include

WO 02/077855

PCT/US02/09268

slow processors and a minimal amount of memory. At the same time, the devices have different, generally limited, operating systems, screen resolution and limited graphic rendering capabilities. Large documents with a wide variety of formats are therefore very unsuitable or impossible for use with these devices in their standard format.

- 5 It would thus be desirable to provide a system, which allows the wireless devices to have access to the electronic documents without regard to their size or formatting of the documents.

SUMMARY OF THE INVENTION

- The present invention provides a system, which analyzes, organizes and stores
10 electronic documents in a document object model using a common markup language. The document object model is stored on a server and the content is delivered to the wireless devices in a universal content stream format. By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for
15 wireless transmission.

To accomplish the above and related advantages, this invention may be embodied in the form illustrated in the accompanying drawings, attention being called to the fact, however, that the drawings are illustrative only, and that changes may be made in the specific construction illustrated without departing from the scope of the present invention.

- 20 **BRIEF DESCRIPTION OF THE DRAWINGS**

Various other functions, features and attendant advantages of the present invention will become fully appreciated as the same is described in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the several views, and wherein:

WO 02/077855

PCT/US02/09268

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is an illustration of a data processing system or server where the invention is employed to communicate with wireless devices of clients over the Internet.

Figure 2 is an illustration of a block diagram of the data processing system and wireless devices represented by **Figure 1**.

Figure 3 is a block diagram illustration of the invention residing on the server in **Figure 1** detailing individual process modules of the present invention.

Figure 4 illustrates the data packaging of a document submitted to the present invention.

Figure 5 is a flowchart illustrating the process flow of a submit request being processed by the data processing system on the server.

Figure 6 is a flowchart detailing the process flow of an incoming request being processed by the Server Communication Bus (SCB) module of the server data processing system.

Figure 7 is a flowchart detailing the process flow of a pre-processing module for invoking external processes for Optical Character Recognition (OCR), Image Enhancement and Raster to Vector conversion (R2V).

Figure 8 is a flowchart detailing the process flow of a data processing system Anatomization module used to construct the documents anatomy in establishing the Document Object Model (DOM).

Figure 9 is a flowchart detailing the process flow of an Indexing Module for indexing content of a submitted document.

Figure 10 is a flowchart detailing the process flow of a Transcription Module of the invention for delivering the relevant content excerpts to the wireless/wired clients.

WO 02/077855

PCT/US02/09268

Figure 11 is a block diagram illustrating the invention residing on a wireless device or client detailing the individual data processing modules.

Figure 12 is a flowchart detailing the process flow of the invention residing on a wireless device receiving transcribed data in the form of the universal content stream (UCS).

5 **Figure 13** is a diagram illustrating the architecture of the processing engine of the present invention with a loaded structured or unstructured document or data stream.

Figure 14 is a diagram illustrating the relationship of a parent and child object in the invention of a constructed Document Object Model (DOM) for a loaded document.

10 **Figure 15** is an illustration of a sample unstructured document with which the invention might be used.

Figure 16 is an illustration of a sample structured document with which the invention might be used.

Figure 17 is an illustration of the constructed Document Object Model (DOM) utilized by the invention for a loaded structured document.

15 **Figure 18** is a diagram illustrating a Document Object Model of the present invention for textual content.

Figure 19A is a diagram illustrating the inventions document object model in constructing a paragraph of textual content.

Figure 19B is a continuation of illustrating the paragraph command in **Figure 19A**.

20 **Figure 20** is a diagram illustrating the Document Object Model for a loaded document containing table data as typically found in a spreadsheet document type.

Figures 21A – 21F are illustrations detailing the command content of a transcript response from the invention to a wireless/wired device in the form of a Universal Content Stream data structure for a structured and unstructured document respectively.

WO 02/077855

PCT/US02/09268

Figure 22 is one embodiment of a layout for the communication format of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A data processing system or server **10** communicating with one or a plurality of wireless devices or clients **12** where the present invention can be utilized is illustrated in **Figure 1**. The data processing system **10** typically includes conventional components, such as a central processing unit, internal mass storage, and random access memory, which are utilized by the present invention. Communication to and from the data processing system **10** will typically come through a secure software or hardware processing system/unit commonly known as a Firewall/Proxy Server **12** in a conventional manner to ensure authorized and valid access to the network in which the invention is deployed. The invention executing on the data processing system or server **10** will typically receive inbound and transmit outbound data via the Internet **16** to a portion of the invention residing on the wireless device or client **12**. A current embodiment of the present invention can utilize as the data processing system **10** DELL, COMPAQ, SUN or other recognized manufacturers of server hardware running a Windows operating system release NT 4/2000/XP or higher or on a Sun Solaris operating system release 8 or higher.

Figure 2 illustrates a more detailed overview of the data processing system or server **10** and the wireless device or client **12**. The server **10** includes a central processing unit (CPU) **20**, a random access memory (RAM) **22** and an internal mass storage device **24** all interconnected by a data channel commonly known as a bus **26**. The data processing system **10** communicates with the wireless device client **12** over the Internet **16**. The wireless device client **12** processing system includes a central processing unit (CPU) **28** designed for small footprint devices such as a Motorola Dragon Ball CPU, or the Intel StrongARM CPU coupled with a read only memory (ROM) **30** which stores a small footprint wireless device

WO 02/077855

PCT/US02/09268

operating system such as Palm OS, or PocketPC OS with an internal flash ram memory storage 32 communicating with the Internet 16 utilizing a wireless radio/modem 34 all coupled together using a databus 36.

5 The data processing system 10 preferably is configured to execute and perform the functionality of the current embodiment of the present invention by having an operating system able to utilize data instructions written to execute using the functionality provided by the operating system to utilize typical programming languages such as C++ and Java.

10 **Figure 3** illustrates a block diagram of the process flow for the invention's server side processing modules of the present invention. A module by definition is an individual data process optional or obligatory that combined with other modules forms the invention's data processing system 10. A structured or unstructured electronic document or data stream 40 is submitted from the wireless or wired client side 12 and enters the server 10 from the network as binary data. A Client Submit Handler 42 receives the client request and verifies that this is a valid request and passes the received data with appropriate routing commands based on the client request over to a Server Communication Bus 44, which reads the routing commands and sends the received data to the required processing modules in the required order as indicated by the routing commands. If the first routing command is a request for a Preprocessing Module (PPM) 46 the received data is passed along for document or data optimization. The Preprocessing Module's 46 main purpose is to improve document quality for utilization by the invention if the document type is a scanned document containing imagery depicting textual content, containing binary imagery representing a picture, or a scanned document depicting a drawing using line or circular drawing elements.

20 For a scanned document containing imagery depicting textual content a technique known as Optical Character Recognition (OCR) is employed with the purpose of extracting the textual characters for processing by the invention. Any OCR style software package such

25

WO 02/077855

PCT/US02/09268

as OmniPage from ScanSoft Corporation can be applied by the invention for this purpose. For imagery representing a picture the invention can apply commonly known image algorithms for improving image quality such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. For a scanned document depicting a drawing using line or circular drawing elements raster to vector (R2V) software is employed for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct the submitted image as vector drawing data on the inventions wireless/wired client. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the invention for this purpose.

The Server Communication Bus 44 will then route the data 40 on to an Anatomization Module (AM) 48, which is responsible for establishing the anatomy of the document and to populate the Document Object Model (DOM) as described in Figure 17 and will as a dependant process utilize the SCB 44 to route the data to a Document Processing Module (DPM) 50 where an Engine as described in Figure 13 will load the structured or unstructured document using a distiller module. If the SCB's 44 routing commands include a command to persist (save) the document data parsed by the DPM 50 the document data will be saved in a Document Database (DDB) 52. The Document Database 52 is a connector to typical software packages commonly referred to as databases such as Microsoft SQL Server from Microsoft Corporation. Upon successful processing completion of the document by the DPM 50 the SCB 44 will issue a command back to the Anatomization Module 48 and the Document Object Model will be updated with the content and anatomy of the document. If an indexing request is part of the command stack of the SCB 44, an Indexing Module (ISM) 54 will be invoked and the textual content available from the Anatomization Module 48 of the document will be indexed using commonly known software products such as Microsoft

WO 02/077855

PCT/US02/09268

Indexing Server and the results along with the associated references to the Document Object Module will be stored in an Indexing Server Database 56.

Upon completion by the ISM 54, the SCB 44 command stack will invoke a Transcript Module (TM) 58 (transcript = transmit excerpt) and the initial excerpt of content will be prepared for transmission by invoking the Anatomization Module 48 and requesting an initial Table of Content or Summary of the document based on the content available in the Document Object Module. Upon return of the transcript by the AM 48, a copy of the initial transcript will be placed in a Transcript Cache (TC) 60 for repeat access purposes. If the SCB 44 command stack contains a Predictive Caching Module 62 request, the AM 48 will be repeatedly invoked for the lifetime of the client session with the purpose of predicting future client requests for related content data from the document. The Predictive Caching Module 62 will, for the lifetime of the client session, then populate the Transcript Cache 60 with the transcript results from the Transcript Module 58. The initial transcript is then upon completion from the TM 58 returned via the SCB 44 to a Transcript Handler 64 where the correct client device 12 for the return data is identified and the transcript is sent to the network 16 and back to the wireless/wired client 12.

If the Client Submit Handler 42 receives a client request for more content from a previously submitted document it passes the request over to the Server Communication Bus 44, which in turn hands it off to the Transcript Module 58, which will then look into the Transcript Cache 60 to see if a transcript generated by the Predictive Caching Module 62 is available for transmittal. If not, the Anatomization Module 48 will be invoked and a new transcript will be prepared following the procedure as described for documental submittal for the System 10. Upon completion the transcript will be placed in the Transcript Cache 60 and the Transcript Handler 64 again will be invoked and deliver the transcript to the client wireless/wired device 12 as described for the initial document submittal.

WO 02/077855

PCT/US02/09268

Figure 4 represents the packaging of a document or content request **70** submitted for publishing by the present invention. An XML based metadata description **72** is the encapsulation of the packaging and will contain the appropriate server command such as publish or query, wireless device profile of the requesting device **12** providing information such as device type, device screen size, free storage space, estimated bandwidth transfer speeds etc. and information on the document **70** such as document name, document date, document type or any other supplemental information. An Original Document or unique document ID **74** for the content request **70** makes up the other part of a document submittal request. If an original document is part of the submitted request the document will be published or republished if required. If a unique document ID is sent as part of the request it indicates that the document is already published and the server will look in the Transcript Cache **60** or retrieve the document from the Document Database **52** based on the unique Document ID.

Figure 5 illustrates a flowchart process **80** for a wireless client **12** submit request to the data processing system server **10**. An incoming request **82** with data has arrived. The first step of the process **80** is to establish if this is a new publish request **84**. If it has a unique document ID **74**, it is not a new publish request (NO) and the document already resides on the server data processing system **10**, then a Transcript processing step **96** will take place directly. If an original document is being submitted, the metadata **72** is processed to see if there is a client request to perform Pre Processing **86** and the PreProcess module **46** will be invoked in a step **88**. If not, the Anatomization process **90** will be performed. Upon process completion from the Anatomization process module **48** the metadata **72** is processed to see if there is a client request to perform indexing **92** on the anatomized content. If this is true, then the Index Processing Module **54** is invoked and the content will be indexed in a step **94**. Upon completion of step **92** and/or step **94**, the Transcript module **58** will be invoked in a

WO 02/077855

PCT/US02/09268

step 96 with the purpose of delivering the initial content result back to the requesting wireless client 12.

Figure 6 is a flowchart 100 illustrative the data flow of the server side data processing system 10 of the Server Communications Bus 44 (SCB). An incoming request and/or data is received 102 by the SCB and the SCB looks to determine if this is a valid request 104 for the SCB 44. If the request is valid (YES), the command and associated data is retrieved 106 by the SCB. Associated data will be an original document or original document unique ID. Pending on the SCB command the data will be routed 108 to the appropriate processing module where any single or combination of SCB commands will be performed by the server side data processing system 10 including any of the system modules; the Document Processing Module 50, the Anatomization Module 48, the Preprocessing Module 46, the Indexing Module, and the Transcriber Module 58. As the SCB 44 is designed to process both asynchronous or synchronous request requests for data processing by the individual data processing system modules 46, 48, 50, 54, 58 the requests may or may not be completed when the SCB 44 will check if there are additional commands and data to be processed in a step 110. If there are more commands or data (YES) the process will continue again with the step 106 until the command stack is emptied. If no more data or commands are pending in the step 110, the SCB 44 will return the appropriate response and resulting data from the server side data processing modules in a step 112 to the requesting wireless/wired client 12. If no valid command is detected by the SCB in the step 104 (NO) the appropriate failure code will be assigned by the step 112 and the resulting return or response will be sent to the requesting wireless or wired client 12.

Figure 7 illustrates a flow chart 120 describing the server side data processing system for the Pre Processing Module 76. Inbound Data 122 is received on the SCB 44. A validation step 124 is performed to see if this is a valid Pre-process command. If the

WO 02/077855

PCT/US02/09268

command is not recognized as a valid pre-process command (NO) the result error code is set to E_FAIL in a step 126 and the error code is returned as outbound data on the SCB in a step 128. If the step 124 determined that the command is a valid preprocess command (YES), a step 130 checks to see if the document type submitted with the preprocess command is a valid supported document. If the answer is negative (NO) the step 126 again sets the result error code to E_FAIL and the error code is returned as outbound data on the SCB in the step 128. If it is determined that this is a valid document type (YES) in the step 130, then the Preprocess module 46 is invoked in a step 132. Based on the document type determined as part of the successful step 130, an appropriate preprocessing sub module 134, 136 or 138 is invoked. Step 134 performs an Optical Character Recognition (commonly referred to as OCR) to convert a scanned document into textual content. The invention is designed to use commercially available software for OCR purposes from software vendors such as OmniPage from ScanSoft Corporation.

Step 136 is invoked by the step 132 if the valid document type determined in the step 130 is an image or picture data in saved form such as BMP (Bitmap) or JPEG (Joint Photographic Experts Group) upon which commonly known mathematical formulas can be employed to improve the visual aspects of the image picture data such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. A step 138 is invoked by the step 132 if the valid document type determined in the step 132 is a scanned drawing depicting a drawing using line or circular drawing elements. A step 138 will then employ commercially available raster to vector (R2V) software for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct or recreate the submitted image as vector drawing data on the wireless/wired client 12. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the present invention for

WO 02/077855

PCT/US02/09268

this purpose. Upon completion of any sub process 134, 136, 138 by step 132 then a step 140 will check if the process is determined to be successful. If the step 140 is a success (YES), the preprocessed data will be returned back to the SCB in the step 128. If the step 140 determines that the result from the step 132 is unsuccessful (NO), the result error code is set to E_FAIL in the step 126 and the error code is returned as outbound data on the SCB in the step 128.

Figure 8 is a flow chart 150 illustrating the process flow of the server side data processing system 10 for the Anatomization process 48. Inbound data from the SCB is received in a step 152, which is passed to a step 154 where it is determined if the document data is valid or not. If the document data is invalid (NO) a step 156 sets the result error code to E_FAIL and the pointer to the DOM is set to equal zero or NIL, and returns the error code as outbound data on the SCB in a step 158. If it is determined that the document data is valid (YES) in the step 154, the document data received will be parsed in a step 160 and the Document Object model (DOM) then is built up in a step 162. After the DOM has been constructed by the step 162, then a step 164 will check if there is a command request to persist or store the constructed DOM. If the step 164 is true (YES) the DOM data will be stored in the Document Database 52 in a step 166 before continuing on to a step 168. If the step 164 is false (NO), then the step 168 will check to see if there is a command to apply a decorator. A decorator is a command to traverse the Document Object Model established by the step 162 and extract content or navigational information such as a Table of Content (TOC), Bookmarks, Hyperlink, Find Text etc. If the step 168 has no decorator command to apply (NO), a step 170 will set the result code to S_OK and set the DOM pointer before returning the outbound data in the step 158. If, the step 168 determines that decorators are to be applied (YES), a step 172 is invoked and the decorators will traverse the DOM established

WO 02/077855

PCT/US02/09268

in the step 162. Upon completion of the step 172 the result code is set to S_OK and the DOM pointer is set in a step 172 before returning the outbound data in the step 158.

Figure 9 illustrates a flow chart 180 of the Indexing Process 54 for the server side data processing system 10. Inbound data from the SCB 44 arrives in a step 182. A step 184 checks if there is a request to index the data. If the step 184 is determined to be false (NO), then a step 186 will set the result code to E_FAIL and the result will be returned as the outbound data to the SCB in a step 188. If the step 184 is true (YES), then the system will start to traverse the Document Object Model (DOM) in a step 190. In a step 192 the content of a single DOM element is retrieved and then the system 10 will start to build the Index in a step 194. In the step 194 the system will typically employ commercially available software indexing packages such as Microsoft Indexing Server. After the content of the retrieved DOM element has been indexed in the step 194 then a step 196 checks to see if the DOM traversal is complete. If the answer is NO, then the steps 192 and 194 will again be processed until the DOM traversal is complete (YES) in the step 196. The index is then stored in a step 198 in the Index Database 56 and the result code is set to S_OK in a step 200 before the result is returned as the outbound data to the SCB in the step 188. The Index DB 56, utilized in the step 198 can be any commercial database product such as MS SQL Server or Oracle.

Figure 10 is a process flowchart 210 of the Transcription module 58 for the server side data processing system 10. Inbound data arrives from the SCB in a step 212 and the data is checked to see if this is a transcript request or not in a step 214. If it is not a transcript request (NO), then a step 216 will set the Result code to E_FAIL and pass the result back to the SCB in a step 218 as outbound data. If the step 214 determines that the data is a transcript request (YES), then a step 220 is invoked to obtain the ID of the Document Object Model. The DOM ID is used to identify the correct document either from the transcript cache 60 or from the Document Database 52. A step 222 checks to see if the data is available

WO 02/077855

PCT/US02/09268

in the cache 60. If the answer is YES, then a step 224 will get the data from the transcript cache 60 and a step 226 will set the result to S_OK and return the outbound data in the step 218 onto the SCB. If there is no data in the cache (NO), during the step 222 then a step 228 will see if there is a DOM loaded in the memory and if it matches the DOM ID.

5 Because the system 10 employs caching it will typically retain the DOM and DOM ID for a specified number of documents, such as the last 32 processed documents and if the DOM ID matches any of the DOM content in the cache it will be put back in memory unless it's already there, meaning it was the last document processed. If the test in the step 228 determines that there is no DOM in memory or the DOM ID does not match (NO) (meaning
10 the DOM is not in the cache) then a step 230 is invoked, which will retrieve the DOM from the Document Database 52 before proceeding with a step 232. The system 10 allows the DOM to be saved to a database or file for immediate retrieval without having to reprocess the original document used to create the DOM initially. If the step 228 determines that there is a DOM in memory and the Document ID matches (YES), then the step 232 is invoked to get
15 the transcript. A step 234 then looks to verify if the SCB command received in the step 212 as inbound data from the SCB has the Predictive Caching Flag set. If the flag is set (YES), then a step 236 will start the asynchronous Predictive Caching process 62 before proceeding with a step 238. If the step 234 determines that the flag is not set (NO), then the step 238 is invoked and the transcript data retrieved in the step 232 will be committed to the cache 60.
20 A step 240 will then test repeatedly to see if the commit is complete, returning back to the step 238 if the test results in a negative (NO). When the step 240 returns true for commit complete (YES) then step 224 is invoked and the transcript data is retrieved from the cache. The step 226 will set the result to S_OK and return the outbound data in the step 218 onto the SCB.

WO 02/077855

PCT/US02/09268

Figure 11 is a detailed diagram 250 of the present invention wireless client processing system residing on a wireless device or Personal Digital Assistant (PDA) 12. A reader 252 is a software application including a plurality of individual software function elements and libraries 254 through 266 that makes function calls to a Wireless PDA/Device operating system 270. The operating system 270 includes a plurality of individual software functions forming the operating system 270. Commonly known operating systems (OS) for wireless PDA/devices are Palm OS by Palm Corporation, Windows CE by Microsoft Corporation, J2ME by Sun Corporation, EPOC by Symbian and BREW by Qualcomm. The inventions client side software functionality is designed to read the defined Universal Content Stream, also commonly referred to as UCS. The reader 252 has the Graphic Function Library 266 that offers a set of graphical instructions used to render textual, vector and image-based data on the device 12. To accomplish this the library 266 uses available graphics operations 254 from the device OS 270. The Core Function Library 264 offers a set of functions derived from the device OS 270 for the memory access and handling 272, the file input/output and/or database access 274, and the wireless data access 276 for retrieving data from the Internet 16. The reader 252 then has the application command layer 262 that invokes the core library 264 and the graphic function library 266 to be able to offer a graphical user interface GUI as defined in the view modes 254, 256, 258 and 260.

Based on the data loaded from the UCS the Image View 254 will display image or picture based data, the Sheet View 256 will display spreadsheet data in a spreadsheet style grid, the Document View 258 will display textual content in a rich and intuitive way and the File View 260 will display a user interface for loading and saving the UCS file on the device 12. The layered design in the reader 252 is designed so that only the core function library 264 and the graphics function library 266 will be different on each wireless/pda device 12 on which the client side processing system is deployed on. This allows the invention to be

WO 02/07855

PCT/US02/09268

quickly moved or ported to different operating systems such as Palm OS, Windows CE, J2ME, EPOC and BREW where the two primary programming languages are Java and C++ as commonly know by those skilled in the art of programming. A USC binary file is delivered to the device 12 from the server side data processing system 10 over the Internet 16 and the wireless client reader 252 retrieves this data by using the functions for wireless data access (load) 276 in the core function library 264, which in turn calls the library for remote HTTP/TCP/IP access 276 of the wireless PDA/device OS 270.

Figure 12 is a detailed flowchart 280 of the client side processing application residing on the wireless PDA/devices 12. In a step 282 the reader 252 is invoked either by being launched by another application on the device 12 such as the Inbox application or by the user selecting the application icon for the reader and invoking it. In a step 284 the passed Universal Content Stream (UCS) if any, will be verified to be local or remote. If local then the UCS is already stored on the device 12 and a step 286 will read the local UCS stream from a file or database 288. If the UCS is determined to be remote in the step 287 (not residing on the device 12) then a step 290 is invoked to read the remote UCS stream. A step 292 will then obtain the remote UCS residing on the server side data processing system 10 using the Internet 16. Upon completion of either step 286 or 290 the execution will continue with a step 294 where the retrieved UCS data is parsed. A step 296 will then load the data in device memory and then a step 298 will test to see if the parsing of the remote or local UCS data is complete. If the step 298 is not complete for local data (NO) then the step 294 will be invoked again until the step 298 is true (YES) for local UCS files. If the step 298 for remote UCS parsing is not complete (NO) then a step 300 will create a remote link list for any UCS data described to still reside on the server 10 for any future transcripts to the client. Then a step 302 updates the UCS memory with the remote links and continues with the step 294 until the remote parsing step 298 is complete (YES).

WO 02/077855

PCT/US02/09268

After the UCS data has been parsed and is loaded into memory, then a step 304 determines the view type or combination of view types for the UCS data. Depending on the view type either Image View 254, Document View 258, or Sheet View mode 256 is invoked and then the UCS data is displayed or rendered in a step 306. After displaying the UCS data

5 in the step 306 the client data processing application waits for user input in a step 308. If the user navigates within the loaded document using menu options, or track/scroll wheel, stylus or keypad then the step 308 will be repeatedly invoked until the user terminates the reader 252 by choosing the exit option and a step 310 will exit the mobile reader and return to the invoking application prior to the step 282. If the user selects or invokes a more request in the

10 step 308 for additional UCS data, then the step 290 will be invoked again with the same following process flow as previously described until the user chooses to terminate the application in the step 308 and then the step 310 will exit the mobile reader and return to the invoking application prior to the step.

Figure 13 is a diagram 320 representing the document engine structure with a

15 hypothetical document loaded for the document-processing module 50. The composition of the document engine is a collection of individual components that are categorized as the following modules, an Engine Module 322, a set of Distiller Modules 324 and a loaded document 326 and Decorator Modules (not illustrated). The distiller module 324 functionality is mainly to handle different document types. This set of modules is responsible

20 in particular for parsing a known document format, constructing the document structure and, if desired, persisting the document structure back to a known document format. The set of distiller modules 324 is open in the sense that new distiller modules can be added or removed at runtime to accommodate the need for a flexible environment. So the set of supported document types can be expanded or reduced as required. The engine module 322

25 functionality is mainly to host the running distiller modules 324 and, if desired, the loaded

WO 02/077855

PCT/US02/09268

documents 326, offering a uniform environment for these. It is also used to provide common behavior for clients (e.g. loading a document). A plurality of the document structure modules are the building pieces of a loaded document. They include: a document node 328 (representing the loaded document and containing all other nodes), contents nodes 330 (representing the contents with a certain detail degree of a document), a pair of container nodes 332, 334 (representing containers for other nodes), a text component 336 (representing textual content), an image component 338 (representing image content), a vector component 340 (representing vector content), and a table component 342 (representing table data such as spreadsheets). The document structure 326 is a tree-like structure (with the possibility of sharing of nodes, so the more appropriate description would be graph-like). This fact allows the user to practically construct documents in any fashion desired without being constrained by a rigid document structure.

This is achieved by the fact that each node implements collection like methods (for children and/or parents). Behavior like rendering can be attached to the node itself or be offered as an external component. This is facilitated by the existence of a set of standard iterators of the document structure. The choice between internal and external behavior can be taken by taking into consideration and the requirements of the intended operation (speed, size, deployment etc.) The decorator modules functionality (not shown in Figure 13) is mainly to perform customized selected operations on the document structure. Examples include the extraction of the table of contents, list of bookmarks or hyperlinks, find operations etc. It is here where the transcripion technology is employed. They are separate modules, with their own framework, that operate completely independent of the other engine components. In fact the engine can operate without the decorator modules being installed. They just add more functionality. The decorator modules framework is composed of a decorator manager (which is by itself a component) and a set of installed modules that

WO 02/07855

PCT/US02/09268

perform certain operations. The individual modules have to register in a predefined way in order to integrate into the framework and be recognized by the decorator manager. A decorator is a software component as commonly referred to by those skilled in the art of software development designed for the sole purpose of navigating the Document Object

5 Model (DOM) to retrieve specific content and/or navigational information. In other words it can be stated that a decorator component constitutes a command. There are no limitations on how many decorators can be used to navigate the DOM. Typical decorators (commands) are: retrieve bookmarks in a document, retrieve hyperlinks in a document, retrieve a Table of Content for a document, construct a summary for a document if no TOC is available, retrieve

10 a specific page, find textual content within a document, retrieve specific font styles or information, retrieve an image or illustration within a document, etc.

Figure 14 illustrates the relationship 350 of a parent 352 and a child node 354 within the document engine 320. The actual link between the parent 352 and the child 354 is a little more complex than a pointer between the two. Between them there is a so-called specifier

15 356 object whose role is to hold attributes of the link. By attributes of the link typically is meant information on the parent or children object IDS that are associated since this is a graph like structure and a single parent can have multiple children and a child can also have multiple parent objects.

The operation of the invention is to take provided input in the form of an electronic

20 document structured or unstructured or data streams which embodies textual and non textual content typically represented by graphical imagery such as pictures or illustrations and drawing data represented in the form of vectors or commonly known as descriptions on how to draw a line from position A to B. Electronic documents are best described but not limited to business or personal correspondence letters, product information and representation

WO 02/077855

PCT/US02/09268

thereof, parts manuals, corporate or personal financial information and such like. **Figures 15** and **16** represent respectively a unstructured document and a structured document.

Figure 15 illustrates how a typical unstructured document **360** might look like. By unstructured is meant that the document has minimal navigation information and structured elements such as headers, footers, tables, and font style use upon which the invention can directly extract such format and navigational information for the purpose of delivering and reconstructing partial document content back to the invention's client side data processing system **250**. The server side **10** data processing capabilities then have to be employed to construct such navigational and reconstructing information by analyzing the unstructured document data.

The analyzation is done by detecting changes in font size, font type, font style and font positioning. By analyzing this all textual content elements **362** will be marked as headers by the following criteria's: largest used font size =14, font style = Bold, font type = Arial. This is reinforced by a text paragraph definition **364**, which all have font size =10, no font style and font type = Times. By identifying the headers **362** based on this information the invention can construct a navigational aid in the form of a Table of Contents where none exists for the unstructured document. An embedded image **366** will have the image insertion point coordinates x, y as well as a width and height information all in pixels. The system **10** of the invention can then provide a link to the image **366** when rendering this on the client side application processing system **12** and element **368** which is marked out because of a font style change within a text paragraph **364** will also be marked as a header, but ranked with a lower importance than the other headers **362** as it is positioned within the text paragraph **364**. The element **368** will then be used by the server side **10** application processing when creating a navigational aide in form of a Table of Contents as the element **368** is deemed to have navigational significance. Likewise elements of a type **370** and **372** will be detected by the

WO 02/077855

PCT/US02/09268

invention's server side processing capabilities as the fonts positioning of these elements are in close proximity in addition to the usage of the font styles Bold and Italic and finally usage of a Tabulator (TAB) character for spacing. These all indicate that this section has significance to the document 360 and can be marked as a navigational aide point when construction a Table of Contents. Finally a page element 374 defines a page number description that is associated as a known navigational index reference for all the unstructured content residing on the documents page 1.

Figure 16 illustrates a structured document 380 that can be processed by the server side data processing system 10 with the purpose of identifying a document navigational structure in the form of a Table of Contents and defining content excerpts with the intent to reconstruct it on the client side processing application 12. A Table of Contents (TOC) 382 is defined as an existing element in the structured document. Content elements 384 and 386 referenced as Heading 1 and Heading 2 are defined as paragraph headers 388 and are used as navigational elements in the construction of the TOC entity 382 by the document itself. The server side data processing application 10 can therefore directly extract these elements without the processing required for an unstructured document. Further the System 10 will be parsing the structured document to identify a hyperlink 390 to a bookmark 1. A hyperlink is a navigational link between to objects or entities within a structured document indicating that they are connected and related and that by interacting with the hyperlink the user can navigate to a different part of the document. The system 10 will make a record of the hyperlink entity 390 and use it for navigational purposes. A bookmark entity 392 is another element defined within the structured document 380 itself that provides a navigational anchor point that the system 10 will make a record of for the purpose of allowing the execution of the navigational hyperlink entity 390. Hyperlink entities 392, 394, 396 are different from the hyperlink entity 390 by being external hyperlinks. External hyperlinks are defined as link

WO 02/077855

PCT/US02/09268

objects pointing to other content or external applications or processes defined within the structured document 380 itself.

One hyperlink 392 will launch an Internet Browser with the address pointing to a website address www.arizan.com, or second hyperlink 394 to www.air2web.com, while the link 396 will launch an email application program with the "To" address set to sales@arizan.com. The server side data processing system 10 will, while processing the structured document, record these elements in the constructed Document Object Model for navigational or possible application launch purposes for software applications employing the invention. An Image element 398 in the structured document 380 provides information on an image/picture and position x, y and width, height information that the System 10 invention will record in the DOM thus giving the client component 12 the ability to reconstruct this element on a wireless device 12 in the right visual and content context. A table element 400 as defined by the structured document 380 contains content displayed and organized in a table like fashion. The invention's server side data processing functionality will read this information and map it when constructing the DOM utilizing the Document Engines table component to allow the invention's client side processing system to reconstruct this in a table like structure on a wireless device/pda 12.

A graph element 402 in the structured document contains information in a graph type format such as bar chart and links to the data provided in the table element 400 used when drawing the chart by the application reading the structured document such as a word processor like MS Word by Microsoft Corporation. The server side data processing capabilities will retrieve and record the graphs link and rendering information when processing the structured document for the intent of having the client side data processing system reconstruct the graph on a wireless device/pda 12. A header element 404 defines page header information for each individual page in the structured document, while a footer

WO 02/077855

PCT/US02/09268

element 406 defines page footer information that will be reused for each single page in the structured document 380. The server side data processing capabilities will record the elements 404 and 406 and construct the appropriate entries in the DOM. A page entity 408 also provides a navigational index while navigating the structured document 380 as each page entity is a sequential entry that allows the user to look up an individual page within the page range starting with start page and ending with last page. The structured document as represented by Figure 16 also embeds non-visual descriptive information about the document referred to as metadata 410. The invention's server side data processing capabilities are able to extract this information and store it in the DOM where it can later be used by the system 10 to provide document information such as document name, author, date, time created etc. as defined by the metadata stored in the structured document.

Figure 17 illustrates the Document Object Model 420 as constructed by the invention's server side data processing system 10. The illustration 420 represents the load document's general structure 422. The DOM's entry point is a content node 424. Each of a plurality of containers 426 can have multiple components of different types. The invention server side data processing application has four component types defined: a table component 430, a text component 431, an image component 432 and a vector component (not illustrated). Each component can also be referenced by other components under the same container 426 or different containers in the form of a reference command

Figure 18 illustrates the text component 430 as employed by the server side data processing system 10 after constructing the Document Object Model (DOM) 420 and after loading the structured document 380 illustrated by Figure 16. Each text component 430 can have more than one section command 440, and as typically found for any standard structured or unstructured document, multiple paragraphs 442 can be included in one single section 440. The paragraph object 442 has one attribute 444 defined, which is a reference to a Table Of

WO 02/077855

PCT/US02/09268

Content hyperlink. A hyperlink start object 446 itself is defined by its attribute object 448, which describes the hyperlink ID and the destination of the link. Then a text segment object 450 has a describing attribute 452 that holds the textual content itself for the text object 450. At the end of the paragraph object 442 a Hyperlink End object 454 is constructed with an attribute object 456 containing the Hyperlink pair ID.

Figure 19A illustrates a Document Object Model 460 of a loaded structured document 380 and the DOM composition for a paragraph as employed by the server side data processing system. A paragraph object 462 has an attribute object 464 that describes the associated style and the textual content of the paragraph 462. A Font object 466 contains a font command and an associated attribute object 468 contains the font description and the font style information. The next command object linked to the paragraph object 462 is a Bookmark object 470 and it's descriptive attribute object 472, which contains the Bookmark ID. Next the paragraph object 462 is linked to a text object 474 and it's associated attribute object 476 with the value of the textual content itself. Figure 19B then continues the DOM paragraph composition 460 started by Figure 19A. The next command object linked to the paragraph object 462 is a text command object 478 and an associated attribute object 480 containing the text value. Next is a new font command object 482 and an associated attribute object 484 containing the font type and style definition. Then the following objects linked to the paragraph object 462 are respectively text command and font command objects and related attributes 486 - 496 until the paragraph composition ends with a reference command object 498 and an associated attribute object 500 containing the reference value.

Figure 20 illustrates the table component 428 as employed by the server side data processing system 10 after constructing the Document Object Model (DOM) and after loading the structured document 380 illustrated by Figure 16. Each table component 428 can have multiple row objects 510, 512, in which there might be one or more cells contained.

WO 02/077855

PCT/US02/09268

Figure 20 illustrates the first row of the sample table in the sample structured document 380. The table row command object 510 links to a cell command object 514, which has an associated attribute object 516 containing the textual content value of the cell. Additional cell command objects 518, 520, 522 and 524 will then be linked to the table row command object 510 until no more data is available for that row.

Figures 21A - 21F illustrate the server side data processing systems 10 description of content output in the form of a Universal Content Stream (UCS) as employed by the invention to transport data in a effective and optimized way for wireless delivery of the data over the Internet 16 to be processed by the client side data processing system 12. Figures 21A - 21F has two tables each representing the server side processing capabilities in outputting a UCS data file representing commands to reproduce a structured document 380 and a unstructured document 360 as represented respectively in Figures 16 and 15.

Referring to Figure 22, a layout of one UCS format 530 is illustrated. The UCS (Universal Content Stream) format 530 provides the mechanism and data transport layer between the server 10 and the client technology or device 12. The format primary purpose is to enable only the required visual image being sent in an optimal fashion over the available bandwidth. The format 530 supports the transfer of graphical images/pictures, 2D/3D drawings and various textual data such as word processor, and spreadsheet content.

The UCS can include several different standard formats for information transport in addition to the transcript command and data transfers to utilize the transcript technology and any available bandwidth. Embedding optional standard formats within UCS opens up for easy integration with client technology that do not require transcription support. And for formats such as image/raster data the existing/coming standards such as JPEG 2000 will offer adequate support for compression without data loss from the server 10 to client 12.

WO 02/077855

PCT/US02/09268

Examples of the industry standard formats that can be accessed by both the server 10 and other clients directly are the following four types:

- XML for textual, metadata and content information
- SVG for vector information
- 5 • JPEG 2000 for images
- OpenGL 3D representation

Other standard formats or proprietary formats can be easily added or removed through the plugable and scalable architecture of the component based server 10.

The UCS format 530 includes three different data layers that together with optional industry standard formats make up the UCS format 530. The three proprietary layers are: a TCL layer, a UCS textual content layer 534, and a UCS image or vector data layer 536.

The TCL (Transcript Command Language) layer 532 provides a data channel for two-way communication between the server 10 and the client 12. The client 12 sends requests to the Client handler on the server side and receives a response with an transcript data stream in return. The TCL handles the following types of commands: Transcript requests for visual images, VQ (View and Query) commands, directory information, published document search/retrieve, and any required third party synchronization information. The TCL language itself is also constructed using XML for easy adaptation and future expansion.

20 The UCS vector stream 536 is a highly optimized vector format to provide vector-drawing capabilities on the thin client software. This enables the thin client software to offer vector representation for higher quality and accuracy on even small PDA's or handheld devices. The vector stream transfer and rendering is controlled through the TCL layer 532 which ensures that the Transcript handler will only generate a vector representation for the
25 clipped viewable area requested by the client 12 and also to render it in an optimal fashion for

WO 02/077855

PCT/US02/09268

the limited client display area that generated the request. The UCS format 530 is designed in such a fashion that new proprietary layers or formats can easily be added. Any of the formats used for transportation of images, vector and text can also be replaced or added depending on integration requirements by third parties.

5 The Universal Content Stream (UCS) 530 is a binary file format specification, which enables delivery of original document content including, but not limited to textual content, image data, vector data and navigational information for the processed document. The UCS format is very efficient in overcoming the current limitation of the wireless devices 12 in respect to available bandwidth, which today in general is in the range of 9.6 to typically 48
10 although higher speeds are theoretically possible. The UCS addresses the limited bandwidth problem in a number of ways.

The UCS format 530 allows original document file sizes to be reduced by up to ninety percent (90%) while retaining all essential content and necessary format information for reconstructing the document on wireless devices using the client side processing capabilities.
15 This reduction in size is achieved by stripping out non-essential formatting information that is used by the original document application, e.g. MS Word for rich presentation on desktop computer. Non-essential formatting information might be use of different font styles, font colors, and revision information, etc. which typically have no value on the wireless device 12 as the device screen might be monochrome or grayscale and the device typically has only one
20 font type. The UCS file format also allows server side compression of the UCS file and client side decompression for further reduction in size. For compression common available technology in the form of compression algorithms such as LZW can be utilized. In addition the server side processing capabilities in producing transcripts (excerpt of content for transmission) allows the generation of multiple UCS files for individual transmission from
25 the server to the client in an iterative process where the resulting UCS file sent to the wireless

WO 02/077855

PCT/US02/09268

device is a result for a specific content query from the user. The iterative query process from the user on a wireless device requesting content from the server might typically result in the delivery of for example ten individual UCS files delivered to the device. The UCS file format includes information on how these individual UCS files link together so that the original document can then be reconstituted by the individual files in memory for a coherent viewing or reading experience. The UCS format also contains the Document Object Model (DOM) ID to ensure that the server 10 processing capabilities can delivery the correct DOM information to the requesting device 12. In addition the UCS file format also contains Internet hyperlinks allowing the client to request more content from within the document.

10 Command codes can also be embedded in the UCS format to facilitate the execution of specific processes by the server side data processing capabilities. This is typically a request for specific content, lookup in the navigation information stored within the DOM, etc.

WO 02/077855

PCT/US02/09268

DEFINITION OF TERMS

1. Anatomization – A specific technology process for parsing and analyzing a structured or unstructured document or data source with the purpose of establishing a uniform Document Object Model containing all content and a navigation document hierarchy.
- 5 2. Bookmarks – A navigational link associating a navigational reference from position X to position Y within a document.
3. DOM – Document Object Model (DOM), a programming interface specification being developed by the World Wide Web Consortium, defines how to create and structure XML based documents as objects.
- 10 4. External File – reference to an associated external data or document file outside the original document.
5. External Object – reference to an associated external data object outside the original document.
6. Font Color – The color of the font.
- 15 7. Font position – The rendering x, y pixel or character positioning information.
8. Font Size – Font size in points indicating the size of a font character.
9. Font Style – The style of the font as in normal, italics, bold, underline, etc.
10. Font Type – Name of font type used in document e.g., Times Roman.
11. Footer – Footnote at the bottom of an individual or all pages in a document
20 containing content.

WO 02/077855

PCT/US02/09268

12. Header – Header at the top of an individual or all pages in a document containing content.
13. Hyperlinks – A navigational link external or internal to the document content referencing a bookmark or external data or content source.
- 5 14. Page – Defines an individual age in a document containing content information typically separated by a page break character of a fixed page or printer page size definition.
15. Paragraph – Collection of textual characters or lines of text grouped together in a cohesive body separated by line breaks.
- 10 16. Predictive Caching – A specific technology process designed to cache relevant or associated content on the server prior to required delivery/transmittal to a wireless/ wired device to minimize bandwidth and network latency.
17. Table of Content (TOC) – Presents a hierarchical navigational structure to the document content for informing and assisting in navigation of the document content.
- 15 18. Title Styles – Predefined styles in a document centric creation application to allow for the construction of a Table of Content such as Heading 1, Heading 2, Heading 3, etc.
19. Transcription – A specific technology process for delivering only the requested content excerpt and transmitting this to the client.
- 20 20. USC – The Universal Content Stream is a content data format optimized for minimized data size while preserving content context, navigational and presentation information for delivery to wireless/wired devices.

WO 02/077855

PCT/US02/09268

21. XML -- extensible Markup Language is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

WO 02/077855

PCT/US02/09268

CLAIMS

1. A method for delivering content to a portable computing device, comprising:
receiving a request from the portable computing device for content associated with a
document remote to the portable computing device;
5 identifying one or more portions of the document associated with the requested
content;
reconstructing the identified portion or portions of the document from elements of the
document stored in a Document Object Model (DOM) Storage; and
transmitting the reconstructed identified portion or portions to the portable computing
10 device for display thereon.
2. The method of Claim 1, wherein identifying one or more portions of the
document includes utilizing a lookup or navigation tree to identify one or more portions of
the document.
3. The method of Claim 1, wherein identifying one or more portions of the
15 document includes utilizing one or more probability algorithms to determine what are the
most relevant portions of the document to the requested content.
4. The method of Claim 1, wherein reconstituting the identified portion or
portions includes reconstructing the identified portion or portions in a predetermined format
compatible with the portable computing device.
- 20 5. A method for publishing a document content on a remote portable computing
device, comprising:
receiving the document to be published, wherein the document includes visual
document representation and document content;
separating the visual document representation and the document content;
25 indexing the document content; and

WO 02/077855

PCT/US02/09268

establishing navigational links, which are mapped to a navigational tree; and storing the document content as Document Object Model structures.

6. The method of Claim 5, further including generating a metadata header for the document.

5 7. A method for wireless delivering document content to a portable computing device, comprising.

analyzing the content of a structured or unstructured document by identifying existing navigational elements or where none exist by analytical processing based on the documents visual representation to establish navigational links for the document content;

10 receiving a request from the portable computing device for content in said document;
using said navigational links to find and transmit said requested content; and
receiving said content at said portable computing device and reconstructing said content on said device.

8. The method of claim 7, including creating a table of contents during said
15 analysis and said portable computing device utilizing said table of contents to request said content.

9. The method of Claim 7, including storing said content in a document object model and obtaining said requested content from said document object model.

10. The method of Claim 7, including transmitting said content in a universal
20 content stream to said portable computing device.

11. The method of Claim 7, wherein analyzing said document includes anatomizing said document.

12. The method of Claim 7, including transmitting content excerpts of said document to said portable computing device.

WO 02/077855

PCT/US02/09268

13. The method of Claim 12, including predictive caching of said content excerpts relating to said content request and said content excerpt previously sent.

14. A system for wireless document content delivery to a portable computing device, comprising:

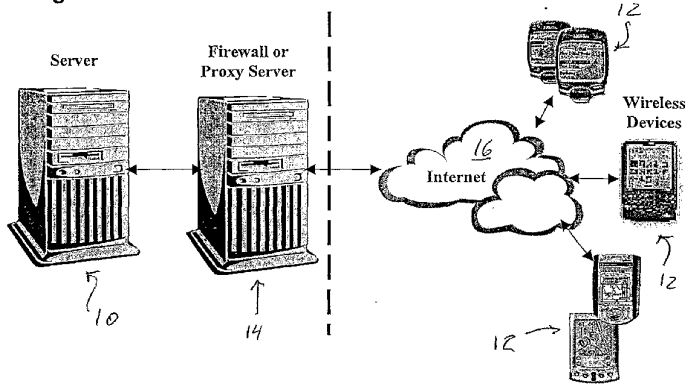
- 5 a communication bus having an input and output coupled thereto.
a document processing module coupled to said bus and to a document data base;
a preprocessing module coupled to said bus;
an anatomizing module coupled to said bus;
an indexing module coupled to said bus and to an indexing database; and
10 a transcript module coupled to said bus and to a transcript cache.
15. The system of Claim 14, including a predictive cache coupled to said transcript module and to said bus.

WO 02/077855

1/28

PCT/US02/09268

Figure 1



WO 02/077855

2/28

PCT/US02/09268

Figure 2

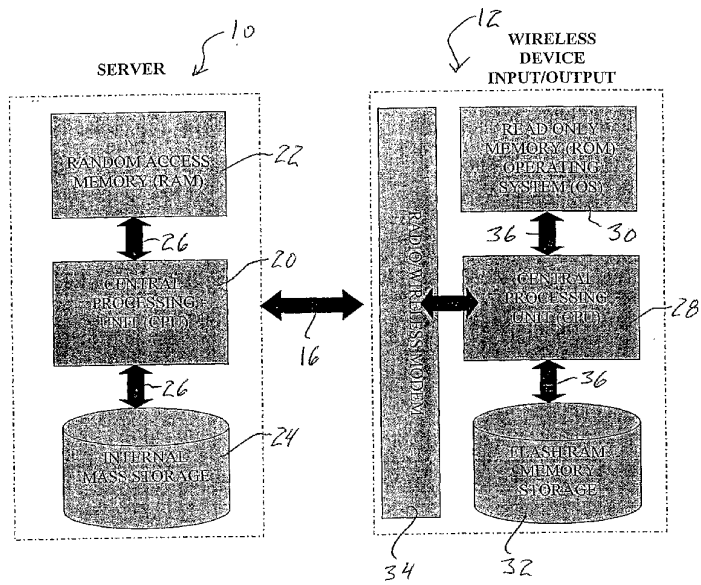
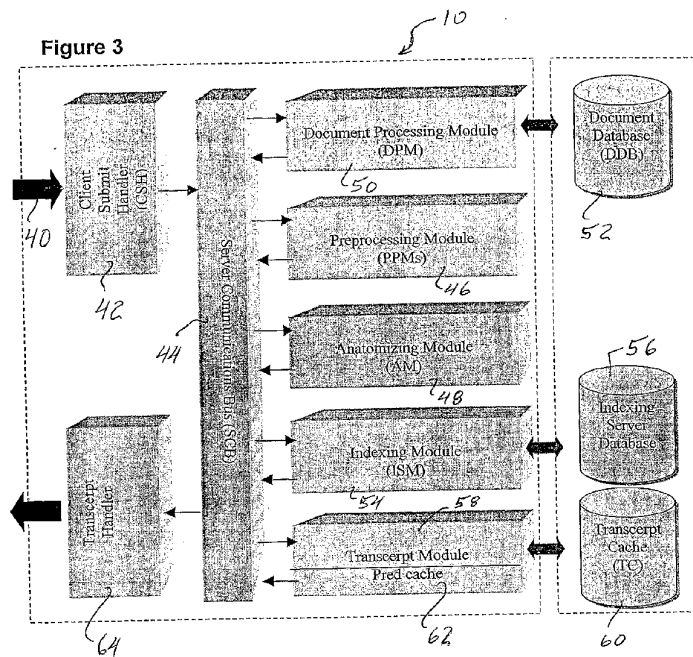


Figure 3

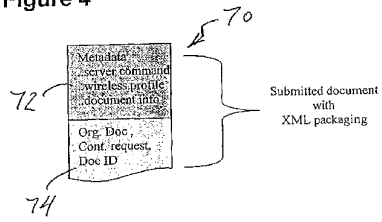


WO 02/077855

4/28

PCT/US02/09268

Figure 4



WO 02/077855

5/28

PCT/US02/09268

Figure 5

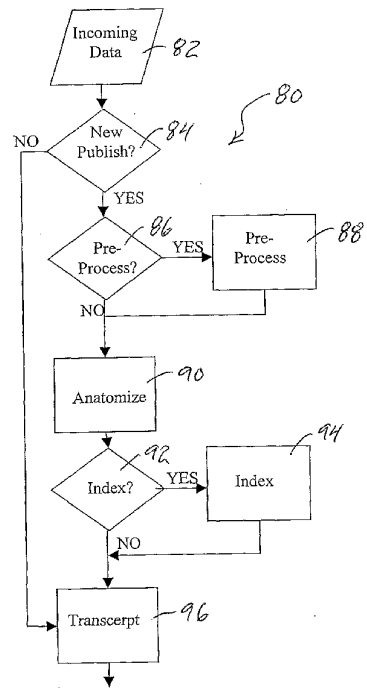
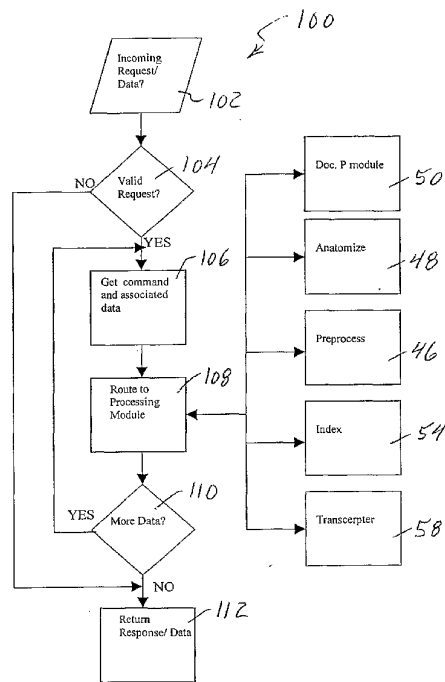


Figure 6

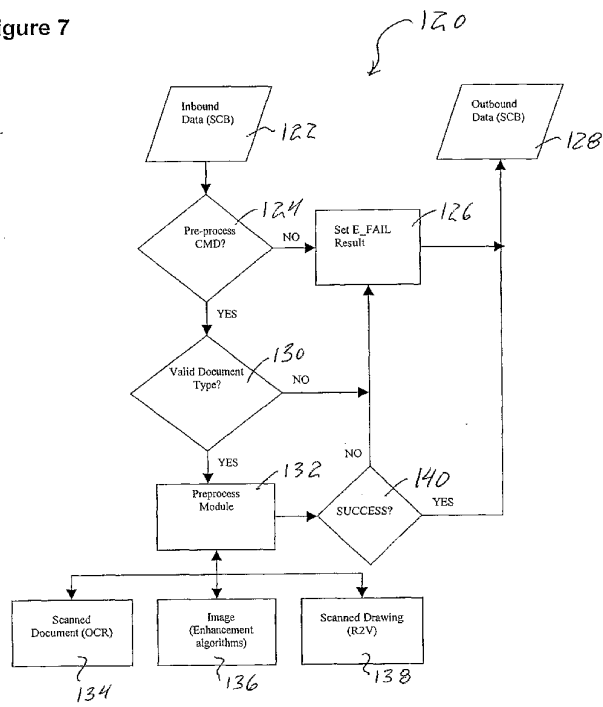


WO 02/077855

7/28

PCT/US02/09268

Figure 7

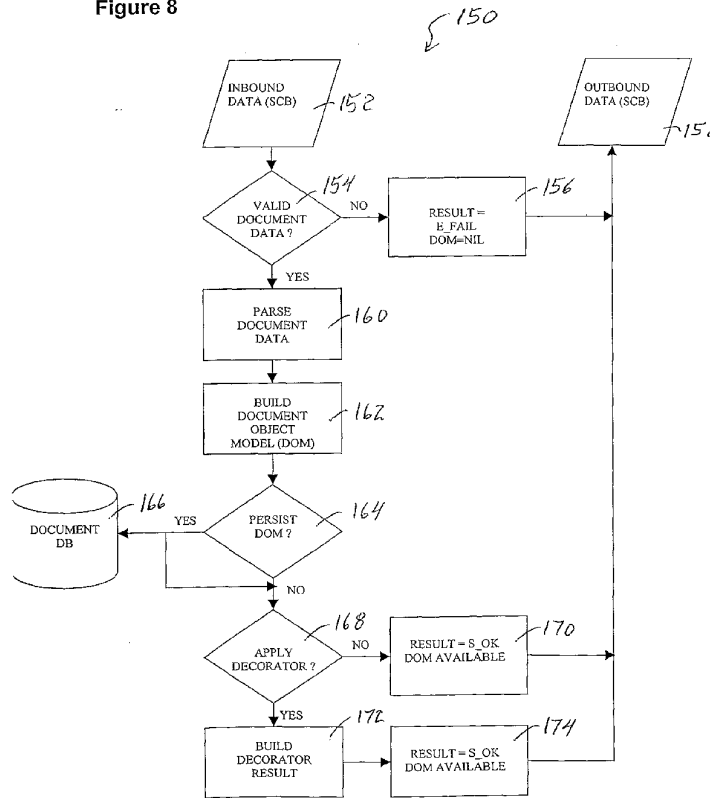


WO 02/077855

8/28

PCT/US02/09268

Figure 8

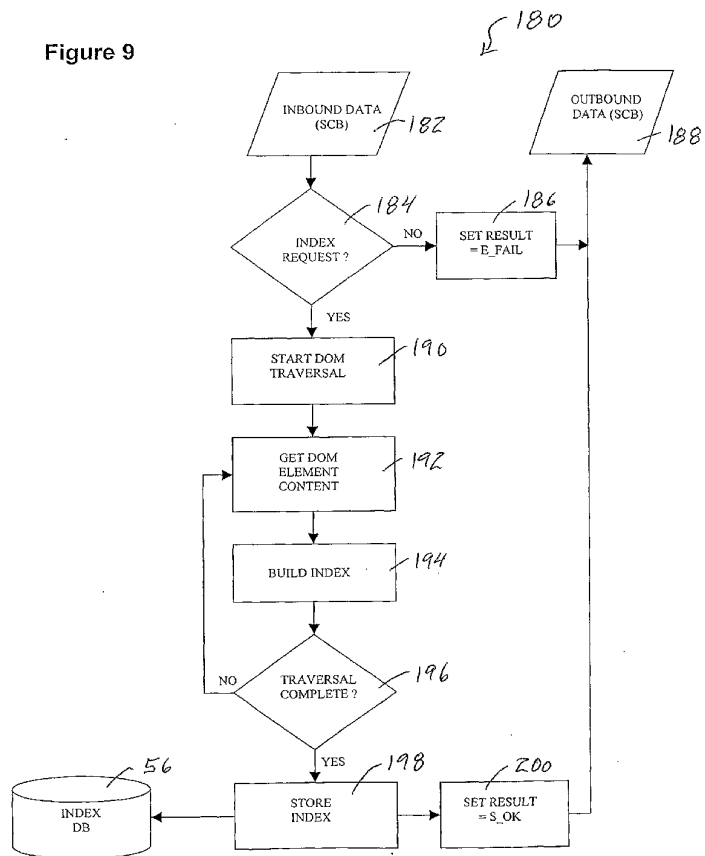


WO 02/077855

9/28

PCT/US02/09268

Figure 9



WO 02/077855

10/28

PCT/US02/09268

Figure 10

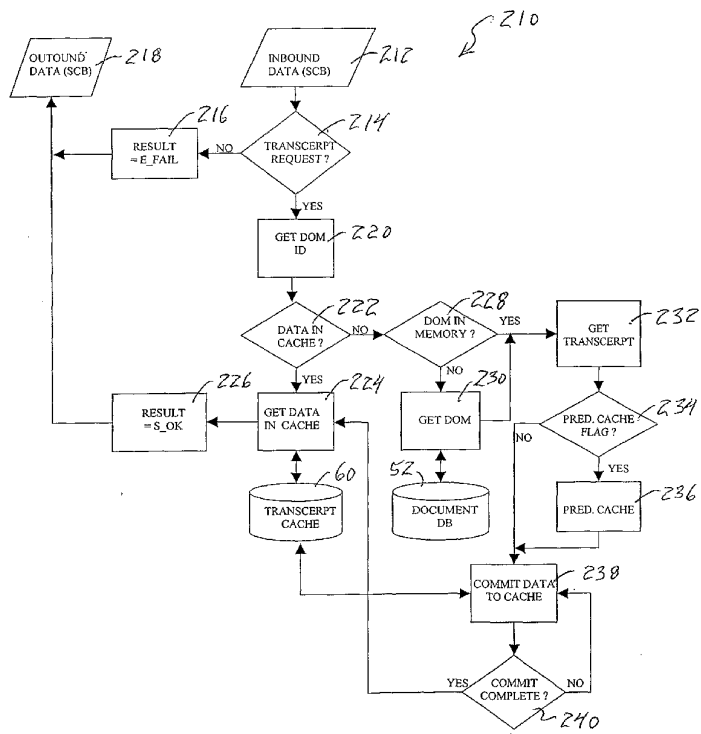
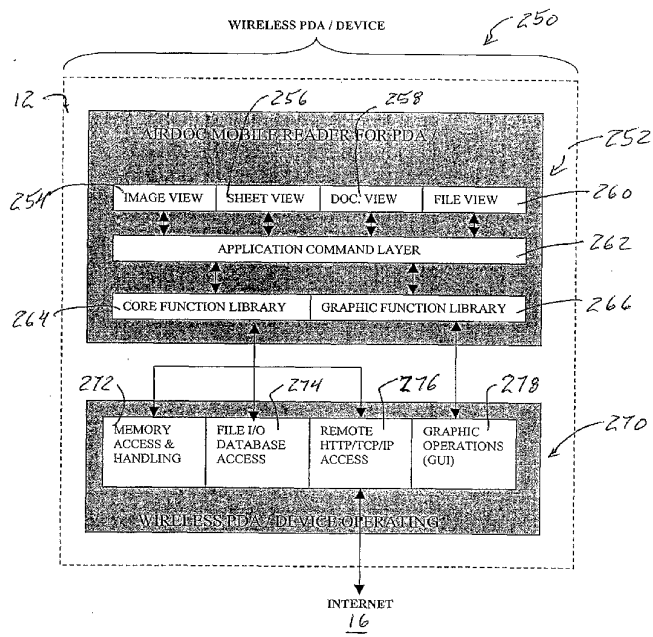


Figure 11



WO 02/077855

12/28

PCT/US02/09268

Figure 12

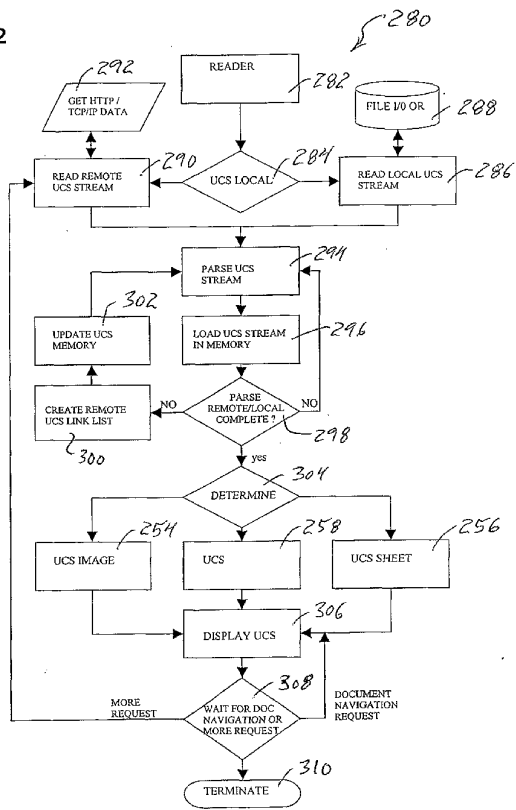


Figure 13

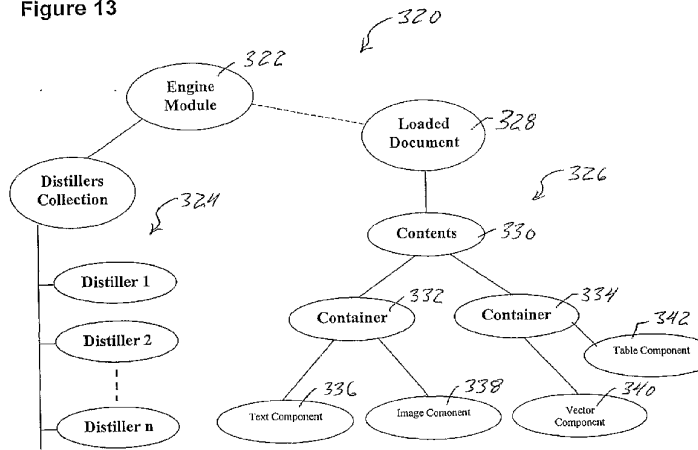


Figure 14

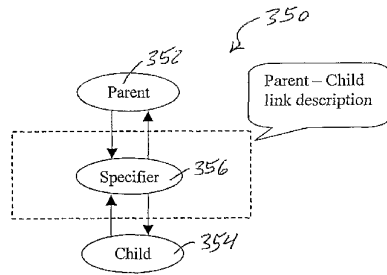


Figure 15

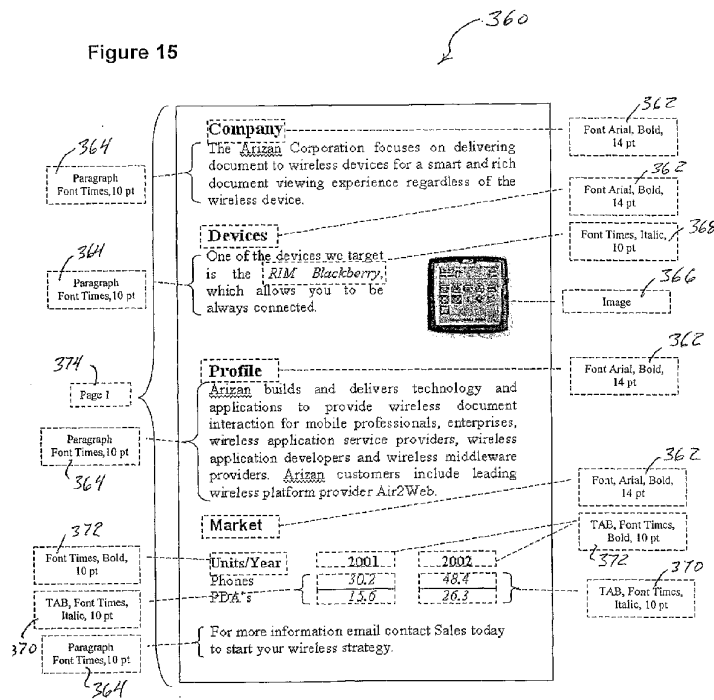


Figure 16

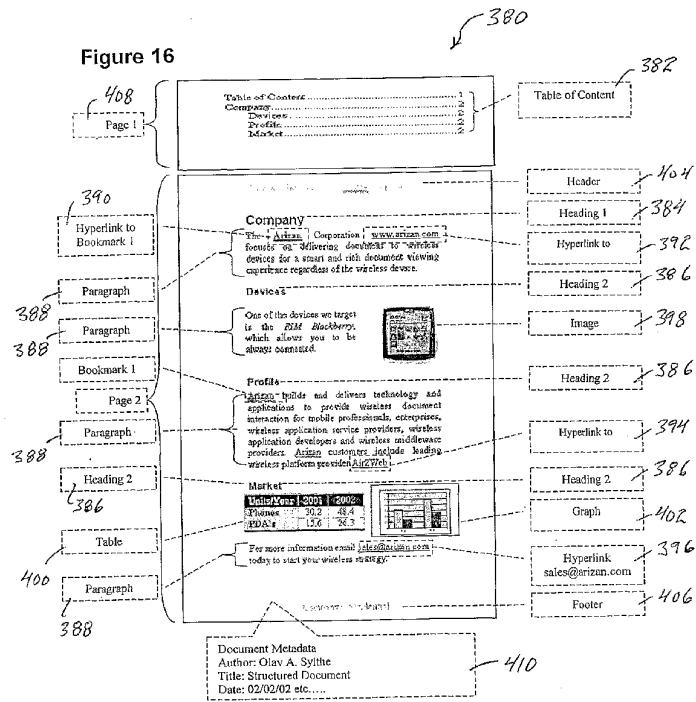


Figure 17
document

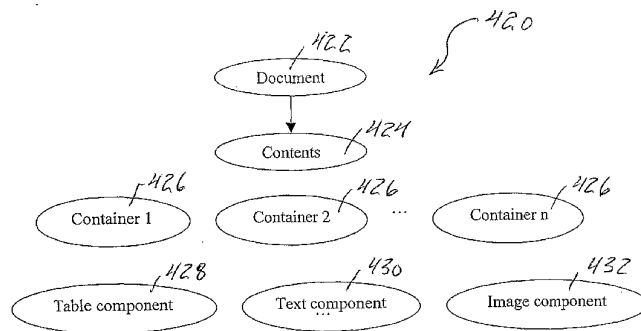


Figure 18

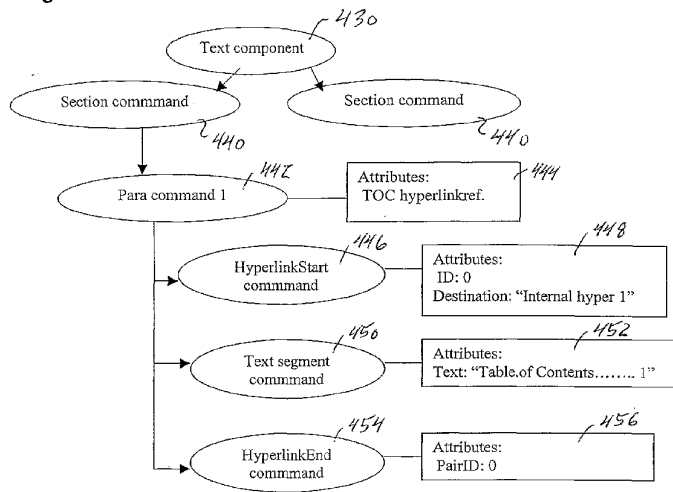


Figure 19 A

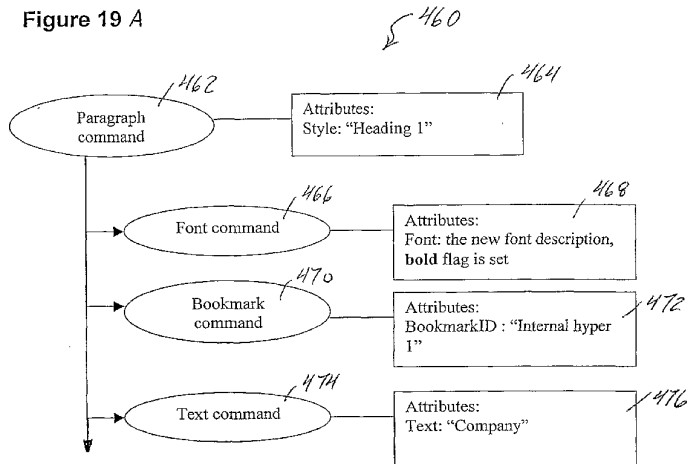


Figure 19B

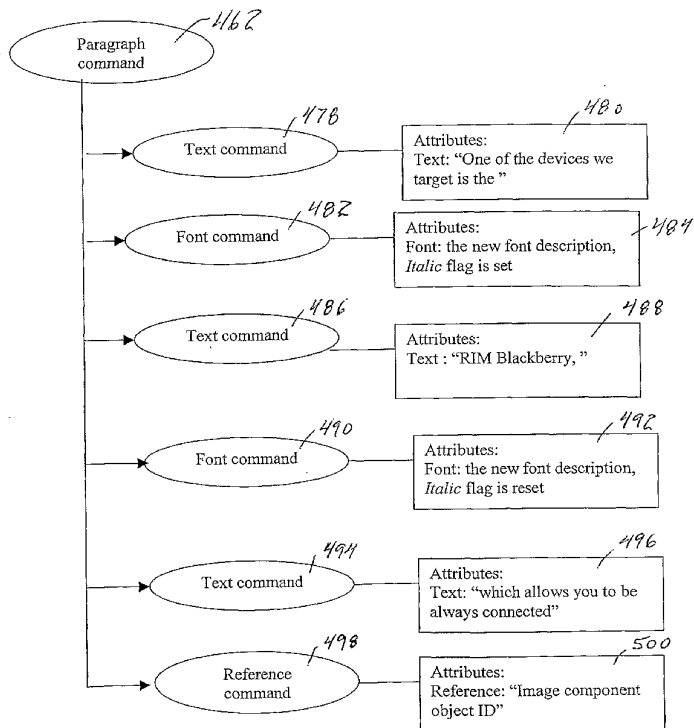
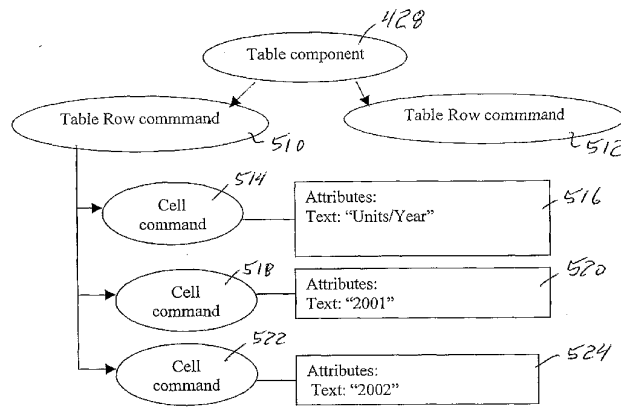


Figure 20



WO 02/077855

22/28

PCT/US02/09268

Figure 21A

380

Record ID	Structured Doc
1	UCS Signature
2	Document Header
3	Document Container
4	Paragraph
5	Paragraph Information
6	Bookmark
7	Simple Font Change
8	Text Segment
9	Paragraph
10	Paragraph Information
11	Paragraph
12	Paragraph Information
13	Hyperlink Begin
14	Simple Font Change
15	Text Segment
16	Simple Font Change
17	Text Segment
18	Text Segment
19	Hyperlink End
20	Paragraph
21	Paragraph Information
22	Hyperlink Begin
23	Simple Font Change
24	Text Segment
25	Simple Font Change
26	Text Segment
27	Text Segment
28	Hyperlink End
29	Paragraph
30	Paragraph Information
31	Bookmark
32	Hyperlink Begin
33	Simple Font Change
34	Text Segment
35	Simple Font Change
36	Text Segment
37	Text Segment
38	Hyperlink End
39	Paragraph
40	Paragraph Information

360

Record ID	Unstructured Doc
1	UCS Signature
2	Document Header
3	Section Container
4	Paragraph
5	Paragraph Information
6	Bookmark
7	Simple Font Change
8	Text Segment
9	Paragraph
10	Paragraph Information
11	Paragraph
12	Paragraph Information
13	Hyperlink Begin
14	Simple Font Change
15	Text Segment
16	Simple Font Change
17	Text Segment
18	Text Segment
19	Hyperlink End
20	Reference 1 - Text Box
21	Paragraph
22	Paragraph Information
23	Hyperlink Begin
24	Simple Font Change
25	Text Segment
26	Simple Font Change
27	Text Segment
28	Text Segment
29	Hyperlink End
30	Paragraph
31	Paragraph Information
32	Bookmark
33	Hyperlink Begin
34	Simple Font Change
35	Text Segment
36	Simple Font Change
37	Text Segment
38	Text Segment
39	Hyperlink End
40	Reference 2 - Image

WO 02/07855

23/28

PCT/US02/09268

Figure 21B

380

41	Hyperlink Begin
42	Simple Font Change
43	Text Segment
44	Simple Font Change
45	Text Segment
46	Text Segment
47	Hyperlink End
48	Paragraph
49	Paragraph Information
50	Hyperlink Begin
51	Simple Font Change
52	Text Segment
53	Simple Font Change
54	Text Segment
55	Text Segment
56	Hyperlink End
57	Paragraph
58	Paragraph Information
59	Paragraph
60	Paragraph Information
61	Simple Font Change
62	Text Segment
63	Paragraph
64	Paragraph Information
65	Simple Font Change
66	Text Segment
67	Hyperlink Begin
68	Simple Font Change
69	Text Segment
70	Hyperlink End
71	Simple Font Change
72	Text Segment
73	Hyperlink Begin
74	Simple Font Change
75	Text Segment
76	Bookmark
77	Text Segment
78	Hyperlink End
79	Simple Font Change
80	Text Segment
81	Paragraph
82	Paragraph Information
83	Reference 1 - Text Box
84	Reference 2 - Image
85	Simple Font Change

360

41	Paragraph
42	Paragraph Information
43	Hyperlink Begin
44	Simple Font Change
45	Text Segment
46	Simple Font Change
47	Text Segment
48	Text Segment
49	Hyperlink End
50	Paragraph
51	Paragraph Information
52	Hyperlink Begin
53	Simple Font Change
54	Text Segment
55	Simple Font Change
56	Text Segment
57	Text Segment
58	Hyperlink End
59	Paragraph
60	Paragraph Information
61	Paragraph
62	Paragraph Information
63	Simple Font Change
64	Text Segment
65	Paragraph
66	Paragraph Information
67	Simple Font Change
68	Text Segment
69	Hyperlink Begin
70	Simple Font Change
71	Text Segment
72	Hyperlink End
73	Simple Font Change
74	Text Segment
75	Hyperlink Begin
76	Simple Font Change
77	Text Segment
78	Bookmark
79	Text Segment
80	Hyperlink End
81	Simple Font Change
82	Text Segment
83	Paragraph
84	Paragraph Information
85	Simple Font Change

Figure 21C

380

86	Text Segment
87	Paragraph
88	Paragraph Information
89	Simple Font Change
90	Text Segment
91	Paragraph
92	Paragraph Information
93	Simple Font Change
94	Text Segment
95	Text Segment
96	Hyperlink Begin
97	Simple Font Change
98	Text Segment
99	Hyperlink End
100	Paragraph
101	Paragraph Information
102	Simple Font Change
103	Text Segment
104	Paragraph
105	Paragraph Information
106	Simple Font Change
107	Text Segment
108	Text Segment
109	Text Segment
110	Paragraph
111	Paragraph Information
112	Reference 3 - Image
113	Simple Font Change
114	Text Segment
115	Reference 4 - Spreadsheet
116	Paragraph
117	Paragraph Information
118	Simple Font Change
119	Text Segment
120	Paragraph
121	Paragraph Information
122	Paragraph
123	Paragraph Information
124	Simple Font Change
125	Text Segment
126	Text Segment
127	Paragraph
128	Paragraph Information
129	Simple Font Change
130	Text Segment

360

86	Text Segment
87	Paragraph
88	Paragraph Information
89	Simple Font Change
90	Text Segment
91	Paragraph
92	Paragraph Information
93	Simple Font Change
94	Text Segment
95	Text Segment
96	Hyperlink Begin
97	Simple Font Change
98	Text Segment
99	Hyperlink End
100	Paragraph
101	Paragraph Information
102	Simple Font Change
103	Text Segment
104	Paragraph
105	Paragraph Information
106	Simple Font Change
107	Text Segment
108	Text Segment
109	Text Segment
110	Paragraph
111	Paragraph Information
112	Simple Font Change
113	Text Segment
114	Paragraph
115	Paragraph Information
116	Simple Font Change
117	Text Segment
118	Paragraph
119	Paragraph Information
120	Paragraph
121	Paragraph Information
122	Simple Font Change
123	Text Segment
124	Text Segment
125	Paragraph
126	Paragraph Information
127	Simple Font Change
128	Text Segment
129	Paragraph
130	Paragraph Information

Figure 21D

380

131	Paragraph
132	Paragraph Information
133	Simple Font Change
134	Text Segment
135	Paragraph
136	Paragraph Information
137	Paragraph
138	Paragraph Information
139	Simple Font Change
140	Text Segment
141	Text Segment
142	Paragraph
143	Paragraph Information
144	Simple Font Change
145	Text Segment
146	Paragraph
147	Paragraph Information
148	Simple Font Change
149	Text Segment
150	Paragraph
151	Paragraph Information
152	Paragraph
153	Paragraph Information
154	Paragraph
155	Paragraph Information
156	Simple Font Change
157	Text Segment
158	Hyperlink Begin
159	Simple Font Change
160	Text Segment
161	Hyperlink End
162	Simple Font Change
163	Text Segment
164	Text Segment
165	Reference Containers
166	Reference 1
167	Reference 2
168	Reference 3
169	Reference 4
170	End of Document

360

131	Simple Font Change
132	Text Segment
133	Paragraph
134	Paragraph Information
135	Paragraph
136	Paragraph Information
137	Simple Font Change
138	Text Segment
139	Text Segment
140	Paragraph
141	Paragraph Information
142	Simple Font Change
143	Text Segment
144	Paragraph
145	Paragraph Information
146	Simple Font Change
147	Text Segment
148	Paragraph
149	Paragraph Information
150	Paragraph
151	Paragraph Information
152	Paragraph
153	Paragraph Information
154	Simple Font Change
155	Text Segment
156	Hyperlink Begin
157	Simple Font Change
158	Text Segment
159	UCS Signature
160	Document Header
161	Document Container
162	Paragraph
163	Paragraph Information
164	Bookmark
165	Simple Font Change
166	Text Segment
167	Paragraph
168	Paragraph Information
169	Simple Font Change
170	Text Segment
171	Paragraph
172	Paragraph Information
173	Paragraph
174	Paragraph Information
175	Bookmark

Figure 21E

360

176	Simple Font Change
177	Text Segment
178	Paragraph
179	Paragraph Information
180	Paragraph
181	Paragraph Information
182	Bookmark
183	Bookmark
184	Simple Font Change
185	Text Segment
186	Paragraph
187	Paragraph Information
188	Simple Font Change
189	Text Segment
190	Paragraph
191	Paragraph Information
192	Bookmark
193	Simple Font Change
194	Text Segment
195	Paragraph
196	Paragraph Information
197	Paragraph
198	Paragraph Information
199	Simple Font Change
200	Text Segment
201	Paragraph
202	Paragraph Information
203	Simple Font Change
204	Text Segment
205	Simple Font Change
206	Text Segment
207	Simple Font Change
208	Text Segment
209	Simple Font Change
210	Text Segment
211	Paragraph
212	Paragraph Information
213	Simple Font Change
214	Text Segment
215	Simple Font Change
216	Text Segment
217	Simple Font Change
218	Text Segment
219	Simple Font Change
220	Text Segment

WO 02/077855

27/28

PCT/US02/09268

Figure 21F

360

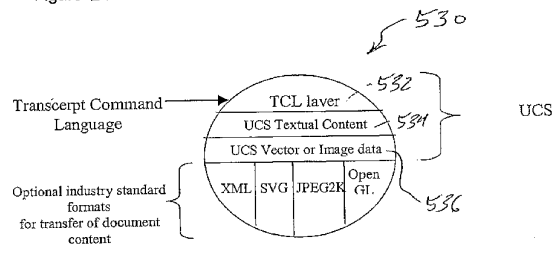
221	Paragraph
222	Paragraph Information
223	Paragraph
224	Paragraph Information
225	Simple Font Change
226	Text Segment
227	Paragraph
228	Paragraph Information
229	Simple Font Change
230	Text Segment
231	Paragraph
232	Paragraph Information
233	Simple Font Change
234	Text Segment
235	Simple Font Change
236	Text Segment
237	Paragraph
238	Paragraph Information
239	Paragraph
240	Paragraph Information
241	Reference containers
242	Reference 1
243	Reference 2
244	End of Document

WO 02/077855

28/28

PCT/US02/09268

Figure 22



【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
3 October 2002 (03.10.2002)

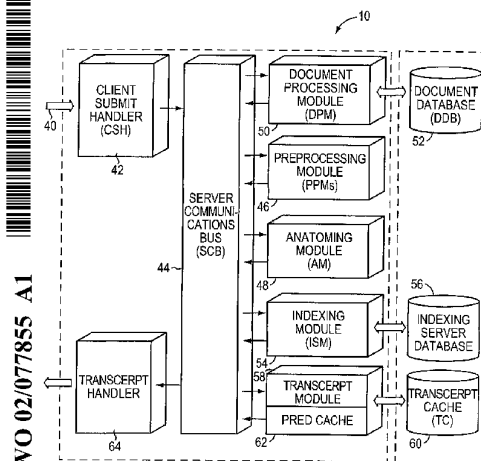
PCT

(10) International Publication Number
WO 02/077855 A1

- (51) International Patent Classification: G06F 17/21 (74) Agent: WINBURN, John, T. Morris Manning & Martin, LLP, 3343 Peachtree Road, NE, Atlanta, GA 30326-1044 (US).
- (21) International Application Number: PCT/US02/09268
- (22) International Filing Date: 25 March 2002 (25.03.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 60/278,137 23 March 2001 (23.03.2001) US
- (71) Applicant: ARIZAN CORPORATION [US/US]; 430 10th Street, N.W., Suite S-105, Atlanta, GE 30318 (US).
- (72) Inventor: SYLTHE, Olav, A.; 4867 Ashford Dunwoody Road #6208, Atlanta, GA 30338 (US).
- (81) Designated States (national): AU, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IL, IT, LU, MC, NL, PT, SE, TR), OAPI patent

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE



(57) Abstract: A system which analyzes, organizes and stores electronic documents in a document object model using a common markup language. The document object model is stored on a server (10,14,50,52) and the content is delivered to the wireless device (12) in a universal content system format (162). By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for transmission.

WO 02/077855 A1

WO 02/077855 A1 

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NL, SN, TD, TG).

(48) Date of publication of this corrected version:

9 January 2003

Declaration under Rule 4.17:

as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations

(15) Information about Correction:

see PCT Gazette No. 02/2003 of 9 January 2003, Section II

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/077855

PCT/US02/09268

**SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS
COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE****CROSS REFERENCE TO RELATED APPLICATION**

5 This application claims priority under 35 U.S.C. 119 to the benefit of the filing date of
the U.S. patent application serial no. 60/278,137, which was filed on March 23, 2001,
entitled "SYSTEMS AND METHODS FOR CONTENT DELIVERY OVER A WIRELESS
COMMUNICATION MEDIUM TO A PORTABLE COMPUTING DEVICE," which is
10 incorporated herein by reference.

BACKGROUND OF THE INVENTION**Field of the Invention**

The present invention relates generally to wireless content delivery of documents and
more specifically relates to organizing document content to be requested by reconstructed and
15 read on a wireless device.

Description of the Prior Art

With the rapidly expanding use of computers, electronic documents are stored and
communicated between computers with relative ease. Full desktop systems and laptop
20 systems readily transfer these documents over various communication media, such as the
Internet.

It is also desirable, however, to utilize these documents with the wireless
communications medium and utilizing mobile wireless devices. This presents a whole series
of different problems, since the bandwidth available can be on the order of 4 to 48 kbps,
25 which is not expected to increase past 56 kbps in the near term.

The wireless devices typically attempt to maximize battery life, while at the same
time minimizing weight and the device physical size. Typically, the devices also include

WO 02/077855

PCT/US02/09268

slow processors and a minimal amount of memory. At the same time, the devices have different, generally limited, operating systems, screen resolution and limited graphic rendering capabilities. Large documents with a wide variety of formats are therefore very unsuitable or impossible for use with these devices in their standard format.

5 It would thus be desirable to provide a system, which allows the wireless devices to have access to the electronic documents without regard to their size or formatting of the documents.

SUMMARY OF THE INVENTION

The present invention provides a system, which analyzes, organizes and stores
10 electronic documents in a document object model using a common markup language. The document object model is stored on a server and the content is delivered to the wireless devices in a universal content stream format. By utilizing the document object model and the universal content stream with a corresponding reader on the wireless device, the electronic document is reduced to only the essential content and the necessary format information for
15 wireless transmission.

To accomplish the above and related advantages, this invention may be embodied in the form illustrated in the accompanying drawings, attention being called to the fact, however, that the drawings are illustrative only, and that changes may be made in the specific construction illustrated without departing from the scope of the present invention.

20 BRIEF DESCRIPTION OF THE DRAWINGS

Various other functions, features and attendant advantages of the present invention will become fully appreciated as the same is described in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the several views, and wherein:

WO 02/077855

PCT/US02/09268

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is an illustration of a data processing system or server where the invention is employed to communicate with wireless devices of clients over the Internet.

Figure 2 is an illustration of a block diagram of the data processing system and
5 wireless devices represented by **Figure 1**.

Figure 3 is a block diagram illustration of the invention residing on the server in **Figure 1** detailing individual process modules of the present invention.

Figure 4 illustrates the data packaging of a document submitted to the present invention.

10 **Figure 5** is a flowchart illustrating the process flow of a submit request being processed by the data processing system on the server.

Figure 6 is a flowchart detailing the process flow of an incoming request being processed by the Server Communication Bus (SCB) module of the server data processing system.

15 **Figure 7** is a flowchart detailing the process flow of a pre-processing module for invoking external processes for Optical Character Recognition (OCR), Image Enhancement and Raster to Vector conversion (R2V).

Figure 8 is a flowchart detailing the process flow of a data processing system Anatomization module used to construct the documents anatomy in establishing the
20 Document Object Model (DOM).

Figure 9 is a flowchart detailing the process flow of an Indexing Module for indexing content of a submitted document.

Figure 10 is a flowchart detailing the process flow of a Transcription Module of the invention for delivering the relevant content excerpts to the wireless/wired clients.

WO 02/077855

PCT/US02/09268

Figure 11 is a block diagram illustrating the invention residing on a wireless device or client detailing the individual data processing modules.

Figure 12 is a flowchart detailing the process flow of the invention residing on a wireless device receiving transcribed data in the form of the universal content stream (UCS).

5 **Figure 13** is a diagram illustrating the architecture of the processing engine of the present invention with a loaded structured or unstructured document or data stream.

Figure 14 is a diagram illustrating the relationship of a parent and child object in the invention of a constructed Document Object Model (DOM) for a loaded document.

10 **Figure 15** is an illustration of a sample unstructured document with which the invention might be used.

Figure 16 is an illustration of a sample structured document with which the invention might be used.

Figure 17 is an illustration of the constructed Document Object Model (DOM) utilized by the invention for a loaded structured document.

15 **Figure 18** is a diagram illustrating a Document Object Model of the present invention for textual content.

Figure 19A is a diagram illustrating the inventions document object model in constructing a paragraph of textual content.

Figure 19B is a continuation of illustrating the paragraph command in **Figure 19A**.

20 **Figure 20** is a diagram illustrating the Document Object Model for a loaded document containing table data as typically found in a spreadsheet document type.

Figures 21A – 21F are illustrations detailing the command content of a transcript response from the invention to a wireless/wired device in the form of a Universal Content Stream data structure for a structured and unstructured document respectively.

WO 02/077855

PCT/US02/09268

Figure 22 is one embodiment of a layout for the communication format of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A data processing system or server **10** communicating with one or a plurality of wireless devices or clients **12** where the present invention can be utilized is illustrated in **Figure 1**. The data processing system **10** typically includes conventional components, such as a central processing unit, internal mass storage, and random access memory, which are utilized by the present invention. Communication to and from the data processing system **10** will typically come through a secure software or hardware processing system/unit commonly known as a Firewall/Proxy Server **12** in a conventional manner to ensure authorized and valid access to the network in which the invention is deployed. The invention executing on the data processing system or server **10** will typically receive inbound and transmit outbound data via the Internet **16** to a portion of the invention residing on the wireless device or client **12**. A current embodiment of the present invention can utilize as the data processing system **10** DELL, COMPAQ, SUN or other recognized manufacturers of server hardware running a Windows operating system release NT 4/2000/XP or higher or on a Sun Solaris operating system release 8 or higher.

Figure 2 illustrates a more detailed overview of the data processing system or server **10** and the wireless device or client **12**. The server **10** includes a central processing unit (CPU) **20**, a random access memory (RAM) **22** and an internal mass storage device **24** all interconnected by a data channel commonly known as a bus **26**. The data processing system **10** communicates with the wireless device client **12** over the Internet **16**. The wireless device client **12** processing system includes a central processing unit (CPU) **28** designed for small footprint devices such as a Motorola Dragon Ball CPU, or the Intel StrongARM CPU coupled with a read only memory (ROM) **30** which stores a small footprint wireless device

WO 02/07855

PCT/US02/09268

operating system such as Palm OS, or PocketPC OS with an internal flash ram memory storage 32 communicating with the Internet 16 utilizing a wireless radio/modem 34 all coupled together using a databus 36.

The data processing system 10 preferably is configured to execute and perform the functionality of the current embodiment of the present invention by having an operating system able to utilize data instructions written to execute using the functionality provided by the operating system to utilize typical programming languages such as C++ and Java.

Figure 3 illustrates a block diagram of the process flow for the invention's server side processing modules of the present invention. A module by definition is an individual data process optional or obligatory that combined with other modules forms the invention's data processing system 10. A structured or unstructured electronic document or data stream 40 is submitted from the wireless or wired client side 12 and enters the server 10 from the network as binary data. A Client Submit Handler 42 receives the client request and verifies that this is a valid request and passes the received data with appropriate routing commands based on the client request over to a Server Communication Bus 44, which reads the routing commands and sends the received data to the required processing modules in the required order as indicated by the routing commands. If the first routing command is a request for a Preprocessing Module (PPM) 46 the received data is passed along for document or data optimization. The Preprocessing Module's 46 main purpose is to improve document quality for utilization by the invention if the document type is a scanned document containing imagery depicting textual content, containing binary imagery representing a picture, or a scanned document depicting a drawing using line or circular drawing elements.

For a scanned document containing imagery depicting textual content a technique known as Optical Character Recognition (OCR) is employed with the purpose of extracting the textual characters for processing by the invention. Any OCR style software package such

WO 02/077855

PCT/US02/09268

as OmniPage from ScanSoft Corporation can be applied by the invention for this purpose. For imagery representing a picture the invention can apply commonly known image algorithms for improving image quality such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. For a scanned document depicting a drawing using line or circular drawing elements raster to vector (R2V) software is employed for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct the submitted image as vector drawing data on the inventions wireless/wired client. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the invention for this purpose.

The Server Communication Bus 44 will then route the data 40 on to an Anatomization Module (AM) 48, which is responsible for establishing the anatomy of the document and to populate the Document Object Model (DOM) as described in Figure 17 and will as a dependant process utilize the SCB 44 to route the data to a Document Processing Module (DPM) 50 where an Engine as described in Figure 13 will load the structured or unstructured document using a distiller module. If the SCB's 44 routing commands include a command to persist (save) the document data parsed by the DPM 50 the document data will be saved in a Document Database (DDB) 52. The Document Database 52 is a connector to typical software packages commonly referred to as databases such as Microsoft SQL Server from Microsoft Corporation. Upon successful processing completion of the document by the DPM 50 the SCB 44 will issue a command back to the Anatomization Module 48 and the Document Object Model will be updated with the content and anatomy of the document. If an indexing request is part of the command stack of the SCB 44, an Indexing Module (ISM) 54 will be invoked and the textual content available from the Anatomization Module 48 of the document will be indexed using commonly known software products such as Microsoft

WO 02/077855

PCT/US02/09268

Indexing Server and the results along with the associated references to the Document Object Module will be stored in an Indexing Server Database 56.

Upon completion by the ISM 54, the SCB 44 command stack will invoke a Transcript Module (TM) 58 (transcript = transmit excerpt) and the initial excerpt of content will be prepared for transmission by invoking the Anatomization Module 48 and requesting an initial Table of Content or Summary of the document based on the content available in the Document Object Module. Upon return of the transcript by the AM 48, a copy of the initial transcript will be placed in a Transcript Cache (TC) 60 for repeat access purposes. If the SCB 44 command stack contains a Predictive Caching Module 62 request, the AM 48 will be repeatedly invoked for the lifetime of the client session with the purpose of predicting future client requests for related content data from the document. The Predictive Caching Module 62 will, for the lifetime of the client session, then populate the Transcript Cache 60 with the transcript results from the Transcript Module 58. The initial transcript is then upon completion from the TM 58 returned via the SCB 44 to a Transcript Handler 64 where the correct client device 12 for the return data is identified and the transcript is sent to the network 16 and back to the wireless/wired client 12.

If the Client Submit Handler 42 receives a client request for more content from a previously submitted document it passes the request over to the Server Communication Bus 44, which in turn hands it off to the Transcript Module 58, which will then look into the Transcript Cache 60 to see if a transcript generated by the Predictive Caching Module 62 is available for transmittal. If not, the Anatomization Module 48 will be invoked and a new transcript will be prepared following the procedure as described for documental submittal for the System 10. Upon completion the transcript will be placed in the Transcript Cache 60 and the Transcript Handler 64 again will be invoked and deliver the transcript to the client wireless/wired device 12 as described for the initial document submittal.

WO 02/077855

PCT/US02/09268

Figure 4 represents the packaging of a document or content request 70 submitted for publishing by the present invention. An XML based metadata description 72 is the encapsulation of the packaging and will contain the appropriate server command such as publish or query, wireless device profile of the requesting device 12 providing information such as device type, device screen size, free storage space, estimated bandwidth transfer speeds etc. and information on the document 70 such as document name, document date, document type or any other supplemental information. An Original Document or unique document ID 74 for the content request 70 makes up the other part of a document submittal request. If an original document is part of the submitted request the document will be published or republished if required. If a unique document ID is sent as part of the request it indicates that the document is already published and the server will look in the Transcript Cache 60 or retrieve the document from the Document Database 52 based on the unique Document ID.

Figure 5 illustrates a flowchart process 80 for a wireless client 12 submit request to the data processing system server 10. An incoming request 82 with data has arrived. The first step of the process 80 is to establish if this is a new publish request 84. If it has a unique document ID 74, it is not a new publish request (NO) and the document already resides on the server data processing system 10, then a Transcript processing step 96 will take place directly. If an original document is being submitted, the metadata 72 is processed to see if there is a client request to perform Pre Processing 86 and the PreProcess module 46 will be invoked in a step 88. If not, the Anatomization process 90 will be performed. Upon process completion from the Anatomization process module 48 the metadata 72 is processed to see if there is a client request to perform indexing 92 on the anatomized content. If this is true, then the Index Processing Module 54 is invoked and the content will be indexed in a step 94. Upon completion of step 92 and/or step 94, the Transcript module 58 will be invoked in a

WO 02/07855

PCT/US02/09268

step 96 with the purpose of delivering the initial content result back to the requesting wireless client 12.

Figure 6 is a flowchart 100 illustrative the data flow of the server side data processing system 10 of the Server Communications Bus 44 (SCB). An incoming request and/or data is received 102 by the SCB and the SCB looks to determine if this is a valid request 104 for the SCB 44. If the request is valid (YES), the command and associated data is retrieved 106 by the SCB. Associated data will be an original document or original document unique ID. Pending on the SCB command the data will be routed 108 to the appropriate processing module where any single or combination of SCB commands will be performed by the server side data processing system 10 including any of the system modules; the Document Processing Module 50, the Anatomization Module 48, the Preprocessing Module 46, the Indexing Module, and the Transcriber Module 58. As the SCB 44 is designed to process both asynchronous or synchronous request requests for data processing by the individual data processing system modules 46, 48, 50, 54, 58 the requests may or may not be completed when the SCB 44 will check if there are additional commands and data to be processed in a step 110. If there are more commands or data (YES) the process will continue again with the step 106 until the command stack is emptied. If no more data or commands are pending in the step 110, the SCB 44 will return the appropriate response and resulting data from the server side data processing modules in a step 112 to the requesting wireless/wired client 12. If no valid command is detected by the SCB in the step 104 (NO) the appropriate failure code will be assigned by the step 112 and the resulting return or response will be sent to the requesting wireless or wired client 12.

Figure 7 illustrates a flow chart 120 describing the server side data processing system for the Pre Processing Module 76. Inbound Data 122 is received on the SCB 44. A validation step 124 is performed to see if this is a valid Pre-process command. If the

WO 02/07855

PCT/US02/09268

command is not recognized as a valid pre-process command (NO) the result error code is set to E_FAIL in a step 126 and the error code is returned as outbound data on the SCB in a step 128. If the step 124 determined that the command is a valid preprocess command (YES), a step 130 checks to see if the document type submitted with the preprocess command is a valid supported document. If the answer is negative (NO) the step 126 again sets the result error code to E_FAIL and the error code is returned as outbound data on the SCB in the step 128. If it is determined that this is a valid document type (YES) in the step 130, then the Preprocess module 46 is invoked in a step 132. Based on the document type determined as part of the successful step 130, an appropriate preprocessing sub module 134, 136 or 138 is invoked. Step 134 performs an Optical Character Recognition (commonly referred to as OCR) to convert a scanned document into textual content. The invention is designed to use commercially available software for OCR purposes from software vendors such as OmniPage from ScanSoft Corporation.

Step 136 is invoked by the step 132 if the valid document type determined in the step 130 is an image or picture data in saved form such as BMP (Bitmap) or JPEG (Joint Photographic Experts Group) upon which commonly known mathematical formulas can be employed to improve the visual aspects of the image picture data such as sharpen, blur, Gaussian blur, motion blur, intensity, contrast, gamma correct, invert, hue and saturation, histogram equalize, emboss, mosaic, median, average etc. A step 138 is invoked by the step 132 if the valid document type determined in the step 132 is a scanned drawing depicting a drawing using line or circular drawing elements. A step 138 will then employ commercially available raster to vector (R2V) software for the purpose of extracting drawing element and positioning information from the scanned drawing to reconstruct or recreate the submitted image as vector drawing data on the wireless/wired client 12. Any R2V style software package such as VP HybridCAD from SoftTelec can be utilized by the present invention for

WO 02/077855

PCT/US02/09268

this purpose. Upon completion of any sub process 134, 136, 138 by step 132 then a step 140 will check if the process is determined to be successful. If the step 140 is a success (YES), the preprocessed data will be returned back to the SCB in the step 128. If the step 140 determines that the result from the step 132 is unsuccessful (NO), the result error code is set to E_FAIL in the step 126 and the error code is returned as outbound data on the SCB in the step 128.

Figure 8 is a flow chart 150 illustrating the process flow of the server side data processing system 10 for the Anatomization process 48. Inbound data from the SCB is received in a step 152, which is passed to a step 154 where it is determined if the document data is valid or not. If the document data is invalid (NO) a step 156 sets the result error code to E_FAIL and the pointer to the DOM is set to equal zero or NIL, and returns the error code as outbound data on the SCB in a step 158. If it is determined that the document data is valid (YES) in the step 154, the document data received will be parsed in a step 160 and the Document Object model (DOM) then is built up in a step 162. After the DOM has been constructed by the step 162, then a step 164 will check if there is a command request to persist or store the constructed DOM. If the step 164 is true (YES) the DOM data will be stored in the Document Database 52 in a step 166 before continuing on to a step 168. If the step 164 is false (NO), then the step 168 will check to see if there is a command to apply a decorator. A decorator is a command to traverse the Document Object Model established by the step 162 and extract content or navigational information such as a Table of Content (TOC), Bookmarks, Hyperlink, Find Text etc. If the step 168 has no decorator command to apply (NO), a step 170 will set the result code to S_OK and set the DOM pointer before returning the outbound data in the step 158. If the step 168 determines that decorators are to be applied (YES), a step 172 is invoked and the decorators will traverse the DOM established

WO 02/07855

PCT/US02/09268

in the step 162. Upon completion of the step 172 the result code is set to S_OK and the DOM pointer is set in a step 172 before returning the outbound data in the step 158.

Figure 9 illustrates a flow chart 180 of the Indexing Process 54 for the server side data processing system 10. Inbound data from the SCB 44 arrives in a step 182. A step 184 checks if there is a request to index the data. If the step 184 is determined to be false (NO), then a step 186 will set the result code to E_FAIL and the result will be returned as the outbound data to the SCB in a step 188. If the step 184 is true (YES), then the system will start to traverse the Document Object Model (DOM) in a step 190. In a step 192 the content of a single DOM element is retrieved and then the system 10 will start to build the Index in a step 194. In the step 194 the system will typically employ commercially available software indexing packages such as Microsoft Indexing Server. After the content of the retrieved DOM element has been indexed in the step 194 then a step 196 checks to see if the DOM traversal is complete. If the answer is NO, then the steps 192 and 194 will again be processed until the DOM traversal is complete (YES) in the step 196. The index is then stored in a step 198 in the Index Database 56 and the result code is set to S_OK in a step 200 before the result is returned as the outbound data to the SCB in the step 188. The Index DB 56, utilized in the step 198 can be any commercial database product such as MS SQL Server or Oracle.

Figure 10 is a process flowchart 210 of the Transcription module 58 for the server side data processing system 10. Inbound data arrives from the SCB in a step 212 and the data is checked to see if this is a transcript request or not in a step 214. If it is not a transcript request (NO), then a step 216 will set the Result code to E_FAIL and pass the result back to the SCB in a step 218 as outbound data. If the step 214 determines that the data is a transcript request (YES), then a step 220 is invoked to obtain the ID of the Document Object Model. The DOM ID is used to identify the correct document either from the transcript cache 60 or from the Document Database 52. A step 222 checks to see if the data is available

WO 02/077855

PCT/US02/09268

in the cache 60. If the answer is YES, then a step 224 will get the data from the transcript cache 60 and a step 226 will set the result to S_OK and return the outbound data in the step 218 onto the SCB. If there is no data in the cache (NO), during the step 222 then a step 228 will see if there is a DOM loaded in the memory and if it matches the DOM ID.

- 5 Because the system 10 employs caching it will typically retain the DOM and DOM ID for a specified number of documents, such as the last 32 processed documents and if the DOM ID matches any of the DOM content in the cache it will be put back in memory unless it's already there, meaning it was the last document processed. If the test in the step 228 determines that there is no DOM in memory or the DOM ID does not match (NO) (meaning
- 10 the DOM is not in the cache) then a step 230 is invoked, which will retrieve the DOM from the Document Database 52 before proceeding with a step 232. The system 10 allows the DOM to be saved to a database or file for immediate retrieval without having to reprocess the original document used to create the DOM initially. If the step 228 determines that there is a DOM in memory and the Document ID matches (YES), then the step 232 is invoked to get
- 15 the transcript. A step 234 then looks to verify if the SCB command received in the step 212 as inbound data from the SCB has the Predictive Caching Flag set. If the flag is set (YES), then a step 236 will start the asynchronous Predictive Caching process 62 before proceeding with a step 238. If the step 234 determines that the flag is not set (NO), then the step 238 is invoked and the transcript data retrieved in the step 232 will be committed to the cache 60.
- 20 A step 240 will then test repeatedly to see if the commit is complete, returning back to the step 238 if the test results in a negative (NO). When the step 240 returns true for commit complete (YES) then step 224 is invoked and the transcript data is retrieved from the cache. The step 226 will set the result to S_OK and return the outbound data in the step 218 onto the SCB.

WO 02/07855

PCT/US02/09268

Figure 11 is a detailed diagram **250** of the present invention wireless client processing system residing on a wireless device or Personal Digital Assistant (PDA) **12**. A reader **252** is a software application including a plurality of individual software function elements and libraries **254** through **266** that makes function calls to a Wireless PDA/Device operating system **270**. The operating system **270** includes a plurality of individual software functions forming the operating system **270**. Commonly known operating systems (OS) for wireless PDA/devices are Palm OS by Palm Corporation, Windows CE by Microsoft Corporation, J2ME by Sun Corporation, EPOC by Symbian and BREW by Qualcomm. The inventions client side software functionality is designed to read the defined Universal Content Stream, also commonly referred to as UCS. The reader **252** has the Graphic Function Library **266** that offers a set of graphical instructions used to render textual, vector and image-based data on the device **12**. To accomplish this the library **266** uses available graphics operations **254** from the device OS **270**. The Core Function Library **264** offers a set of functions derived from the device OS **270** for the memory access and handling **272**, the file input/output and/or database access **274**, and the wireless data access **276** for retrieving data from the Internet **16**. The reader **252** then has the application command layer **262** that invokes the core library **264** and the graphic function library **266** to be able to offer a graphical user interface GUI as defined in the view modes **254**, **256**, **258** and **260**.

Based on the data loaded from the UCS the Image View **254** will display image or picture based data, the Sheet View **256** will display spreadsheet data in a spreadsheet style grid, the Document View **258** will display textual content in a rich and intuitive way and the File View **260** will display a user interface for loading and saving the UCS file on the device **12**. The layered design in the reader **252** is designed so that only the core function library **264** and the graphics function library **266** will be different on each wireless/pda device **12** on which the client side processing system is deployed on. This allows the invention to be

WO 02/077855

PCT/US02/09268

quickly moved or ported to different operating systems such as Palm OS, Windows CE, J2ME, EPOC and BREW where the two primary programming languages are Java and C++ as commonly know by those skilled in the art of programming. A USC binary file is delivered to the device 12 from the server side data processing system 10 over the Internet 16 and the wireless client reader 252 retrieves this data by using the functions for wireless data access (load) 276 in the core function library 264, which in turn calls the library for remote HTTP/TCP/IP access 276 of the wireless PDA/device OS 270.

Figure 12 is a detailed flowchart 280 of the client side processing application residing on the wireless PDA/devices 12. In a step 282 the reader 252 is invoked either by being launched by another application on the device 12 such as the Inbox application or by the user selecting the application icon for the reader and invoking it. In a step 284 the passed Universal Content Stream (UCS) if any, will be verified to be local or remote. If local then the UCS is already stored on the device 12 and a step 286 will read the local UCS stream from a file or database 288. If the UCS is determined to be remote in the step 287 (not residing on the device 12) then a step 290 is invoked to read the remote UCS stream. A step 292 will then obtain the remote UCS residing on the server side data processing system 10 using the Internet 16. Upon completion of either step 286 or 290 the execution will continue with a step 294 where the retrieved UCS data is parsed. A step 296 will then load the data in device memory and then a step 298 will test to see if the parsing of the remote or local UCS data is complete. If the step 298 is not complete for local data (NO) then the step 294 will be invoked again until the step 298 is true (YES) for local UCS files. If the step 298 for remote UCS parsing is not complete (NO) then a step 300 will create a remote link list for any UCS data described to still reside on the server 10 for any future transcripts to the client. Then a step 302 updates the UCS memory with the remote links and continues with the step 294 until the remote parsing step 298 is complete (YES).

WO 02/077855

PCT/US02/09268

After the UCS data has been parsed and is loaded into memory, then a step 304 determines the view type or combination of view types for the UCS data. Depending on the view type either Image View 254, Document View 258, or Sheet View mode 256 is invoked and then the UCS data is displayed or rendered in a step 306. After displaying the UCS data

5 in the step 306 the client data processing application waits for user input in a step 308. If the user navigates within the loaded document using menu options, or track/scroll wheel, stylus or keypad then the step 308 will be repeatedly invoked until the user terminates the reader 252 by choosing the exit option and a step 310 will exit the mobile reader and return to the invoking application prior to the step 282. If the user selects or invokes a more request in the

10 step 308 for additional UCS data, then the step 290 will be invoked again with the same following process flow as previously described until the user chooses to terminate the application in the step 308 and then the step 310 will exit the mobile reader and return to the invoking application prior to the step.

Figure 13 is a diagram 320 representing the document engine structure with a

15 hypothetical document loaded for the document-processing module 50. The composition of the document engine is a collection of individual components that are categorized as the following modules, an Engine Module 322, a set of Distiller Modules 324 and a loaded document 326 and Decorator Modules (not illustrated). The distiller module 324 functionality is mainly to handle different document types. This set of modules is responsible

20 in particular for parsing a known document format, constructing the document structure and, if desired, persisting the document structure back to a known document format. The set of distiller modules 324 is open in the sense that new distiller modules can be added or removed at runtime to accommodate the need for a flexible environment. So the set of supported document types can be expanded or reduced as required. The engine module 322

25 functionality is mainly to host the running distiller modules 324 and, if desired, the loaded

WO 02/077855

PCT/US02/09268

documents 326, offering a uniform environment for these. It is also used to provide common behavior for clients (e.g. loading a document). A plurality of the document structure modules are the building pieces of a loaded document. They include: a document node 328 (representing the loaded document and containing all other nodes), contents nodes 330
5 (representing the contents with a certain detail degree of a document), a pair of container nodes 332, 334 (representing containers for other nodes), a text component 336 (representing textual content), an image component 338 (representing image content), a vector component 340 (representing vector content), and a table component 342 (representing table data such as spreadsheets). The document structure 326 is a tree-like structure (with the possibility of
10 sharing of nodes, so the more appropriate description would be graph-like). This fact allows the user to practically construct documents in any fashion desired without being constrained by a rigid document structure.

This is achieved by the fact that each node implements collection like methods (for children and/or parents). Behavior like rendering can be attached to the node itself or be
15 offered as an external component. This is facilitated by the existence of a set of standard iterators of the document structure. The choice between internal and external behavior can be taken by taking into consideration and the requirements of the intended operation (speed, size, deployment etc.) The decorator modules functionality (not shown in Figure 13) is mainly to perform customized selected operations on the document structure. Examples
20 include the extraction of the table of contents, list of bookmarks or hyperlinks, find operations etc. It is here where the transcripction technology is employed. They are separate modules, with their own framework, that operate completely independent of the other engine components. In fact the engine can operate without the decorator modules being installed. They just add more functionality. The decorator modules framework is composed of a
25 decorator manager (which is by itself a component) and a set of installed modules that

WO 02/07855

PCT/US02/09268

perform certain operations. The individual modules have to register in a predefined way in order to integrate into the framework and be recognized by the decorator manager. A decorator is a software component as commonly referred to by those skilled in the art of software development designed for the sole purpose of navigating the Document Object Model (DOM) to retrieve specific content and/or navigational information. In other words it can be stated that a decorator component constitutes a command. There are no limitations on how many decorators can be used to navigate the DOM. Typical decorators (commands) are: retrieve bookmarks in a document, retrieve hyperlinks in a document, retrieve a Table of Content for a document, construct a summary for a document if no TOC is available, retrieve a specific page, find textual content within a document, retrieve specific font styles or information, retrieve an image or illustration within a document, etc.

Figure 14 illustrates the relationship 350 of a parent 352 and a child node 354 within the document engine 320. The actual link between the parent 352 and the child 354 is a little more complex than a pointer between the two. Between them there is a so-called specifier 356 object whose role is to hold attributes of the link. By attributes of the link typically is meant information on the parent or children object IDS that are associated since this is a graph like structure and a single parent can have multiple children and a child can also have multiple parent objects.

The operation of the invention is to take provided input in the form of an electronic document structured or unstructured or data streams which embodies textual and non textual content typically represented by graphical imagery such as pictures or illustrations and drawing data represented in the form of vectors or commonly known as descriptions on how to draw a line from position A to B. Electronic documents are best described but not limited to business or personal correspondence letters, product information and representation

WO 02/07855

PCT/US02/09268

thereof, parts manuals, corporate or personal financial information and such like. **Figures 15** and **16** represent respectively a unstructured document and a structured document.

Figure 15 illustrates how a typical unstructured document **360** might look like. By unstructured is meant that the document has minimal navigation information and structured elements such as headers, footers, tables, and font style use upon which the invention can directly extract such format and navigational information for the purpose of delivering and reconstructing partial document content back to the invention's client side data processing system **250**. The server side **10** data processing capabilities then have to be employed to construct such navigational and reconstructing information by analyzing the unstructured document data.

The analyzation is done by detecting changes in font size, font type, font style and font positioning. By analyzing this all textual content elements **362** will be marked as headers by the following criteria's: largest used font size =14, font style = Bold, font type = Arial. This is reinforced by a text paragraph definition **364**, which all have font size =10, no font style and font type = Times. By identifying the headers **362** based on this information the invention can construct a navigational aid in the form of a Table of Contents where none exists for the unstructured document. An embedded image **366** will have the image insertion point coordinates x, y as well as a width and height information all in pixels. The system **10** of the invention can then provide a link to the image **366** when rendering this on the client side application processing system **12** and element **368** which is marked out because of a font style change within a text paragraph **364** will also be marked as a header, but ranked with a lower importance then the other headers **362** as it is positioned within the text paragraph **364**. The element **368** will then be used by the server side **10** application processing when creating a navigational aide in form of a Table of Contents as the element **368** is deemed to have navigational significance. Likewise elements of a type **370** and **372** will be detected by the

WO 02/077855

PCT/US02/09268

invention's server side processing capabilities as the fonts positioning of these elements are in close proximity in addition to the usage of the font styles Bold and Italic and finally usage of a Tabulator (TAB) character for spacing. These all indicate that this section has significance to the document 360 and can be marked as a navigational aide point when
5 construction a Table of Contents. Finally a page element 374 defines a page number description that is associated as a known navigational index reference for all the unstructured content residing on the documents page 1.

Figure 16 illustrates a structured document 380 that can be processed by the server side data processing system 10 with the purpose of identifying a document navigational
10 structure in the form of a Table of Contents and defining content excerpts with the intent to reconstruct it on the client side processing application 12. A Table of Contents (TOC) 382 is defined as an existing element in the structured document. Content elements 384 and 386 referenced as Heading 1 and Heading 2 are defined as paragraph headers 388 and are used as navigational elements in the construction of the TOC entity 382 by the document itself. The
15 server side data processing application 10 can therefore directly extract these elements without the processing required for an unstructured document. Further the System 10 will be parsing the structured document to identify a hyperlink 390 to a bookmark 1. A hyperlink is a navigational link between to objects or entities within a structured document indicating that they are connected and related and that by interacting with the hyperlink the user can
20 navigate to a different part of the document. The system 10 will make a record of the hyperlink entity 390 and use it for navigational purposes. A bookmark entity 392 is another element defined within the structured document 380 itself that provides a navigational anchor point that the system 10 will make a record of for the purpose of allowing the execution of the navigational hyperlink entity 390. Hyperlink entities 392, 394, 396 are different from the
25 hyperlink entity 390 by being external hyperlinks. External hyperlinks are defined as link

WO 02/077855

PCT/US02/09268

objects pointing to other content or external applications or processes defined within the structured document 380 itself.

One hyperlink 392 will launch an Internet Browser with the address pointing to a website address www.arizan.com, or second hyperlink 394 to www.air2web.com, while the link 396 will launch an email application program with the "To" address set to sales@arizan.com. The server side data processing system 10 will, while processing the structured document, record these elements in the constructed Document Object Model for navigational or possible application launch purposes for software applications employing the invention. An Image element 398 in the structured document 380 provides information on an image/picture and position x, y and width, height information that the System 10 invention will record in the DOM thus giving the client component 12 the ability to reconstruct this element on a wireless device 12 in the right visual and content context. A table element 400 as defined by the structured document 380 contains content displayed and organized in a table like fashion. The invention's server side data processing functionality will read this information and map it when constructing the DOM utilizing the Document Engines table component to allow the invention's client side processing system to reconstruct this in a table like structure on a wireless device/pda 12.

A graph element 402 in the structured document contains information in a graph type format such as bar chart and links to the data provided in the table element 400 used when drawing the chart by the application reading the structured document such as a word processor like MS Word by Microsoft Corporation. The server side data processing capabilities will retrieve and record the graphs link and rendering information when processing the structured document for the intent of having the client side data processing system reconstruct the graph on a wireless device/pda 12. A header element 404 defines page header information for each individual page in the structured document, while a footer

WO 02/07855

PCT/US02/09268

element 406 defines page footer information that will be reused for each single page in the structured document 380. The server side data processing capabilities will record the elements 404 and 406 and construct the appropriate entries in the DOM. A page entity 408 also provides a navigational index while navigating the structured document 380 as each page entity is a sequential entry that allows the user to look up an individual page within the page range starting with start page and ending with last page. The structured document as represented by Figure 16 also embeds non-visual descriptive information about the document referred to as metadata 410. The invention's server side data processing capabilities are able to extract this information and store it in the DOM where it can later be used by the system 10 to provide document information such as document name, author, date, time created etc. as defined by the metadata stored in the structured document.

Figure 17 illustrates the Document Object Model 420 as constructed by the invention's server side data processing system 10. The illustration 420 represents the load document's general structure 422. The DOM's entry point is a content node 424. Each of a plurality of containers 426 can have multiple components of different types. The invention server side data processing application has four component types defined: a table component 430, a text component 431, an image component 432 and a vector component (not illustrated). Each component can also be referenced by other components under the same container 426 or different containers in the form of a reference command

Figure 18 illustrates the text component 430 as employed by the server side data processing system 10 after constructing the Document Object Model (DOM) 420 and after loading the structured document 380 illustrated by Figure 16. Each text component 430 can have more than one section command 440, and as typically found for any standard structured or unstructured document, multiple paragraphs 442 can be included in one single section 440. The paragraph object 442 has one attribute 444 defined, which is a reference to a Table Of

WO 02/07855

PCT/US02/09268

Content hyperlink. A hyperlink start object 446 itself is defined by its attribute object 448, which describes the hyperlink ID and the destination of the link. Then a text segment object 450 has a describing attribute 452 that holds the textual content itself for the text object 450. At the end of the paragraph object 442 a Hyperlink End object 454 is constructed with an attribute object 456 containing the Hyperlink pair ID.

Figure 19A illustrates a Document Object Model 460 of a loaded structured document 380 and the DOM composition for a paragraph as employed by the server side data processing system. A paragraph object 462 has an attribute object 464 that describes the associated style and the textual content of the paragraph 462. A Font object 466 contains a font command and an associated attribute object 468 contains the font description and the font style information. The next command object linked to the paragraph object 462 is a Bookmark object 470 and it's descriptive attribute object 472, which contains the Bookmark ID. Next the paragraph object 462 is linked to a text object 474 and it's associated attribute object 476 with the value of the textual content itself. Figure 19B then continues the DOM paragraph composition 460 started by Figure 19A. The next command object linked to the paragraph object 462 is a text command object 478 and an associated attribute object 480 containing the text value. Next is a new font command object 482 and an associated attribute object 484 containing the font type and style definition. Then the following objects linked to the paragraph object 462 are respectively text command and font command objects and related attributes 486 - 496 until the paragraph composition ends with a reference command object 498 and an associated attribute object 500 containing the reference value.

Figure 20 illustrates the table component 428 as employed by the server side data processing system 10 after constructing the Document Object Model (DOM) and after loading the structured document 380 illustrated by Figure 16. Each table component 428 can have multiple row objects 510, 512, in which there might be one or more cells contained.

WO 02/077855

PCT/US02/09268

Figure 20 illustrates the first row of the sample table in the sample structured document **380**. The table row command object **510** links to a cell command object **514**, which has an associated attribute object **516** containing the textual content value of the cell. Additional cell command objects **518**, **520**, **522** and **524** will then be linked to the table row command object **510** until no more data is available for that row.

Figures 21A - 21F illustrate the server side data processing systems **10** description of content output in the form of a Universal Content Stream (UCS) as employed by the invention to transport data in a effective and optimized way for wireless delivery of the data over the Internet **16** to be processed by the client side data processing system **12**. **Figures 21A - 21F** has two tables each representing the server side processing capabilities in outputting a UCS data file representing commands to reproduce a structured document **380** and a unstructured document **360** as represented respectively in **Figures 16** and **15**.

Referring to **Figure 22**, a layout of one UCS format **530** is illustrated. The UCS (Universal Content Stream) format **530** provides the mechanism and data transport layer between the server **10** and the client technology or device **12**. The format primary purpose is to enable only the required visual image being sent in an optimal fashion over the available bandwidth. The format **530** supports the transfer of graphical images/pictures, 2D/3D drawings and various textual data such as word processor, and spreadsheet content.

The UCS can include several different standard formats for information transport in addition to the transcript command and data transfers to utilize the transcript technology and any available bandwidth. Embedding optional standard formats within UCS opens up for easy integration with client technology that do not require transcription support. And for formats such as image/raster data the existing/coming standards such as JPEG 2000 will offer adequate support for compression without data loss from the server **10** to client **12**.

WO 02/077855

PCT/US02/09268

Examples of the industry standard formats that can be accessed by both the server 10 and other clients directly are the following four types:

- XML for textual, metadata and content information
- SVG for vector information
- 5 • JPEG 2000 for images
- OpenGL 3D representation

Other standard formats or proprietary formats can be easily added or removed through the plugable and scalable architecture of the component based server 10.

The UCS format 530 includes three different data layers that together with optional industry standard formats make up the UCS format 530. The three proprietary layers are: a TCL layer, a UCS textual content layer 534, and a UCS image or vector data layer 536.

The TCL (Transcript Command Language) layer 532 provides a data channel for two-way communication between the server 10 and the client 12. The client 12 sends requests to the Client handler on the server side and receives a response with an transcript data stream in return. The TCL handles the following types of commands: Transcript requests for visual images, VQ (View and Query) commands, directory information, published document search/retrieve, and any required third party synchronization information. The TCL language itself is also constructed using XML for easy adaptation and future expansion.

20 The UCS vector stream 536 is a highly optimized vector format to provide vector-drawing capabilities on the thin client software. This enables the thin client software to offer vector representation for higher quality and accuracy on even small PDA's or handheld devices. The vector stream transfer and rendering is controlled through the TCL layer 532 which ensures that the Transcript handler will only generate a vector representation for the clipped viewable area requested by the client 12 and also to render it in an optimal fashion for

25

WO 02/077855

PCT/US02/09268

the limited client display area that generated the request. The UCS format 530 is designed in such a fashion that new proprietary layers or formats can easily be added. Any of the formats used for transportation of images, vector and text can also be replaced or added depending on integration requirements by third parties.

5 The Universal Content Stream (UCS) 530 is a binary file format specification, which enables delivery of original document content including, but not limited to textual content, image data, vector data and navigational information for the processed document. The UCS format is very efficient in overcoming the current limitation of the wireless devices 12 in respect to available bandwidth, which today in general is in the range of 9.6 to typically 48
10 although higher speeds are theoretically possible. The UCS addresses the limited bandwidth problem in a number of ways.

The UCS format 530 allows original document file sizes to be reduced by up to ninety percent (90%) while retaining all essential content and necessary format information for reconstructing the document on wireless devices using the client side processing capabilities.

15 This reduction in size is achieved by stripping out non-essential formatting information that is used by the original document application, e.g. MS Word for rich presentation on desktop computer. Non-essential formatting information might be use of different font styles, font colors, and revision information, etc. which typically have no value on the wireless device 12 as the device screen might be monochrome or grayscale and the device typically has only one
20 font type. The UCS file format also allows server side compression of the UCS file and client side decompression for further reduction in size. For compression common available technology in the form of compression algorithms such as LZW can be utilized. In addition the server side processing capabilities in producing transcripts (excerpt of content for transmission) allows the generation of multiple UCS files for individual transmission from
25 the server to the client in an iterative process where the resulting UCS file sent to the wireless

WO 02/077855

PCT/US02/09268

device is a result for a specific content query from the user. The iterative query process from the user on a wireless device requesting content from the server might typically result in the delivery of for example ten individual UCS files delivered to the device. The UCS file format includes information on how these individual UCS files link together so that the original document can then be reconstituted by the individual files in memory for a coherent viewing or reading experience. The UCS format also contains the Document Object Model (DOM) ID to ensure that the server 10 processing capabilities can delivery the correct DOM information to the requesting device 12. In addition the UCS file format also contains Internet hyperlinks allowing the client to request more content from within the document.

10 Command codes can also be embedded in the UCS format to facilitate the execution of specific processes by the server side data processing capabilities. This is typically a request for specific content, lookup in the navigation information stored within the DOM, etc.

WO 02/077855

PCT/US02/09268

DEFINITION OF TERMS

1. Anatomization – A specific technology process for parsing and analyzing a structured or unstructured document or data source with the purpose of establishing a uniform Document Object Model containing all content and a navigation document hierarchy.
- 5 2. Bookmarks – A navigational link associating a navigational reference from position X to position & within a document.
3. DOM – Document Object Model (DOM), a programming interface specification being developed by the World Wide Web Consortium, defines how to create and structure XML based documents as objects.
- 10 4. External File – reference to an associated external data or document file outside the original document.
5. External Object – reference to an associated external data object outside the original document.
6. Font Color – The color of the font.
- 15 7. Font position – The rendering x, y pixel or character positioning information.
8. Font Size – Font size in points indicating the size of a font character.
9. Font Style – The style of the font as in normal, italics, bold, underline, etc.
10. Font Type – Name of font type used in document e.g., Times Roman.
11. Footer – Footnote at the bottom of an individual or all pages in a document
- 20 containing content.

WO 02/077855

PCT/US02/09268

12. Header – Header at the top of an individual or all pages in a document containing content.
13. Hyperlinks – A navigational link external or internal to the document content referencing a bookmark or external data or content source.
- 5 14. Page – Defines an individual page in a document containing content information typically separated by a page break character of a fixed page or printer page size definition.
15. Paragraph – Collection of textual characters or lines of text grouped together in a cohesive body separated by line breaks.
- 10 16. Predictive Caching – A specific technology process designed to cache relevant or associated content on the server prior to required delivery/transmittal to a wireless/ wired device to minimize bandwidth and network latency.
17. Table of Content (TOC) – Presents a hierarchical navigational structure to the document content for informing and assisting in navigation of the document content.
- 15 18. Title Styles – Predefined styles in a document centric creation application to allow for the construction of a Table of Content such as Heading 1, Heading 2, Heading 3, etc.
19. Transcription – A specific technology process for delivering only the requested content excerpt and transmitting this to the client.
- 20 20. USC – The Universal Content Stream is a content data format optimized for minimized data size while preserving content context, navigational and presentation information for delivery to wireless/wired devices.

WO 02/077855

PCT/US02/09268

21. XML – extensible Markup Language is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

WO 02/077855

PCT/US02/09268

CLAIMS

1. A method for delivering content to a portable computing device, comprising:
receiving a request from the portable computing device for content associated with a
document remote to the portable computing device;
5 identifying one or more portions of the document associated with the requested
content;
reconstructing the identified portion or portions of the document from elements of the
document stored in a Document Object Model (DOM) Storage; and
transmitting the reconstructed identified portion or portions to the portable computing
10 device for display thereon.
2. The method of Claim 1, wherein identifying one or more portions of the
document includes utilizing a lookup or navigation tree to identify one or more portions of
the document.
3. The method of Claim 1, wherein identifying one or more portions of the
15 document includes utilizing one or more probability algorithms to determine what are the
most relevant portions of the document to the requested content.
4. The method of Claim 1, wherein reconstituting the identified portion or
portions includes reconstructing the identified portion or portions in a predetermined format
compatible with the portable computing device.
- 20 5. A method for publishing a document content on a remote portable computing
device, comprising:
receiving the document to be published, wherein the document includes visual
document representation and document content;
separating the visual document representation and the document content;
25 indexing the document content; and

WO 02/077855

PCT/US02/09268

establishing navigational links, which are mapped to a navigational tree; and storing the document content as Document Object Model structures.

6. The method of Claim 5, further including generating a metadata header for the document.

5 7. A method for wireless delivering document content to a portable computing device, comprising.

analyzing the content of a structured or unstructured document by identifying existing navigational elements or where none exist by analytical processing based on the documents visual representation to establish navigational links for the document content;

10 receiving a request from the portable computing device for content in said document;
using said navigational links to find and transmit said requested content; and
receiving said content at said portable computing device and reconstructing said content on said device.

8. The method of claim 7, including creating a table of contents during said
15 analysis and said portable computing device utilizing said table of contents to request said content.

9. The method of Claim 7, including storing said content in a document object model and obtaining said requested content from said document object model.

10. The method of Claim 7, including transmitting said content in a universal
20 content stream to said portable computing device.

11. The method of Claim 7, wherein analyzing said document includes anatomizing said document.

12. The method of Claim 7, including transmitting content excerpts of said document to said portable computing device.

WO 02/077855

PCT/US02/09268

13. The method of Claim 12, including predictive caching of said content excerpts relating to said content request and said content excerpt previously sent.
14. A system for wireless document content delivery to a portable computing device, comprising:
- 5 a communication bus having an input and output coupled thereto.
a document processing module coupled to said bus and to a document data base;
a preprocessing module coupled to said bus;
an anatomizing module coupled to said bus;
an indexing module coupled to said bus and to an indexing database; and
10 a transcript module coupled to said bus and to a transcript cache.
15. The system of Claim 14, including a predictive cache coupled to said transcript module and to said bus.

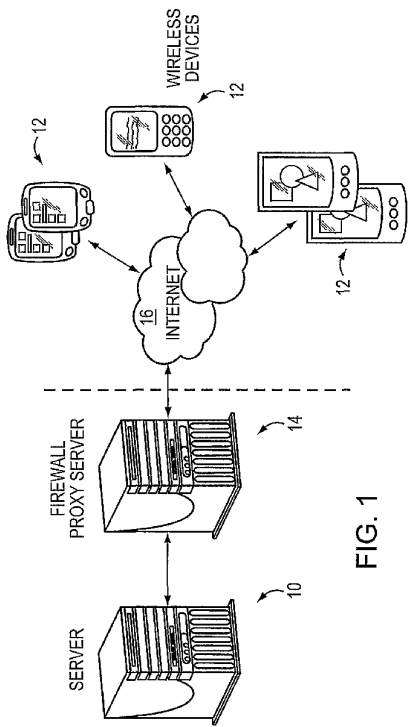


FIG. 1

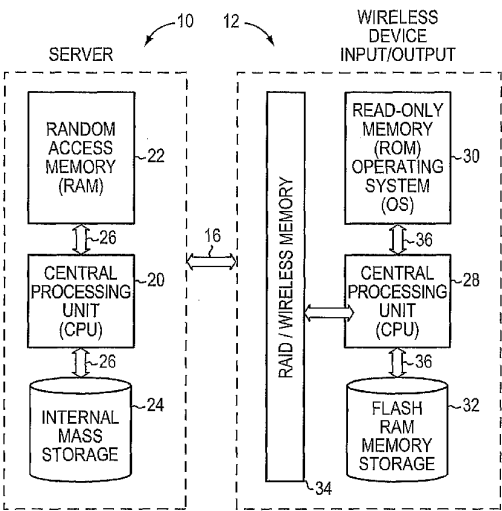


FIG. 2

WO 02/077855

PCT/US02/09268

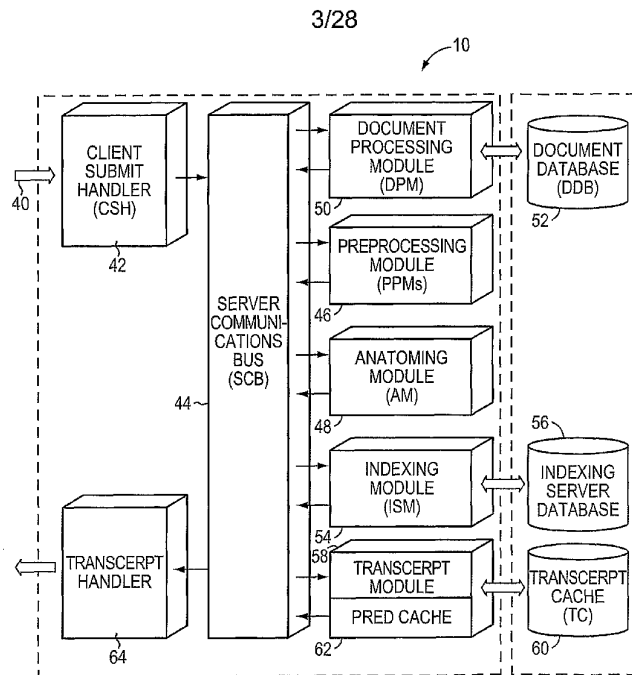


FIG. 3

WO 02/077855

PCT/US02/09268

4/28

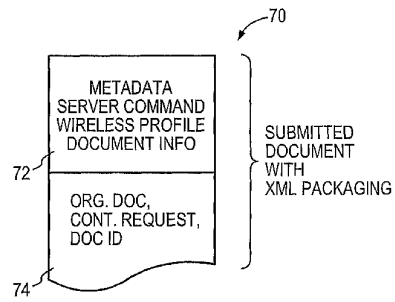


FIG. 4

WO 02/077855

PCT/US02/09268

5/28

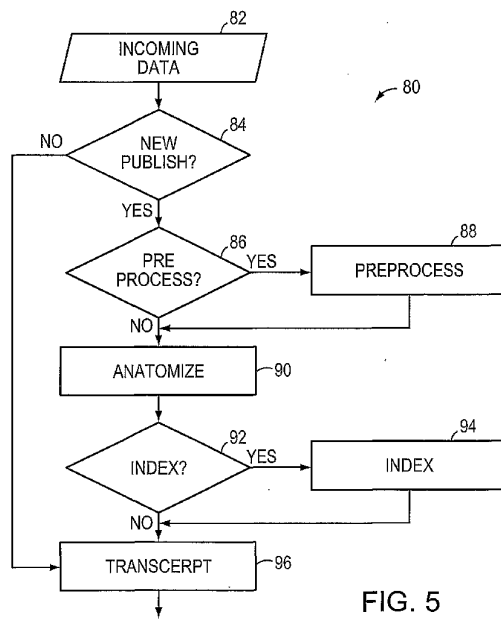


FIG. 5

6/28

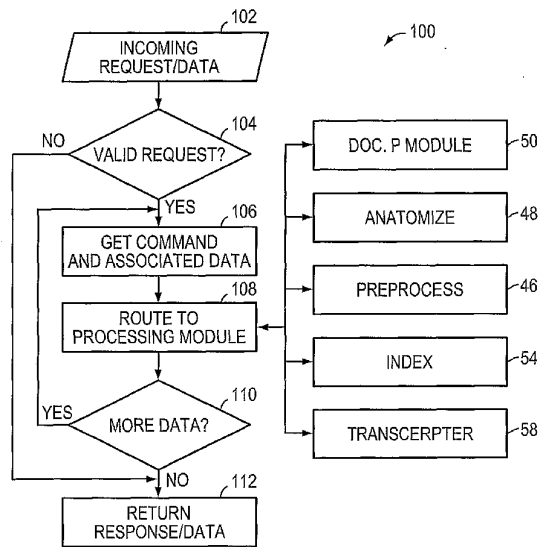


FIG. 6

WO 02/077855

PCT/US02/09268

7/28

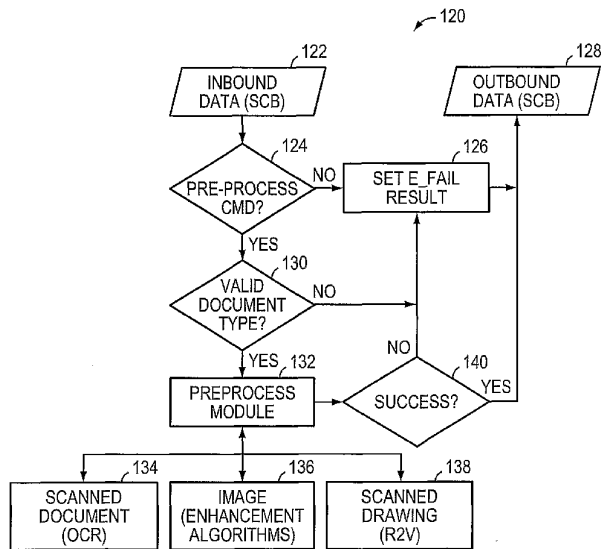


FIG. 7

WO 02/07855

PCT/US02/09268

8/28

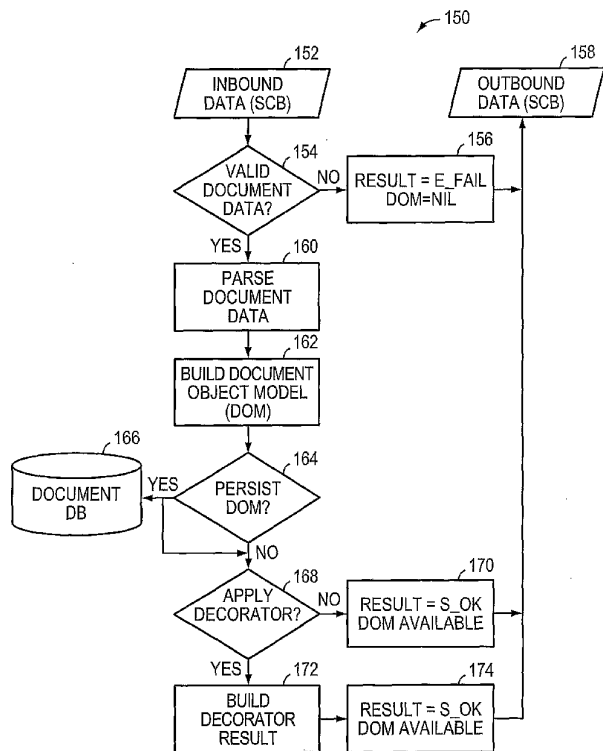


FIG. 8

WO 02/077855

PCT/US02/09268

9/28

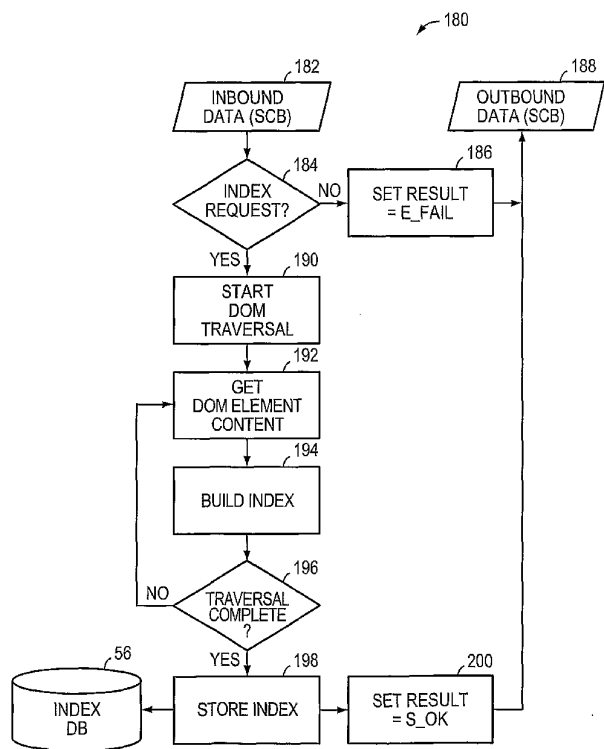


FIG. 9

WO 02/077855

PCT/US02/09268

10/28

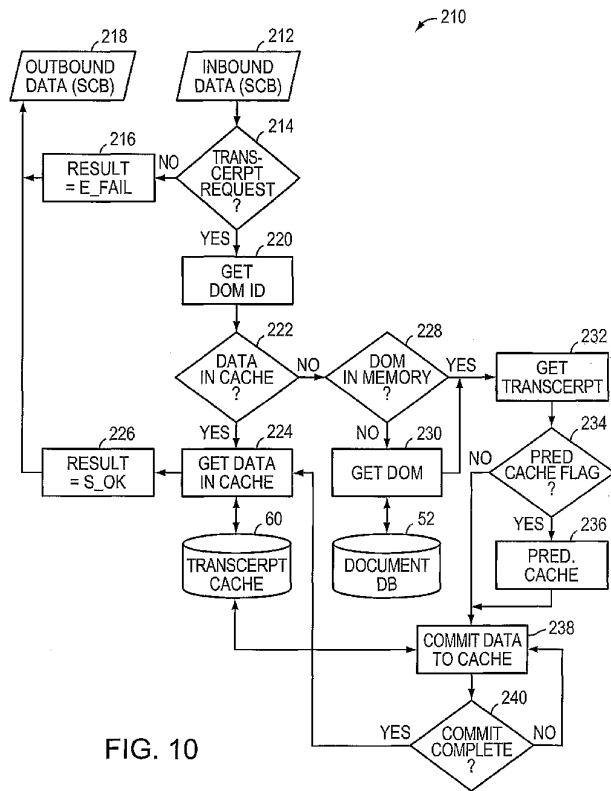


FIG. 10

SUBSTITUTE SHEET (RULE 26)

WO 02/077855

PCT/US02/09268

11/28

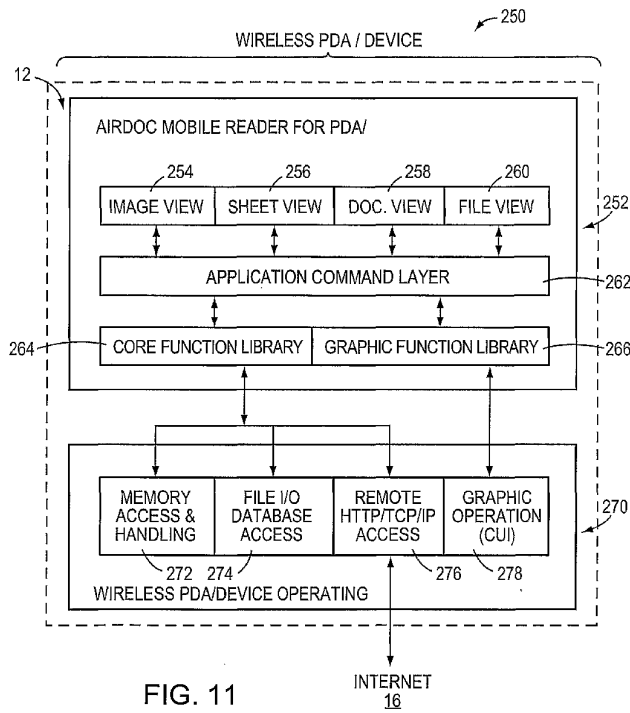
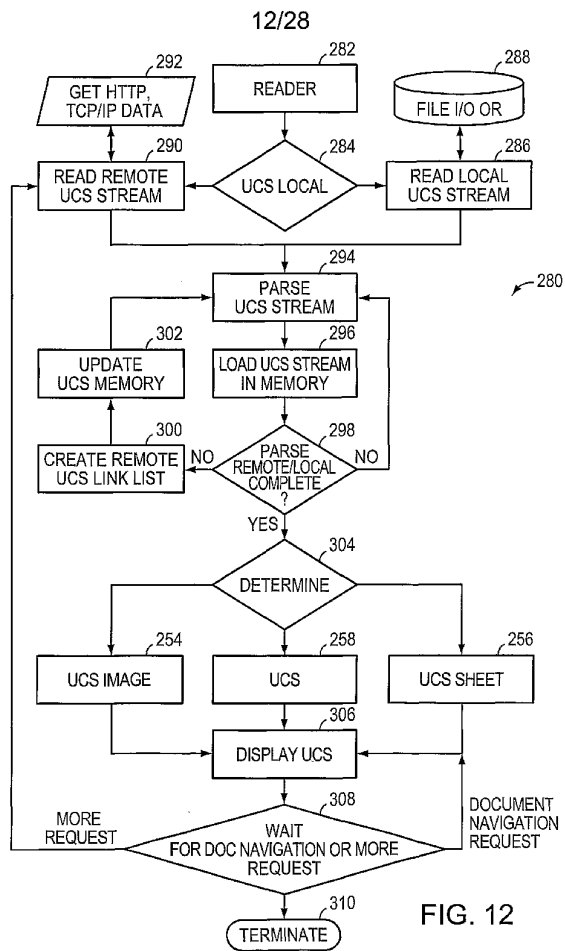


FIG. 11

SUBSTITUTE SHEET (RULE 26)

WO 02/07855

PCT/US02/09268



SUBSTITUTE SHEET (RULE 26)

WO 02/077855

PCT/US02/09268

13/28

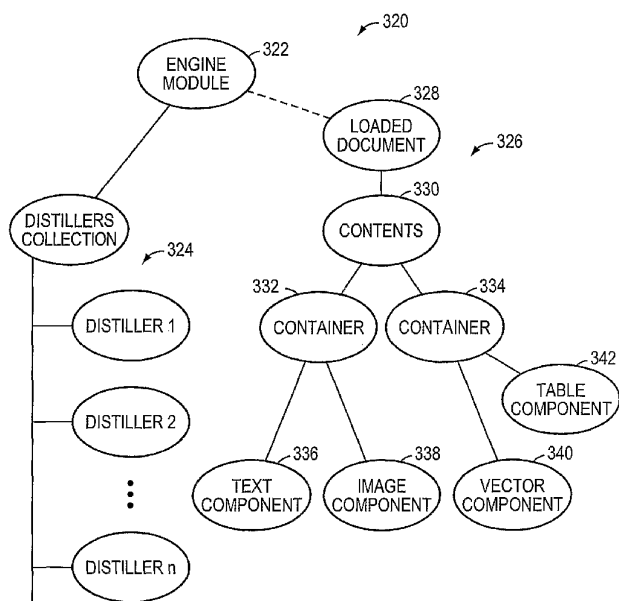
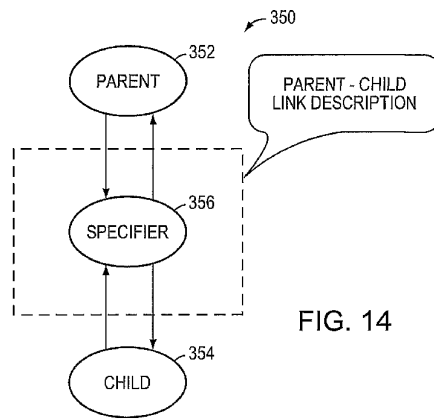


FIG. 13

WO 02/077855

PCT/US02/09268

14/28



15/28

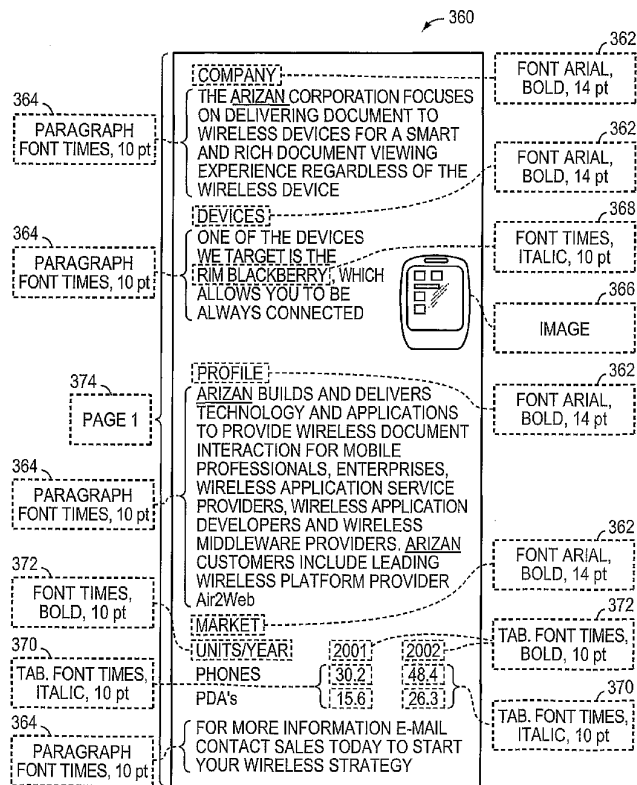


FIG. 15

16/28

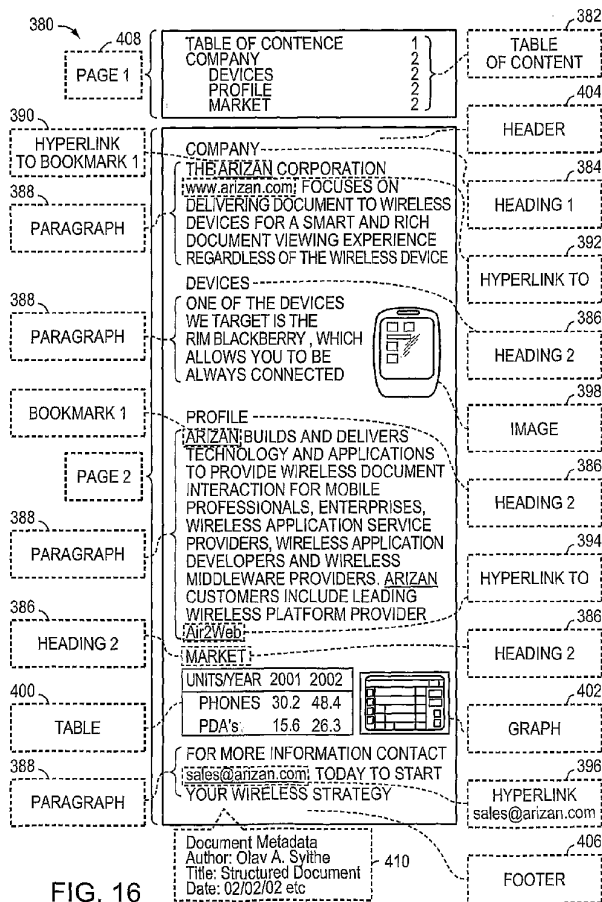


FIG. 16

WO 02/07855

PCT/US02/09268

17/28

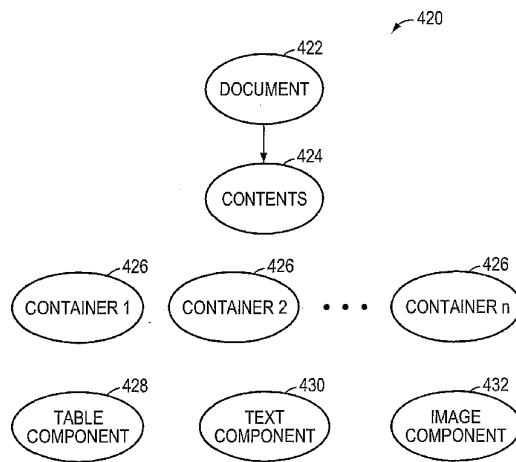


FIG. 17

WO 02/077855

PCT/US02/09268

18/28

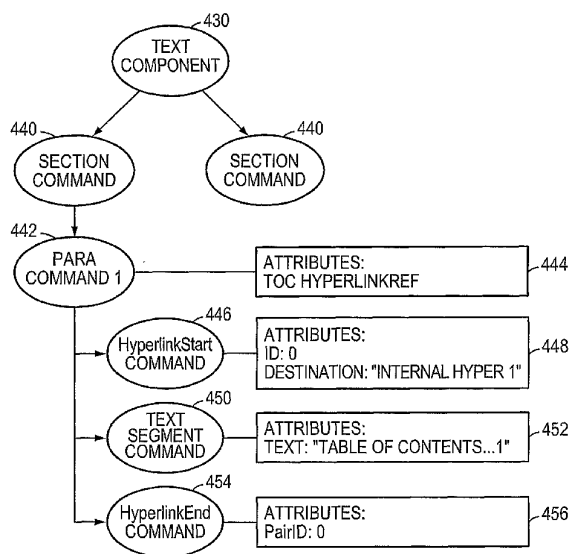


FIG. 18

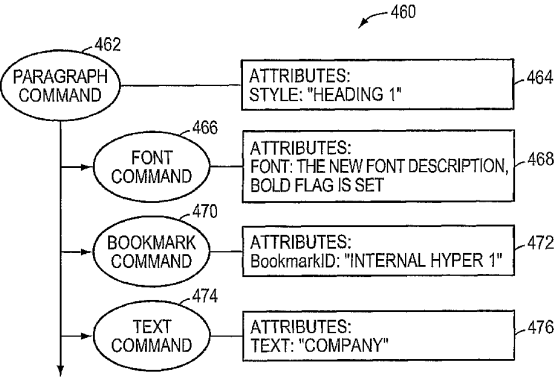


FIG. 19A

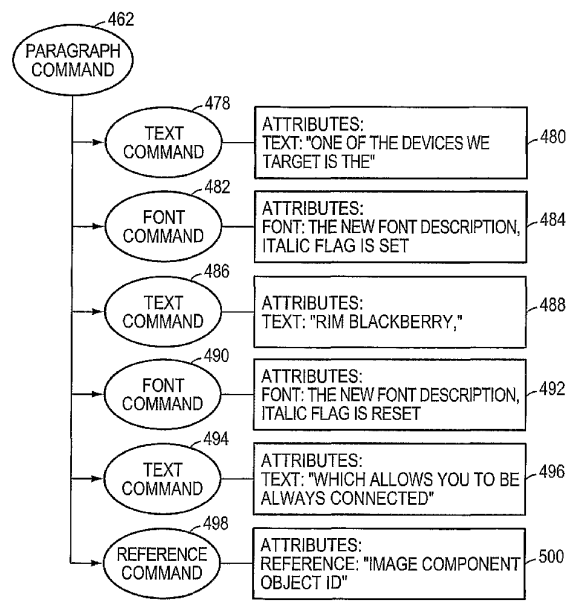


FIG. 19B

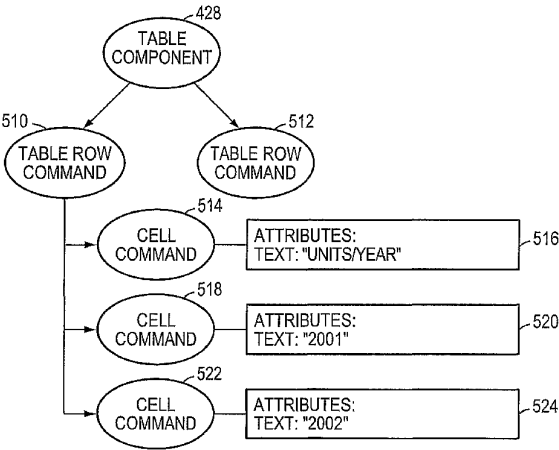


FIG. 20

WO 02/07855

PCT/US02/09268

22/28

380		360	
RECORD ID	STRUCTURED DOC	RECORD ID	UNSTRUCTURED DOC
1	UCS SIGNATURE	1	UCS SIGNATURE
2	DOCUMENT HEADER	2	DOCUMENT HEADER
3	DOCUMENT CONTAINER	3	SECTION CONTAINER
4	PARAGRAPH	4	PARAGRAPH
5	PARAGRAPH INFORMATION	5	PARAGRAPH INFORMATION
6	BOOKMARK	6	BOOKMARK
7	SIMPLE FONT CHANGE	7	SIMPLE FONT CHANGE
8	TEXT SEGMENT	8	TEXT SEGMENT
9	PARAGRAPH	9	PARAGRAPH
10	PARAGRAPH INFORMATION	10	PARAGRAPH INFORMATION
11	PARAGRAPH	11	PARAGRAPH
12	PARAGRAPH INFORMATION	12	PARAGRAPH INFORMATION
13	HYPERLINK BEGIN	13	HYPERLINK BEGIN
14	SIMPLE FONT CHANGE	14	SIMPLE FONT CHANGE
15	TEXT SEGMENT	15	TEXT SEGMENT
16	SIMPLE FONT CHANGE	16	SIMPLE FONT CHANGE
17	TEXT SEGMENT	17	TEXT SEGMENT
18	TEXT SEGMENT	18	TEXT SEGMENT
19	HYPERLINK END	19	HYPERLINK END
20	PARAGRAPH	20	REFERENCE 1 - TEXT BOX
21	PARAGRAPH INFORMATION	21	PARAGRAPH
22	HYPERLINK BEGIN	22	PARAGRAPH INFORMATION
23	SIMPLE FONT CHANGE	23	HYPERLINK BEGIN
24	TEXT SEGMENT	24	SIMPLE FONT CHANGE
25	SIMPLE FONT CHANGE	25	TEXT SEGMENT
26	TEXT SEGMENT	26	SIMPLE FONT CHANGE
27	TEXT SEGMENT	27	TEXT SEGMENT
28	HYPERLINK END	28	TEXT SEGMENT
29	PARAGRAPH	29	HYPERLINK END
30	PARAGRAPH INFORMATION	30	PARAGRAPH
31	BOOKMARK	31	PARAGRAPH INFORMATION
32	HYPERLINK BEGIN	32	BOOKMARK
33	SIMPLE FONT CHANGE	33	HYPERLINK BEGIN
34	TEXT SEGMENT	34	SIMPLE FONT CHANGE
35	SIMPLE FONT CHANGE	35	TEXT SEGMENT
36	TEXT SEGMENT	36	SIMPLE FONT CHANGE
37	TEXT SEGMENT	37	TEXT SEGMENT
38	HYPERLINK END	38	TEXT SEGMENT
39	PARAGRAPH	39	HYPERLINK END
40	PARAGRAPH INFORMATION	40	REFERENCE 2 - IMAGE

FIG. 21A

SUBSTITUTE SHEET (RULE 26)

WO 02/07855

PCT/US02/09268

23/28

41	HYPERLINK BEGIN	41	PARAGRAPH
42	SIMPLE FONT CHANGE	42	PARAGRAPH INFORMATION
43	TEXT SEGMENT	43	HYPERLINK BEGIN
44	SIMPLE FONT CHANGE	44	SIMPLE FONT CHANGE
45	TEXT SEGMENT	45	TEXT SEGMENT
46	TEXT SEGMENT	46	SIMPLE FONT CHANGE
47	HYPERLINK END	47	TEXT SEGMENT
48	PARAGRAPH	48	TEXT SEGMENT
49	PARAGRAPH INFORMATION	49	HYPERLINK END
50	HYPERLINK BEGIN	50	PARAGRAPH
51	SIMPLE FONT CHANGE	51	PARAGRAPH INFORMATION
52	TEXT SEGMENT	52	HYPERLINK BEGIN
53	SIMPLE FONT CHANGE	53	SIMPLE FONT CHANGE
54	TEXT SEGMENT	54	TEXT SEGMENT
55	TEXT SEGMENT	55	SIMPLE FONT CHANGE
56	HYPERLINK END	56	TEXT SEGMENT
57	PARAGRAPH	57	TEXT SEGMENT
58	PARAGRAPH INFORMATION	58	HYPERLINK END
59	PARAGRAPH	59	PARAGRAPH
60	PARAGRAPH INFORMATION	60	PARAGRAPH INFORMATION
61	SIMPLE FONT CHANGE	61	PARAGRAPH
62	TEXT SEGMENT	62	PARAGRAPH INFORMATION
63	PARAGRAPH	63	SIMPLE FONT CHANGE
64	PARAGRAPH INFORMATION	64	TEXT SEGMENT
65	SIMPLE FONT CHANGE	65	PARAGRAPH
66	TEXT SEGMENT	66	PARAGRAPH INFORMATION
67	HYPERLINK BEGIN	67	SIMPLE FONT CHANGE
68	SIMPLE FONT CHANGE	68	TEXT SEGMENT
69	TEXT SEGMENT	69	HYPERLINK BEGIN
70	HYPERLINK END	70	SIMPLE FONT CHANGE
71	SIMPLE FONT CHANGE	71	TEXT SEGMENT
72	TEXT SEGMENT	72	HYPERLINK END
73	HYPERLINK BEGIN	73	SIMPLE FONT CHANGE
74	SIMPLE FONT CHANGE	74	TEXT SEGMENT
75	TEXT SEGMENT	75	HYPERLINK BEGIN
76	BOOKMARK	76	SIMPLE FONT CHANGE
77	TEXT SEGMENT	77	TEXT SEGMENT
78	HYPERLINK END	78	BOOKMARK
79	SIMPLE FONT CHANGE	79	TEXT SEGMENT
80	TEXT SEGMENT	80	HYPERLINK END
81	PARAGRAPH	81	SIMPLE FONT CHANGE
82	PARAGRAPH INFORMATION	82	TEXT SEGMENT
83	REFERENCE 1 - TEXT BOX	83	PARAGRAPH
84	REFERENCE 2 - IMAGE	84	PARAGRAPH INFORMATION
85	SIMPLE FONT CHANGE	85	SIMPLE FONT CHANGE

FIG. 21B

SUBSTITUTE SHEET (RULE 26)

WO 02/077855

PCT/US02/09268

24/28

86	TEXT SEGMENT	86	TEXT SEGMENT
87	PARAGRAPH	87	PARAGRAPH
88	PARAGRAPH INFORMATION	88	PARAGRAPH INFORMATION
89	SIMPLE FONT CHANGE	89	SIMPLE FONT CHANGE
90	TEXT SEGMENT	90	TEXT SEGMENT
91	PARAGRAPH	91	PARAGRAPH
92	PARAGRAPH INFORMATION	92	PARAGRAPH INFORMATION
93	SIMPLE FONT CHANGE	93	SIMPLE FONT CHANGE
94	TEXT SEGMENT	94	TEXT SEGMENT
95	TEXT SEGMENT	95	TEXT SEGMENT
96	HYPERLINK BEGIN	96	HYPERLINK BEGIN
97	SIMPLE FONT CHANGE	97	SIMPLE FONT CHANGE
98	TEXT SEGMENT	98	TEXT SEGMENT
99	HYPERLINK END	99	HYPERLINK END
100	PARAGRAPH	100	PARAGRAPH
101	PARAGRAPH INFORMATION	101	PARAGRAPH INFORMATION
102	SIMPLE FONT CHANGE	102	SIMPLE FONT CHANGE
103	TEXT SEGMENT	103	TEXT SEGMENT
104	PARAGRAPH	104	PARAGRAPH
105	PARAGRAPH INFORMATION	105	PARAGRAPH INFORMATION
106	SIMPLE FONT CHANGE	106	SIMPLE FONT CHANGE
107	TEXT SEGMENT	107	TEXT SEGMENT
108	TEXT SEGMENT	108	TEXT SEGMENT
109	TEXT SEGMENT	109	TEXT SEGMENT
110	PARAGRAPH	110	PARAGRAPH
111	PARAGRAPH INFORMATION	111	PARAGRAPH INFORMATION
112	REFERENCE 3 - IMAGE	112	SIMPLE FONT CHANGE
113	SIMPLE FONT CHANGE	113	TEXT SEGMENT
114	TEXT SEGMENT	114	PARAGRAPH
115	REFERENCE 4 - SPREADSHEET	115	PARAGRAPH INFORMATION
116	PARAGRAPH	116	SIMPLE FONT CHANGE
117	PARAGRAPH INFORMATION	117	TEXT SEGMENT
118	SIMPLE FONT CHANGE	118	PARAGRAPH
119	TEXT SEGMENT	119	PARAGRAPH INFORMATION
120	PARAGRAPH	120	PARAGRAPH
121	PARAGRAPH INFORMATION	121	PARAGRAPH INFORMATION
122	PARAGRAPH	122	SIMPLE FONT CHANGE
123	PARAGRAPH INFORMATION	123	TEXT SEGMENT
124	SIMPLE FONT CHANGE	124	TEXT SEGMENT
125	TEXT SEGMENT	125	PARAGRAPH
126	TEXT SEGMENT	126	PARAGRAPH INFORMATION
127	PARAGRAPH	127	SIMPLE FONT CHANGE
128	PARAGRAPH INFORMATION	128	TEXT SEGMENT
129	SIMPLE FONT CHANGE	129	PARAGRAPH
130	TEXT SEGMENT	130	PARAGRAPH INFORMATION

FIG. 21C

SUBSTITUTE SHEET (RULE 26)

WO 02/07855

PCT/US02/09268

25/28

131	PARAGRAPH	131	SIMPLE FONT CHANGE
132	PARAGRAPH INFORMATION	132	TEXT SEGMENT
133	SIMPLE FONT CHANGE	133	PARAGRAPH
134	TEXT SEGMENT	134	PARAGRAPH INFORMATION
135	PARAGRAPH	135	PARAGRAPH
136	PARAGRAPH INFORMATION	136	PARAGRAPH INFORMATION
137	PARAGRAPH	137	SIMPLE FONT CHANGE
138	PARAGRAPH INFORMATION	138	TEXT SEGMENT
139	SIMPLE FONT CHANGE	139	TEXT SEGMENT
140	TEXT SEGMENT	140	PARAGRAPH
141	TEXT SEGMENT	141	PARAGRAPH INFORMATION
142	PARAGRAPH	142	SIMPLE FONT CHANGE
143	PARAGRAPH INFORMATION	143	TEXT SEGMENT
144	SIMPLE FONT CHANGE	144	PARAGRAPH
145	TEXT SEGMENT	145	PARAGRAPH INFORMATION
146	PARAGRAPH	146	SIMPLE FONT CHANGE
147	PARAGRAPH INFORMATION	147	TEXT SEGMENT
148	SIMPLE FONT CHANGE	148	PARAGRAPH
149	TEXT SEGMENT	149	PARAGRAPH INFORMATION
150	PARAGRAPH	150	PARAGRAPH
151	PARAGRAPH INFORMATION	151	PARAGRAPH INFORMATION
152	PARAGRAPH	152	PARAGRAPH
153	PARAGRAPH INFORMATION	153	PARAGRAPH INFORMATION
154	PARAGRAPH	154	SIMPLE FONT CHANGE
155	PARAGRAPH INFORMATION	155	TEXT SEGMENT
156	SIMPLE FONT CHANGE	156	HYPERLINK BEGIN
157	TEXT SEGMENT	157	SIMPLE FONT CHANGE
158	HYPERLINK BEGIN	158	TEXT SEGMENT
159	SIMPLE FONT CHANGE	159	UCS SIGNATURE
160	TEXT SEGMENT	160	DOCUMENT HEADER
161	HYPERLINK END	161	DOCUMENT CONTAINER
162	SIMPLE FONT CHANGE	162	PARAGRAPH
163	TEXT SEGMENT	163	PARAGRAPH INFORMATION
164	TEXT SEGMENT	164	BOOKMARK
165	REFERENCE CONTAINERS	165	SIMPLE FONT CHANGE
166	REFERENCE 1	166	TEXT SEGMENT
167	REFERENCE 2	167	PARAGRAPH
168	REFERENCE 3	168	PARAGRAPH INFORMATION
169	REFERENCE 4	169	SIMPLE FONT CHANGE
170	END OF DOCUMENT	170	TEXT SEGMENT
		171	PARAGRAPH
		172	PARAGRAPH INFORMATION
		173	PARAGRAPH
		174	PARAGRAPH INFORMATION
		175	BOOKMARK

FIG. 21D

SUBSTITUTE SHEET (RULE 26)

WO 02/077855

PCT/US02/09268

26/28

360

176	SIMPLE FONT CHANGE
177	TEXT SEGMENT
178	PARAGRAPH
179	PARAGRAPH INFORMATION
180	PARAGRAPH
181	PARAGRAPH INFORMATION
182	BOOKMARK
183	BOOKMARK
184	SIMPLE FONT CHANGE
185	TEXT SEGMENT
186	PARAGRAPH
187	PARAGRAPH INFORMATION
188	SIMPLE FONT CHANGE
189	TEXT SEGMENT
190	PARAGRAPH
191	PARAGRAPH INFORMATION
192	BOOKMARK
193	SIMPLE FONT CHANGE
194	TEXT SEGMENT
195	PARAGRAPH
196	PARAGRAPH INFORMATION
197	PARAGRAPH
198	PARAGRAPH INFORMATION
199	SIMPLE FONT CHANGE
200	TEXT SEGMENT
201	PARAGRAPH
202	PARAGRAPH INFORMATION
203	SIMPLE FONT CHANGE
204	TEXT SEGMENT
205	SIMPLE FONT CHANGE
206	TEXT SEGMENT
207	SIMPLE FONT CHANGE
208	TEXT SEGMENT
209	SIMPLE FONT CHANGE
210	TEXT SEGMENT
211	PARAGRAPH
212	PARAGRAPH INFORMATION
213	SIMPLE FONT CHANGE
214	TEXT SEGMENT
215	SIMPLE FONT CHANGE
216	TEXT SEGMENT
217	SIMPLE FONT CHANGE
218	TEXT SEGMENT
219	SIMPLE FONT CHANGE
220	TEXT SEGMENT

FIG. 21E

SUBSTITUTE SHEET (RULE 26)

WO 02/077855

PCT/US02/09268

27/28

↖ 360

221	PARAGRAPH
222	PARAGRAPH INFORMATION
223	PARAGRAPH
224	PARAGRAPH INFORMATION
225	SIMPLE FONT CHANGE
226	TEXT SEGMENT
227	PARAGRAPH
228	PARAGRAPH INFORMATION
229	SIMPLE FONT CHANGE
230	TEXT SEGMENT
231	PARAGRAPH
232	PARAGRAPH INFORMATION
233	SIMPLE FONT CHANGE
234	TEXT SEGMENT
235	SIMPLE FONT CHANGE
236	TEXT SEGMENT
237	PARAGRAPH
238	PARAGRAPH INFORMATION
239	PARAGRAPH
240	PARAGRAPH INFORMATION
241	REFERENCE CONTAINERS
242	REFERENCE 1
243	REFERENCE 2
244	END OF DOCUMENT

FIG. 21F

WO 02/077855

PCT/US02/09268

28/28

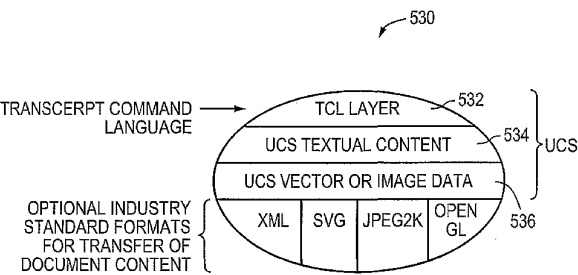


FIG. 22

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US02/09268
A. CLASSIFICATION OF SUBJECT MATTER IPC(7) : G05F 1/721 US CL : 707/513 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 707/513; 709/246		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WEST USPAT		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,B	US 6,404,445 B1 (GALEA et al) 11 June 2002 (11.06.2002).	1-15
A,E	US 6,401,132 B1 (BELLWOOD et al) 04 June 2002 (04.06.2002).	1-15
A,E	US 6,389,467 B1 (EYAL) 14 May 2002 (15.05.2002).	1-15
A,B	US 6,377,957 B1 (JEYARAMAN) 23 April 2002 (23.04.2002).	1-15
A,P	US 6,311,187 B1 (JEYARAMAN) 30 October 2001 (30.10.2001).	1-15
<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: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent 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 12 June 2002 (12.06.2002)		Date of mailing of the international search report 08 JUL 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-5230		Authorized officer JOSEPH H FIELD <i>Joseph H. Field</i> Telephone No. (703)305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT, BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,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,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE, GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,PL,PT,R O,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,UZ,VN,YU,ZA,ZW

(72)発明者 シルス, オラヴ エイ.

アメリカ合衆国 ジョージア 30338, アトランタ, アシュフォード ダンウッドイー
ロード 4867 ナンバー6208

Fターム(参考) 5B009 QA06 SA03 VC01