(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2005-527881
(P2005-527881A)
(43)公表日　平成17年9月15日(2005.9.15)

(51)Int.Cl.[7]	F I	テーマコード（参考）
G06F 13/00	G06F 13/00 550L	

審査請求 未請求　予備審査請求 有　（全 50 頁）

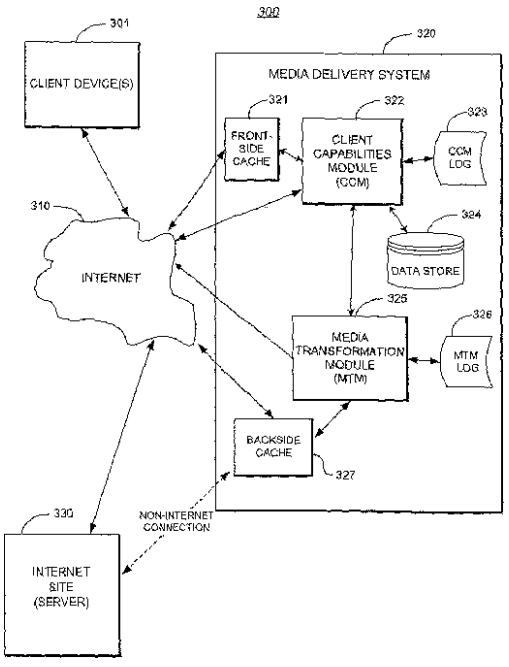
(21)出願番号	特願2003-542457 (P2003-542457)
(86)(22)出願日	平成14年11月7日 (2002.11.7)
(85)翻訳文提出日	平成16年7月7日 (2004.7.7)
(86)国際出願番号	PCT/US2002/036064
(87)国際公開番号	WO2003/040893
(87)国際公開日	平成15年5月15日 (2003.5.15)
(31)優先権主張番号	10/010,616
(32)優先日	平成13年11月8日 (2001.11.8)
(33)優先権主張国	米国 (US)

(71)出願人　500421370
ライトサーフ　テクノロジーズ　インコーポレイテッド
アメリカ合衆国　９５０６０　カリフォルニア州　サンタ　クルッツ　クーパー　ストリート　１１０　フォース　フロアー
(74)代理人　100082005
弁理士　熊倉　禎男
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100074228
弁理士　今城　俊夫
(74)代理人　100086771
弁理士　西島　孝喜

最終頁に続く

(54)【発明の名称】多数の異なるクライアントデバイスへそれらの能力に基づいてメディアを配送するシステム及び方法

(57)【要約】
オンザフライのメディア再フォーマッティングを進歩型クライアント検出能力と結合して、適切なメディアコンテンツを、接続されたクライアントデバイスに配送できるようにすることを含むオンラインメディア配送システムが開示される。このシステムは、メディア文書又はオブジェクトに対してクライアントデバイスから要求を受け取り、特定のメディアオブジェクトを要求しているクライアントデバイスをＨＴＴＰ要求から識別し、クライアントデバイスのメディア出力能力を決定し、それらの能力に基づいてソースメディアを再フォーマットし、そしてその再フォーマットされたメディアをクライアントデバイスへ配送する。このシステムは、異なるハードウェア及びソフトウェア能力の多数の種々のクライアントデバイスに対してメディアコンテンツへのアクセスを与える。このシステムは、接続能力をもつ実際上いかなるデバイスにも適当なメディアコンテンツを配送できるようにする。



【特許請求の範囲】
【請求項１】
オンラインシステムにおいて、クライアントデバイスの能力を決定し、そのようなデバイスに適したフォーマットでメディアコンテンツを供給するための方法が、
特定メディアオブジェクトのコピーをターゲットデバイスに与える要求を受け取り、
上記ターゲットデバイスの能力を決定し、
上記ターゲットデバイスの能力に基づいて、上記メディアオブジェクトのコピーを上記ターゲットデバイスに与えるために必要なフォーマットを決定し、
上記特定メディアオブジェクトを、上記決定されたフォーマットを有するコピーへ変換し、そして
上記決定されたフォーマットを有する上記コピーを上記ターゲットデバイスに与える、
という段階を備えた方法。
【請求項２】
上記決定されたフォーマットを有する上記コピーをキャッシュメモリに記憶する段階を更に備えた請求項１に記載の方法。
【請求項３】
上記決定されたフォーマットでの上記特定オブジェクトに対する後続の要求をターゲットデバイスから受け取り、そして
上記キャッシュメモリに記憶された上記コピーを上記ターゲットデバイスに与える、
という段階を備えた請求項２に記載の方法。
【請求項４】
上記メディアオブジェクトを変換するために上記特定メディアオブジェクトのコピーを、接続されたサーバーから得るという段階を更に備えた請求項１に記載の方法。
【請求項５】
上記接続されたサーバーから受け取られた上記メディアオブジェクトのキャッシュコピーをキャッシュメモリに記憶し、そして
上記メディアオブジェクトを変換するための後続の要求に応答して、上記キャッシュメモリに記憶された上記メディアオブジェクトの上記コピーを使用する、
という段階を更に備えた請求項４に記載の方法。
【請求項６】
上記ターゲットデバイスの能力は、スクリーン解像度を含む請求項１に記載の方法。
【請求項７】
上記ターゲットデバイスの能力は、スクリーンサイズを含む請求項１に記載の方法。
【請求項８】
上記ターゲットデバイスの能力は、カラーサポートを含む請求項１に記載の方法。
【請求項９】
上記ターゲットデバイスの能力は、ビットレートを含む請求項１に記載の方法。
【請求項１０】
上記ターゲットデバイスの能力は、上記ターゲットデバイスがその要求を送信するために使用する現在使用可能な通信メディアを含む請求項１に記載の方法。
【請求項１１】
上記現在使用可能な通信メディアは、ワイヤレス通信を含む請求項１０に記載の方法。
【請求項１２】
上記現在使用可能な通信メディアは、有線通信を含む請求項１０に記載の方法。
【請求項１３】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにより提出された要求を検討することを含む請求項１に記載の方法。
【請求項１４】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにより提出されたＨＴＴＰヘッダを検討することを含む請求項１に記載の方法。

【請求項１５】
上記デバイスにより提出されたＨＴＴＰヘッダを検討することは、ＨＴＴＰユーザーエージェントヘッダを検討することを含む請求項１４に記載の方法。
【請求項１６】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにその能力について問合せることを含む請求項１に記載の方法。
【請求項１７】
ターゲットデバイスの能力を決定する上記段階は、上記ターゲットデバイスのデバイスクラスに基づき知識ベースから能力を決定することを含む請求項１に記載の方法。
【請求項１８】
上記デバイスの能力を上記知識ベースに追加できるようにするために確認されていないターゲットデバイスのログ記録を記録する段階を更に備えた請求項１７に記載の方法。
【請求項１９】
確認されていない上記ターゲットデバイスに関する通知を自動的に発生する段階を更に備えた請求項１８に記載の方法。
【請求項２０】
必要なフォーマットを決定する上記段階は、上記ターゲットデバイスにおいて特定のイメージをレンダリングするための適当な解像度を決定することを含む請求項１に記載の方法。
【請求項２１】
必要なフォーマットを決定する上記段階は、上記ターゲットデバイスにおいて特定のイメージをレンダリングするための適当なカラースペースを決定することを含む請求項１に記載の方法。
【請求項２２】
必要なフォーマットを決定する上記段階は、上記ターゲットデバイスにおいて特定のイメージをレンダリングするための適当なイメージサイズを決定することを含む請求項１に記載の方法。
【請求項２３】
必要なフォーマットを決定する上記段階は、上記ターゲットデバイスの表示のアスペクト比に合致するように特定イメージを回転すべきかどうか決定することを含む請求項１に記載の方法。
【請求項２４】
必要なフォーマットを決定する上記段階は、上記ターゲットデバイスに対する適当なビットレートを決定することを含む請求項１に記載の方法。
【請求項２５】
必要なフォーマットを決定する上記段階は、特定メディアオブジェクトのコピーを上記ターゲットデバイスへ送信するのに使用できる通信帯域巾を決定することを含む請求項１に記載の方法。
【請求項２６】
上記使用できる通信帯域巾は、上記ターゲットデバイスから受け取られるＨＴＴＰ要求ヘッダに少なくとも一部分基づいて決定される請求項２５に記載の方法。
【請求項２７】
上記使用できる通信帯域巾は、上記ターゲットデバイスのデバイスクラスに少なくとも一部分基づいて決定される請求項２５に記載の方法。
【請求項２８】
上記ターゲットデバイスは、表示能力を有するハンドヘルド計算デバイスを含む請求項１に記載の方法。
【請求項２９】
上記ターゲットデバイスは、デジタルオーディオ能力を有するハンドヘルド計算デバイスを含む請求項１に記載の方法。

【請求項３０】
上記ターゲットデバイスは、表示能力を有するセルラー電話デバイスを含む請求項１に記載の方法。
【請求項３１】
上記ターゲットデバイスは、デジタルオーディオ能力を有するセルラー電話デバイスを含む請求項１に記載の方法。
【請求項３２】
上記ターゲットデバイスは、表示能力を有するページャーデバイスを含む請求項１に記載の方法。
【請求項３３】
上記ターゲットデバイスは、表示能力を有するパーソナルコンピュータを含む請求項１に記載の方法。
【請求項３４】
上記ターゲットデバイスは、デジタルオーディオ能力を有するパーソナルコンピュータを含む請求項１に記載の方法。
【請求項３５】
上記ターゲットデバイスは、ＷＡＰ（ワイヤレスアプリケーションプロトコル）サポートを含む請求項１に記載の方法。
【請求項３６】
上記メディアオブジェクトはデジタルイメージを含む請求項１に記載の方法。
【請求項３７】
上記デジタルオブジェクトはデジタルビデオを含む請求項１に記載の方法。
【請求項３８】
上記デジタルオブジェクトはデジタルオーディオを含む請求項１に記載の方法。
【請求項３９】
ターゲットデバイスへデジタルメディアを供給するためのオンラインシステムにおいて、
特定ターゲットデバイスの能力を決定するための能力モジュールと、
特定メディアオブジェクトのコピーを自動的に検索し、そして上記オブジェクトのコピーを上記ターゲットデバイスに供給するための変換モジュールであって、上記コピーを、上記ターゲットデバイスの能力に基づいて特定フォーマットへ自動的に変換するような変換モジュールと、
を備えたオンラインシステム。
【請求項４０】
変換されたメディアオブジェクトのコピーを記憶するためのキャッシュメモリを更に備えた請求項３９に記載のシステム。
【請求項４１】
上記システムは、先ず、上記キャッシュメモリから上記特定フォーマットの上記特定オブジェクトのコピーを検索することにより上記要求を満足しようと試みる請求項４０に記載のシステム。
【請求項４２】
検索されたメディアオブジェクトのコピーを記憶するためのキャッシュメモリを更に備えた請求項３９に記載のシステム。
【請求項４３】
上記システムは、先ず、リモートサーバーからコピーを検索する前に、上記キャッシュメモリから上記特定メディアオブジェクトのコピーを検索するように試みる請求項４２に記載のシステム。
【請求項４４】
上記システムにより記憶された各デジタルオブジェクトは、独特の均一リソースロケータ（ＵＲＬ）により識別される請求項３９に記載のシステム。

【請求項４５】
上記独特のＵＲＬは、上記デジタルオブジェクトの特性でエンコードされる請求項４４に記載のシステム。
【請求項４６】
上記独特のＵＲＬは、上記デジタルオブジェクトのカラー深度を含む請求項４４に記載のシステム。
【請求項４７】
上記独特のＵＲＬは、上記デジタルオブジェクトのイメージサイズを含む請求項４４に記載のシステム。
【請求項４８】
上記独特のＵＲＬは、上記デジタルオブジェクトの解像度を含む請求項４４に記載のシステム。
【請求項４９】
上記独特のＵＲＬは、上記デジタルオブジェクトのビットレートを含む請求項４４に記載のシステム。
【請求項５０】
上記システムは、デジタルオブジェクトの各々に対してＵＲＬを記憶し、そして上記能力モジュールは、上記ＵＲＬから上記ターゲットデバイスへ供給できる特定のデジタルオブジェクトを決定することができる請求項４４に記載のシステム。
【請求項５１】
上記ターゲットデバイスの能力は、スクリーン解像度を含む請求項３９に記載のシステム。
【請求項５２】
上記ターゲットデバイスの能力は、スクリーンサイズを含む請求項３９に記載のシステム。
【請求項５３】
上記ターゲットデバイスの能力は、カラーサポートを含む請求項３９に記載のシステム。
【請求項５４】
上記ターゲットデバイスの能力は、ビットレートを含む請求項３９に記載のシステム。
【請求項５５】
上記ターゲットデバイスの能力は、上記ターゲットデバイスがその要求を送信するために使用する現在使用可能な通信メディアを含む請求項３９に記載のシステム。
【請求項５６】
上記現在使用可能な通信メディアは、ワイヤレス通信を含む請求項５５に記載のシステム。
【請求項５７】
上記現在使用可能な通信メディアは、有線通信を含む請求項５５に記載のシステム。
【請求項５８】
上記能力モジュールは、上記ターゲットデバイスの能力をそのＨＴＴＰヘッダから決定するための機能を含む請求項３９に記載のシステム。
【請求項５９】
上記能力モジュールは、上記ターゲットデバイスの能力をそのＨＴＴＰユーザーエージェントヘッダから決定するための機能を含む請求項５８に記載のシステム。
【請求項６０】
上記能力モジュールは、上記ターゲットデバイスにその能力について問合せる機能を含む請求項３９に記載のシステム。
【請求項６１】
上記能力モジュールは、上記ターゲットデバイスの能力をそのデバイスクラスに基づいて決定するための知識ベースを含む請求項３９に記載のシステム。

【請求項６２】
上記デバイスの能力を上記知識ベースに追加できるようにするために確認されていないターゲットデバイスを記録するログ記録を更に備えた請求項６１に記載のシステム。
【請求項６３】
上記特定のフォーマットは、上記ターゲットデバイスにおいて上記特定メディアオブジェクトをレンダリングするための適当な解像度に基づいて選択される請求項３９に記載のシステム。
【請求項６４】
上記特定のフォーマットは、上記ターゲットデバイスにおいて上記特定メディアオブジェクトをレンダリングするための適当なカラースペースに基づいて選択される請求項３９に記載のシステム。
【請求項６５】
上記特定のフォーマットは、上記ターゲットデバイスにおいて上記特定メディアオブジェクトをレンダリングするための適当なイメージサイズに基づいて選択される請求項３９に記載のシステム。
【請求項６６】
上記特定のフォーマットは、上記ターゲットデバイスにおいて上記特定メディアオブジェクトをレンダリングするための適当なビットレートに基づいて選択される請求項３９に記載のシステム。
【請求項６７】
上記特定のフォーマットは、上記特定メディアオブジェクトのコピーを上記ターゲットデバイスに送信するのに使用できる通信帯域巾に基づいて選択される請求項３９に記載のシステム。
【請求項６８】
上記使用できる通信帯域巾は、上記ターゲットデバイスのデバイスクラスに少なくとも一部分基づいて決定される請求項６７に記載のシステム。
【請求項６９】
上記ターゲットデバイスは、表示能力を有するハンドヘルド計算デバイスを含む請求項３９に記載のシステム。
【請求項７０】
上記ターゲットデバイスは、デジタルオーディオ能力を有するハンドヘルド計算デバイスを含む請求項３９に記載のシステム。
【請求項７１】
上記ターゲットデバイスは、表示能力を有するセルラー電話デバイスを含む請求項３９に記載のシステム。
【請求項７２】
上記ターゲットデバイスは、デジタルオーディオ能力を有するセルラー電話デバイスを含む請求項３９に記載のシステム。
【請求項７３】
上記ターゲットデバイスは、表示能力を有するページャーデバイスを含む請求項３９に記載のシステム。
【請求項７４】
上記ターゲットデバイスは、表示能力を有するパーソナルコンピュータを含む請求項３９に記載のシステム。
【請求項７５】
上記ターゲットデバイスは、デジタルオーディオ能力を有するパーソナルコンピュータを含む請求項３９に記載のシステム。
【請求項７６】
上記ターゲットデバイスは、ＷＡＰ（ワイヤレスアプリケーションプロトコル）サポートを含む請求項３９に記載のシステム。

【請求項７７】
オンラインシステムにおいて、クライアントデバイスの能力を決定する方法が、
ターゲットデバイスからオリジナル要求を受け取り、上記ターゲットデバイスは、その能力に関する情報を含んでおらず、
上記ターゲットデバイスの能力を決定し、
上記ターゲットデバイスから受け取られた上記オリジナル要求を、上記ターゲットデバイスの能力に関する情報で補足し、そして
上記補足された要求を、上記オリジナル要求で指定された行先に転送する、
という段階を備えた方法。
【請求項７８】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにより提出された要求を検討することを含む請求項７７に記載の方法。
【請求項７９】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにより提出されたＨＴＴＰヘッダを検討することを含む請求項７７に記載の方法。
【請求項８０】
上記デバイスにより提出されたＨＴＴＰヘッダを検討することは、ＨＴＴＰユーザーエージェントヘッダを検討することを含む請求項７９に記載の方法。
【請求項８１】
ターゲットデバイスの能力を決定する上記段階は、上記デバイスにその能力について問合せることを含む請求項７７に記載の方法。
【請求項８２】
ターゲットデバイスの能力を決定する上記段階は、上記ターゲットデバイスのデバイスクラスに基づいて知識ベースから能力を決定することを含む請求項７７に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、情報処理に係り、より詳細には、デジタルメディア及びその関連情報へのオンラインアクセスを改善するための方法を与えるオンラインシステムに係る。
版権の通告
本特許文書の開示の一部分は、版権保護を受ける資料を含んでいる。版権所有者は、特許商標庁のファイルや記録に現われるように特許文書又は特許開示を誰かが複製複写することに異議を唱えないが、他の点では何であれ全ての版権が保存されるものとする。
コンピュータプログラムリストの添付資料
本出願には、コンピュータプログラムリストを含むコンピュータプログラムリスト添付資料Ａが添付されている。コンピュータプログラムリスト添付資料Ａの開示は、参考としてここに援用する。
【背景技術】
【０００２】
現在、消費者には、種々のデジタルイメージ、デジタルビデオ及びデジタルオーディオ製品が入手できる。デジタルメディアがいかに記録されるかに関わらず、その後のある時点で、情報が他のデバイスで入手でき、即ちデジタルデバイスの大きなネットワークで入手できて、その情報をスクリーンに表示したり、ハードコピーにプリントしたり、記憶したり、又は他の人々と共有したりすることができねばならない。今日、イメージ、ビデオ、オーディオ及び他の形式のメディア及びデジタルオブジェクトを編成及び表示する能力をもつサーバーコンピュータを伴う多数のウェブサイトがインターネット上に存在する。相補的に、このメディアを潜在的に見る（及び／又は聞く）のに充分なグラフィック能力をもつ多数の異なるクライアントデバイスが存在する。例えば、このようなクライアントデバイスは、ウェブブラウザソフトウェアを実行するデスクトップ又はラップトップコンピュータから、ハンドヘルドデバイス（例えば、パーム又はウインドウズＣＥのもとで動

作するパーソナルデジタルアシスタント）まであり、これらは全てＴＣＰ／ＩＰを経てインターネットへ接続され、そして各々情報を表示する能力を有している。

【０００３】

最近、ワイヤレスデータ能力及び表示能力をもつ新たなクラスのデバイスが導入されている。これらのデバイスは、通常、ＷＡＰ（ワイヤレスアプリケーションプロトコル）を経てインターネットに接続される。ＷＡＰは、ワイヤレスネットワークを経てデータを送信するためにモトローラ、エリクソン及びノキアを含むワイヤレス会社の共同企業体により開発されたＴＣＰ／ＩＰのような通信プロトコルである。ＷＡＰの説明については、例えば、１９９９年１０月、マンＳ．著「The Wireless Application Protocol」、ドクタードブズジャーナル、第５６－６６ページを参照されたい。

【０００４】

全体的に見て、ワイヤレス接続（例えば、９６００ボー）及び有線接続（例えば、５６ｋボー、ＤＳＬ及びケーブルモデム）の両方を経てインターネットに接続できるグラフィック装備のデバイスは、今日、おびただしい量が存在する。これらのデバイスは、デジタルイメージを含むグラフィックスを表示することができる。ハンドヘルド型の計算及びワイヤレス技術の最近の進歩は、製造者が、パーム型オーガナイザー、ワイヤレス電話及び両方向ページャーを含む巾広いデバイスアレーにウェブブラウザを埋設し又は含ませることができるようにした。これらのウェブイネーブル型クライアントは、その全てが、テキストを少なくとも基本的に表示できるが、あるものは、イメージ、ビデオ及びサウンドのような種々のマルチメディアコンテンツもサポートする。高速のワイヤレスインターネットアクセスは、大きな豊富なメディアフォーマットを再生できるクライアントデバイスの開発を推進する。

【０００５】

しかしながら、イメージ及び他のデジタルメディアを多数のこれらデバイスで表示する現在の解決策には基本的な問題が存在する。パーソナルコンピュータやラップトップコンピュータのような幾つかのデバイスは、デジタルイメージ、ストリーミングオーディオ及びストリーミングビデオを含む多数の異なる形式のメディアを受け取ってレンダリングすることができる。しかしながら、他のデバイスは、デジタルメディアをレンダリングする能力に甚だしい制約がある。更に、異なるデバイスが、異なるフォーマットや通信搬送プロトコルをしばしばサポートする。

【０００６】

例えば、パームハンドヘルドデバイスの例を取り上げる。ユーザが自分の家族の「真のカラー(true-color)」（例えば、３２ビット深度）の１０２４ｘ７６８デジタル写真をウェブに有すると仮定する。ユーザは、パームデバイスを使用してインターネットに接続する場合に、パームデバイスが４レベル又は１６レベルのグレースケールしかサポートできないので、この写真を表示することができない。イメージを表示できても、イメージをパームデバイスへダウンロードするための送信時間は、実際的でないものとなる。更に、イメージをダウンロードできても、パーム用のディスプレイは、ユーザに受け入れられるようにイメージをレンダリングするには物理的に小さ過ぎる。

【０００７】

この問題は、イメージデータのみに限定されず、他の形式のデジタルオブジェクトについてもそうである。多くのインターネットサイトは、デスクトップ及びラップトップコンピュータデバイス並びにウェブブラウザソフトウェアをもつユーザを主としてターゲットとする多色イメージや、ストリーミングビデオや、ストリーミングオーディオや、他のコンテンツを表示する。デスクトップ又はラップトップコンピュータは、著しい処理、記憶及び表示リソースを有し、大きなイメージ及びビデオファイルを多数のカラー及びフォーマットでレンダリングすることができる。他方、セルラー電話やパーソナルデジタルアシスタント（ＰＤＡ）のような小さなデバイスは、通常、カラーを表示したりメディアを特定のフォーマットで取り扱ったりするに必要な能力をもっていない。例えば、メッセージを表示するための小さなスクリーンを通常有するセルラー電話は、ＪＰＥＧイメージをレ

ンダリングするためのソフトウェア又は表示能力を有していない。現在、ＰＤＡ又はセルラー電話をもつユーザは、通常、多くのインターネットサイトで入手できる情報の多くにアクセスすることができない。

【０００８】

主としてセルラー電話やＰＤＡのユーザに焦点を合わせたあるコンテンツプロバイダーは、それらのユーザに適したメディアを表示するインターネットサイトをオファーする。これらのサイトは、これら形式のデバイスによりサポートされたフォーマットで低い解像度及び少数のカラーのイメージを含む。しかしながら、セルラー電話及びハンドヘルドデバイスのユーザに焦点を合わせたコンテンツプロバイダーにも、現在使用されている広範囲な多数のデバイスをサポートする問題がある。多数の異なる形式のセルラー電話及びハンドヘルドデバイスが使用され、その各々が異なる仕様及び能力を有している。これらのデバイスは、スクリーンサイズ、解像度及びカラー能力が異なる。従って、コンテンツプロバイダーは、たとえセルラー電話ユーザに焦点を合わせても、最も広範囲な多数のユーザにサービスするために最低能力デバイスに合わせたイメージ及び他のメディアオファーをしばしば形成しなければならない。

【０００９】

メディアをワイヤレスユーザに配送する際の別の特定の問題は、帯域巾が限定されていることである。多数のワイヤレスデバイスのイメージ表示及びオーディオ出力の制限に加えて、使用可能な帯域巾は、これらデバイスが豊富なメディアフォーマットにアクセスしてそれを使用する能力も制限する。従って、オーディオ及びビデオ能力に優れたデバイスでも、帯域巾を考えると、これらのデバイスへ供給できるメディアの形式が著しく制約される。ユーザのワイヤレスデバイスが高解像度イメージを表示できても、ユーザは、使用帯域巾に限度が与えられると、イメージをダウンロードするのに要する時間のために、低解像度のイメージを要求することになる。

【００１０】

多くのワイヤレスデバイスに使用できる帯域巾及び表示リソースに限度があるために生じるこれらの問題に加えて、別の問題は、今日のインターネットサービスが特定の接続されたデバイスの能力を理解しようとしないことである。更に、このようなインターネットサービスが特定のクライアントデバイスの若干の能力を理解したとしても、今日のサービスは、その情報に作用してその情報を、そのようなクライアントデバイスにとって意味のあるフォーマットのみで送信するように設計されていない。益々多くのクラスのネットワーク接続デバイスが市場に出回るにつれて、各デバイスがそれ自身の特定の特性を有するので、この問題が増大することが予想される。

【発明の開示】

【発明が解決しようとする課題】

【００１１】

そこで、オンザフライのメディア再フォーマッティングを、進歩したクライアント検出と結合して、プロバイダーのメディアを各接続されたクライアントデバイスに適切に且つ考えられる最良の形態で配送できるようにする解決策が要望される。

【００１２】

用語

以下の説明の理解を助けるために、説明上、以下の定義を行なうが、これに限定されるものではない。

Ａｐａｃｈｅ：これは、「アパッチグループ」と称されるボランティアプログラマーのグループにより開発されたパブリックドメインウェブサーバーである。アパッチサーバーの初期バージョンは、１９９５年に、National Center for Supercomputing Applications, University of Illinois, Urbana-Champaignにおいて開発されたｈｔｔｐｄウェブサーバーに基づいて開発されたものである。アパッチウェブサーバーの付加的な情報及びこのウェブサーバーに対するソースコードのコピーは、現在、インターネットのｈｔｔｐ：／／ｗｗｗ．ａｐａｃｈｅ．ｏｒｇ．において入手できる。

【００１３】
ＣＣ／ＰＰ：これは、Composite Capabilities / Preference Profilesの省略形であり、ワールドワイドウェブコンソーシアム（Ｗ３Ｃ）により開発されつつある提案規格である。ＣＣ／ＰＰプロフィールは、デバイスの能力及びユーザの好みの記述で、そのデバイスに与えられるコンテンツの適応をガイドするのに使用できる。現在提案されている仕様は、ワールドワイドウェブコンソーシアム（Ｗ３Ｃ）から入手できる１９９９年７月２７日、Ｗ３Ｃノート、「Composite Capabilities / Preference Profiles (CC/PP): A user side framework for content negotiation」により説明されている。提案された規格のコピーは、インターネットのｈｔｔｐ：／／ｗｗｗ．ｗ３．ｏｒｇ／ＴＲ／ＮＯＴＥ－ＣＣＰＰ／．において現在見ることができる。

【００１４】
ＨＴＭＬ：これは、ハイパーテキストマークアップ言語を意味する。各ＨＴＭＬ文書は、ウェブブラウザによって正しく解釈されるために、ある標準的なＨＴＭＬタグを必要とする。各文書は、ヘッド及びボディーテキストで構成される。ヘッドは、タイトルを含み、そしてボディーは、パラグラフ、リスト及び他のエレメントで構成された実際のテキストを含む。ブラウザは、ＨＴＭＬ及びＳＧＭＬ仕様に基づいてプログラムされるので、特定の情報を予想する。ＨＴＭＬ文書の更なる説明は、技術及び取引文献において得られ、例えば、１９９５年６月１３日、ＰＣマガジン、レイ・ダンカン著、「Power Programming: An HTML Primer」を参照されたい。

【００１５】
ＨＴＴＰ：これは、ハイパーテキスト転送プロトコルを意味するもので、インターネットにおいてワールドワイドウェブにより使用される基礎的な通信プロトコルである。ＨＴＴＰは、メッセージをいかにフォーマットして送信するか、ウェブサーバー及びブラウザが種々のコマンドに応答してどんなアクションを取るべきかを定義する。例えば、ユーザが自分のブラウザにおいてＵＲＬに入ると、これが実際にＨＴＴＰコマンドをウェブサーバーに送信し、要求されたウェブページをフェッチして送信するように命令する。ＨＴＴＰの更なる説明は、ＲＦＣ２６１６：Hypertext Transfer Protocol HTTP/1.1において得られる。ＲＦＣ２６１６は、ワールドワイドウェブコンソーシアム（Ｗ３Ｃ）から得られ、そしてインターネットのｈｔｔｐ：／／ｗｗｗ．ｗ３．ｏｒｇ／Ｐｒｏｔｏｃｏｌｓ／．において現在得られる。ＨＴＴＰの更なる説明は、技術及び取引文献において得られ、例えば、１９９６年１０月、ＢＹＴＥ、ウイリアム・スターリングズ著、「The Backbone of the Web」を参照されたい。

【００１６】
ＪＰＥＧ：ジョイント・フォトグラフィック・エキスパート・グループ即ちＪＰＥＧフォーマットでフルサイズのデジタルイメージが維持される。例えば、１９９６年、Ｍ＆Ｔブックス、ネルソン氏等著の「The Data Compression Book, Second Edition, Chapter 11: Lossy Graphics Compression」（特に、第３２６－３３０ページ）を参照されたい。又、例えば、１９９５年７月及び１９９５年８月、ドクタードブズジャーナル、「JPEG-like Image Compression」（パート１及び２）（カリフォルニア州サンマテオのドクタードブズジャーナルからドクタードブズ／ＣＤリリース６としてＣＤＲＯＭで入手できる）を参照されたい。

【００１７】
ＳＭＴＰ：これは、シンプル・メール・トランスファー・プロトコルを意味し、サーバー間にｅ－メールメッセージを送信するためのプロトコルである。インターネットを経てメールを送信するほとんどのｅ－メールシステムは、ＳＭＴＰを使用して、あるサーバーから別のサーバーへメッセージを送信し、従って、メッセージは、ｅ－メールクライアントがＰＯＰ又はＩＭＡＰを使用して検索することができる。更に、ＳＭＴＰは、一般に、メールクライアントからメールサーバーへメッセージを送信するのに使用される。

【００１８】
ＴＣＰ：これは、トランスミッション・コントロール・プロトコルを意味する。ＴＣＰ

は、ＴＣＰ／ＩＰネットワークにおけるメインプロトコルの１つである。ＩＰプロトコルはパケットのみを取り扱うが、ＴＣＰは、２つのホストが接続を確立しそしてデータの流れを交換できるようにする。ＴＣＰは、データの配送を保証すると共に、パケットが送信されたのと同じ順序で配送されるよう保証するものである。ＴＣＰの紹介については、例えば、ＲＦＣ７９３を参照されたい。

【００１９】

ＴＣＰ／ＩＰ：これは、トランスミッション・コントロール・プロトコル／インターネットプロトコル、即ちインターネットのホストに接続するのに使用される通信プロトコルの組を意味する。ＴＣＰ／ＩＰは、多数のプロトコルを使用し、２つの主たるものがＴＣＰ及びＩＰである。ＴＣＰ／ＩＰは、ＵＮＩＸ（登録商標。以下同様。）オペレーティングシステムに組み込まれ、インターネットにより使用されて、ネットワークを経てデータを送信するための事実上の規格とされる。ＴＣＰ／ＩＰの紹介については、例えば、ＲＦＣ１１８０：A TCP/IP Tutorialを参照されたい。ＲＦＣ１１８０のコピーは、ｆｔｐ：／／ｆｔｐ．ｉｓｉ．ｅｄｕ／ｉｎ－ｎｏｔｅｓ／ｒｆｃ１１８０．ｔｘｔ．において現在入手できる。

【００２０】

ＵＡＰｒｏｆ：ＵＡＰｒｏｆ又はＷＡＧ　ＵＡＰｒｏｆは、セルラー電話のようなデバイスがそれらの能力をサーバーへいかに記述すべきかに関して提案された「ワイヤレス・アクセス・グループ・ユーザ・エージェント・プロフィール仕様」を指す。現在提案される仕様は、ＷＡＰフォーラムから入手できる「ＷＡＧ　ＵＡＰｒｏｆ」（ワイヤレス・アプリケーション・グループ・ユーザ・エージェント・プロフィール仕様）、ワイヤレス・アプリケーション・プロトコル・フォーラム・リミテッド、プロポーズドバージョン３０－Ｍａｙ－２００１として説明されている。この仕様のコピーは、インターネットのｈｔｔｐ：／／ｗｗｗ１．ｗａｐｆｏｒｕｍ．ｏｒｇ／ｔｅｃｈ／ｄｏｃｕｍｅｎｔｓ／ＷＡＰ－２４８－ＵＡＰｒｏｆ－２００１０５３０－ｐ．ｐｄｆにおいて現在見ることができる。

【００２１】

ＵＲＬ：ユニフォーム・リソース・ロケータの省略形で、ワールドワイドウェブにおける文書及び他のリソースのグローバルアドレスである。アドレスの第１部分は、どんなプロトコルを使用すべきか指示し、そして第２部分は、リソースが位置するＩＰアドレス又はドメイン名を指定する。

【００２２】

ＷＡＰ：これは、ワイヤレス・アプリケーション・プロトコルを意味するもので、ワイヤレスネットワークを経てデータを送信するためにモトローラ、エリクソン及びノキアを含むワイヤレス会社の共同企業体により開発されたＴＣＰ／ＩＰのような通信プロトコルである。ＷＡＰの説明については、例えば、１９９９年１０月、マンＳ．著「The Wireless Application Protocol」、ドクタードブズジャーナル、第５６－６６ページを参照されたい。より詳細には、ＷＡＰは、既存のインターネットプロトコルの種々の等効物を含む。例えば、ＷＭＬは、ＨＴＭＬプロトコルのＷＡＰバージョンである。他の例は、ＨＴＭＬブラウザのＷＡＰ等効物であるＷＡＰブラウザと、ＨＴＴＰサーバーのＷＡＰ等効物であるＷＡＰゲートウェイ（サーバー側）とを含む。ＷＡＰゲートウェイでは、ＨＴＴＰがＷＡＰへ／ＷＡＰから変換される。

【００２３】

ＸＭＬ：エクステンシブル・マークアップ言語の短縮形で、Ｗ３Ｃにより開発された仕様である。ＸＭＬは、ウェブ文書に対して特に設計されたＳＧＭＬのペア・ダウンバージョンである。これは、設計者がそれら自身のカスタマイズされたタグを形成できるようにし、アプリケーション間及び組織間でデータの定義、送信、有効確認及び解釈を行なえるようにする。ＸＭＬの更なる説明については、例えば、ワールドワイドウェブコンソーシアム（ｗｗｗ．ｗ３．ｏｒｇ）から入手できるエクステンシブル・マークアップ言語（ＸＭＬ）１．０仕様を参照されたい。又、この仕様は、インターネットのｈｔｔｐ：／／ｗ

ｗｗ．ｗ３．ｏｒｇ／ＴＲ／ＲＥＣ－ｘｍｌにおいても現在入手できる。

【課題を解決するための手段】

【００２４】

本発明は、コンテンツプロバイダーが最も巾広いクライアントアレーに効果的にサービスできるようにする改良された方法を組み込んだオンラインシステムを提供する。これは、オンザフライのメディア再フォーマッティングを、進歩型クライアント検出能力と結合して、プロバイダーのメディアコンテンツを各接続されたクライアントデバイスに適切に且つ考えられる最良の形態で配送できるようにする。

【００２５】

本発明のオンラインメディア配送システム（商業的にはｅＳｗｉｔｃｈ（登録商標）ブランドのメディア配送システムで実施される）は、メディア文書又はオブジェクトについてクライアントデバイスから要求を受け取り、その要求を発するデバイスのメディア出力能力を決定し、そして要求を発しているクライアントデバイスに適したメディアオブジェクトの変換されたバージョンにサービスする。このシステムは、クライアント能力モジュール（ＣＣＭ）と、メディア変換モジュール（ＭＴＭ）とを備えている。これら２つのモジュールは、ＨＴＴＰ要求からクライアントを識別し、そのメディア出力能力を決定し、そしてそれら能力に基づいてソースメディアを再フォーマットするように協働する。又、このシステムは、種々のクライアントデバイスの能力に関する情報を含むデータ記憶デバイスと、変換されたメディアコンテンツを記憶するための（任意の）フロントサイドキャッシュと、コンテンツのオリジナル項目をローカル記憶するためのバックサイドキャッシュとを備えている。

【００２６】

このシステムは、多数の異なるクライアントデバイスのメディアコンテンツ、即ちハードウェア及びソフトウェア能力の異なるターゲットデバイスへのアクセスを与える。このシステムは、接続能力をもつ実際上いかなるデバイスにも適当なメディアコンテンツを配送できるようにする。ターゲットデバイスは、ワイヤレスデバイス（例えば、セルラー電話、ハンドヘルドパーソナルデジタルアシスタント（ＰＤＡ）及びページャー）と、有線デバイス（例えば、デスクトップコンピュータ、ラップトップコンピュータ及びビデオフォン）の両方を含んでもよい。

【００２７】

特定のデバイスに適したメディアコンテンツを供給するシステムの改良された方法は、次のように要約することができる。最初に、インターネットウェブサイトにおける特定の全フォーマットマルチメディアオブジェクトのＵＲＬが、本発明のシステムによりサービスされるように変更される。これは、メディア配送システムがインストールされたサーバーを指し且つメディアオブジェクトへの経路を含んでいるＵＲＬを、これら全フォーマットマルチメディアオブジェクトのＵＲＬに置き換えるように、ウェブサイトのＨＴＭＬページを変更することにより行われる。このシステムは、これらオリジナルオブジェクトに対するＨＴＴＰプロキシーとして働き、オリジナルコンテンツの要求を遮断し、そして要求を発しているクライアントに適用できるコンテンツの変換されたバージョンにサービスする。

【００２８】

クライアントデバイスが、特定のマルチメディア文書を要求するための変更されたＵＲＬを呼び出すユーザ入力（例えば、クリック）を受け取ると、メディア出力能力がデバイスによりシステムに通信されるか、又はシステムのクライアント能力モジュールにより推測される。デバイスがその能力を通信する場合には、初期動作中又は各相互作用中にこれを行う。或いは又、デバイスの能力が、デバイスの以前の知識に基づいてシステムのデータ記憶デバイスに予め記憶される。通信されるか又はシステムにより推測された情報に基づいて、ターゲットデバイスの能力を記述する情報記録が生成される。この情報は、メディアコンテンツのオリジナル項目をターゲットデバイスに適したフォーマットに変換するためにどんな変換（もしあれば）が要求されるかをシステムに指示する。

【００２９】
デバイスにより要求された適当なメディアフォーマットが決定された後に、クライアント能力モジュールは、フロントサイドキャッシュを（任意に）チェックして、この特定のターゲットデバイスに適したフォーマットに変換されたオブジェクトのバージョンをそのキャッシュが既に記憶しているかどうか調べる。そのオブジェクト（ターゲットデバイスに対して変換された）がフロントサイドキャッシュに既に存在する場合には、クライアント能力モジュールは、そのオブジェクトをクライアントデバイスへ単に返送すればよい。しかしながら、適当に変換されたオブジェクトがキャッシュにない場合には、クライアント能力モジュールがオブジェクト識別子及びクライアントデバイスパラメータをメディア変換モジュールへ通すように動作を進める。

【００３０】
メディア変換モジュールは、クライアント能力モジュールから特定のフォーマットでメディアコンテンツの特定項目の要求を受け取る。メディア変換モジュールは、オリジナルメディアオブジェクトを得、そのオブジェクトを、オリジナルフォーマットからターゲットデバイスの希望のフォーマット（指定のターゲットデバイス能力に基づく）へと変換し、そしてその変換されたオブジェクトをクライアントデバイスへ返送する。メディア変換モジュールは、高い効率を与えるための最適なものとしてバックサイドキャッシュを使用する。メディアオブジェクトは、頻繁に又は最近要求された項目を各要求に応答して検索しなければならないのを回避するために、バックサイドキャッシュに保持される。このバックサイドキャッシュの使用は、ネットワークを経てのコールの数を減少し、メディアオブジェクトの変換及びクライアントデバイスへの返送を促進する。

【発明を実施するための最良の形態】

【００３１】
以下、本発明の現在好ましいと思われる実施形態、即ちデスクトップオペレーティングシステム、例えば、ＩＢＭ対応のＰＣで実行されるＭｉｃｒｏｓｏｆｔ（登録商標）Ｗｉｎｄｏｗｓ（登録商標。以下同様。）オペレーティングシステムのもとで実行されるインターネット接続環境において動作するデスクトップアプリケーションで具現化されるものに焦点を合わせて説明する。しかしながら、本発明は、１つの特定のアプリケーション又は特定の環境に限定されない。むしろ、本発明のシステム及び方法は、Ｍａｃｉｎｔｏｓｈ、Ｌｉｎｕｘ、ＢｅＯＳ、Ｓｏｌａｒｉｓ、ＵＮＩＸ、ＮｅｘｔＳｔｅｐ、ＦｒｅｅＢＳＤ等を含む種々の異なるプラットホームにおいて効果的に実施できることが当業者に明らかであろう。それ故、以下の実施形態の説明は、本発明を例示するものに過ぎず、本発明をこれに限定するものではない。

【００３２】
Ｉ．コンピュータベースの実施
Ａ．基本的システムハードウェア（例えばデスクトップ及びサーバーコンピュータ用）
本発明は、ＩＢＭ対応のパーソナルコンピュータ（ＰＣ）又はサーバーコンピュータのような従来の又は汎用のコンピュータシステムにおいて実施することができる。図１は、ＩＢＭ対応システム１００の非常に一般的なブロック図である。図示されたように、このシステム１００は、中央処理ユニット（ＣＰＵ）即ちプロセッサ（１つ又は複数）１０１を備え、これは、ランダムアクセスメモリ（ＲＡＭ）１０２と、リードオンリメモリ（ＲＯＭ）１０３と、キーボード１０６と、プリンタ１０７と、ポインティングデバイス１０８と、ディスプレイデバイス１０５に接続されたディスプレイ又はビデオアダプタ１０４と、取り外し可能な（大量）記憶デバイス１１５（例えば、フレキシブルディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、ＣＤ－ＲＷ、ＤＶＤ等）と、固定（大量）記憶デバイス１１６（例えば、ハードディスク）と、通信（ＣＯＭＭ）ポート又はインターフェイス１１０と、モデム１１２と、ネットワークインターフェイスカード（ＮＩＣ）又はコントローラ１１１（例えば、イーサネット（登録商標。以下同様。））とに接続される。別々に示されていないが、従来そうであるように、システム１００にはリアルタイムシステムクロックが含まれている。

【００３３】
ＣＰＵ１０１は、Ｉｎｔｅｌ　Ｐｅｎｔｉｕｍ（登録商標。以下同様。）マイクロプロセッサファミリーのプロセッサで構成される。しかしながら、他の適当なプロセッサを使用して、本発明を実施することもできる。ＣＰＵ１０１は、両方向性システムバス（必要な入力／出力（Ｉ／Ｏ）コントローラ回路及び他の「グリュー(glue)ロジック」を含む）を経てシステムの他の要素と通信する。システムメモリをアドレスするためのアドレスラインを含むバスは、種々の要素間にデータ転送を与える。Ｐｅｎｔｉｕｍクラスのマイクロプロセッサ及びそれらの命令セット、バスアーキテクチャー、及び制御ラインの説明は、カリフォルニア州サンタクララのインテル・コーポレーションから入手できる。ランダムアクセスメモリ１０２は、ＣＰＵ１０１のためのワーキングメモリとして働く。典型的な構成では、６４メガバイト以上のＲＡＭが使用される。本発明の範囲から逸脱せずに、それより大きな又は小さなメモリが使用されてもよい。リードオンリメモリ（ＲＯＭ）１０３は、基本的な入力／出力システムコード（ＢＩＯＳ）、即ちアプリケーションプログラム及びオペレーティングシステムがハードウェアと相互作用するために使用できるＲＯＭにおける１組の低レベルルーチンを含み、これは、キーボードからのキャラクタの読み取り、プリンタへのキャラクタの出力、等々を含む。

【００３４】
大量記憶デバイス１１５、１１６は、磁気、光学、又は磁気－光学記憶システムや、フラッシュメモリや、他の使用可能な大量記憶技術のような固定及び取り外し可能なメディアにおいて永続的な記憶を与える。大量記憶デバイスは、ネットワークに共有されてもよいし、又は専用の大量記憶デバイスであってもよい。図１に示すように、固定の記憶デバイス１１６は、コンピュータシステムのオペレーションを指令するためのプログラム及びデータのボディーを記憶し、これは、オペレーティングシステム、ユーザアプリケーションプログラム、ドライバー及び他のサポートファイル、並びに全ての種類の他のデータファイルを含む。通常、固定記憶デバイス１１６は、システムのメインハードディスクとして働く。

【００３５】
基本的な動作において、プログラムロジック（以下に述べる本発明の方法を実施するものを含む）が、取り外し可能な記憶デバイス１１５又は固定記憶デバイス１１６からメイン（ＲＡＭ）メモリ１０２へロードされ、ＣＰＵ１０１により実行される。プログラムロジックのオペレーション中に、システム１００は、キーボード１０６及びポインティングデバイス１０８からユーザ入力を受け入れると共に、音声認識システム（図示せず）からスピーチベースの入力を受け入れる。キーボード１０６は、アプリケーションプログラムの選択、キーボードベースの入力又はデータのエントリー、スクリーン又はディスプレイデバイス１０５に表示された個々のデータオブジェクトの選択及び操作を行うことができる。同様に、マウス、トラックボール、ペンデバイス等のポインティングデバイス１０８は、ディスプレイスクリーン上でオブジェクトの選択及び操作を行うことができる。このようにして、これらの入力デバイスは、システムで実行されるプロセスに対する手動ユーザ入力をサポートする。

【００３６】
コンピュータシステム１００は、テキスト及び／又はグラフィックイメージ並びに他のデータをディスプレイデバイス１０５に表示する。ディスプレイ１０５とシステムバスとの間の介在されるビデオアダプタ１０４は、ディスプレイデバイス１０５を駆動する。ＣＰＵ１０１へアクセスできるビデオメモリを含むビデオアダプタ１０４は、ビデオメモリに記憶されたピクセルデータを、陰極線管（ＣＲＴ）ラスタ又は液晶ディスプレイ（ＬＣＤ）モニタに使用するのに適したタスタ信号へ変換する回路を備えている。表示された情報又はシステム１００内の他の情報のハードコピーをプリンタ１０７又は他の出力デバイスから得ることができる。プリンタ１０７は、システムの出力のハードコピーイメージを生成するために、例えば、ＨＰ　ＬａｓｅｒＪｅｔ（登録商標）プリンタ（カリフォルニア州パロアルトのヒューレットパッカードから入手できる）を含んでもよい。

【００３７】
システムそれ自身は、ネットワーク（例えば、イーサネットネットワーク、ブルーツースワイヤレスネットワーク等）に接続されたネットワークインターフェイスカード（ＮＩＣ）１１１、及び／又はモデム１１２（例えば、５６Ｋボー、ＩＳＤＮ、ＤＳＬ又はケーブルモデム）を経て他のデバイス（例えば、他のコンピュータ）と通信し、モデムは、例えば、カリフォルニア州サンタクララの３Ｃｏｍから入手できる。又、システム１００は、ＲＳ－２３２シリアルポート、ユニバーサルシリアルバス（ＵＳＢ）インターフェイス等を含む通信（ＣＯＭＭ）インターフェス１１０を経て、ローカルの時々接続されるデバイス（例えば、シリアルケーブルリンクデバイス）と通信することもできる。インターフェイス１１０に通常ローカル接続されるデバイスは、ラップトップコンピュータ、ハンドヘルドオーガナイザー、デジタルカメラ等を含む。

【００３８】
ＩＢＭ対応のパーソナルコンピュータ及びサーバーコンピュータは、種々の売主から入手できる。代表的な売主は、テキサス州ラウンドロックのデル・コンピュータ、テキサス州ハウストンのコンパック・コンピュータ、及びニューヨーク州アルモンクのＩＢＭを含む。他の適当なコンピュータは、カリフォルニア州キュペーチノのアップル・コンピュータから入手できるアップル対応コンピュータ（例えば、マッキントッシュ）や、カリフォルニア州マウンテンビューのサン・マイクロシステムズから入手できるサン・ソラリスワークステーションを含む。

【００３９】
Ｂ．基本的システムソフトウェア
図２に示すように、コンピュータシステム１００のオペレーションを指令するためのコンピュータソフトウェアシステム２００が設けられる。システムメモリ（ＲＡＭ）１０２及び固定記憶デバイス（例えば、ハードディスク）１１６に記憶されたこのソフトウェアシステム２００は、カーナル即ちオペレーティングシステム（ＯＳ）２１０を含む。このＯＳ２１０は、プロセスの実行、メモリの割り当て、ファイル入力及び出力（Ｉ／Ｏ）、及びデバイスＩ／Ｏを管理することを含むコンピュータオペレーションの低レベル観点を管理する。クライアントアプリケーションソフトウェア又は「プログラム」２０１（例えば、２０１ａ、２０１ｂ、２０１ｃ、２０１ｄ）のような１つ以上のアプリケーションプログラムが「ロード」され（即ち固定記憶デバイス１１６からメモリ１０２へ転送され）、システム１００により実行される。

【００４０】
システム２００は、ユーザコマンド及びデータをグラフィック（例えば、「ポイント・アンド・クリック」）形態で受け取るためのグラフィックユーザインターフェイス（ＧＵＩ）２１５を備えている。これらの入力は、次いで、オペレーティングシステム２１０及び／又はクライアントアプリケーションモジュール２０１からの命令に基づいてシステム１００により作用されてもよい。又、ＧＵＩは、ＯＳ２１０及びアプリケーション２０１からのオペレーションの結果を表示するようにも働き、その際に、ユーザは、付加的な入力を供給してもよいし、セッションを終了してもよい。通常、ＯＳ２１０は、特に周辺デバイスとインターフェイスするときには、デバイスドライバー２２０（例えば、「Ｗｉｎｓｏｃｋ」ドライバー、即ちＴＣＰ／ＩＰスタックのウインドウズ実施）及びシステムＢＩＯＳマイクロコード２３０（即ちＲＯＭベースのマイクロコード）に関連して動作する。ＯＳ２１０は、従来のオペレーティングシステム、例えば、Ｍｉｃｒｏｓｏｆｔ（登録商標）Ｗｉｎｄｏｗｓ９ｘ、Ｍｉｃｒｏｓｏｆｔ（登録商標）ＷｉｎｄｏｗｓＮＴ、Ｍｉｃｒｏｓｏｆｔ（登録商標）Ｗｉｎｄｏｗｓ２０００又はＭｉｃｒｏｓｏｆｔ（登録商標）ＷｉｎｄｏｗｓＸＰによって形成することができ、これらは全てワシントン州レドモンドのマイクロソフト・コーポレーションから入手できる。或いは又、ＯＳ２１０は、上述したオペレーティングシステムのような別のオペレーティングシステムであってもよい。

【００４１】
上述したコンピュータハードウェア及びソフトウェアは、本発明を実施するために使用

できる基本的基礎的なデスクトップ及びサーバーコンピュータ要素を説明する目的で示されている。検討の目的で、以下の説明は、１つ以上の「クライアント」（例えば、メディア表示デバイス）と通信する「サーバー」（例えば、ウェブサーバー）が存在すると仮定した例を示す。しかしながら、本発明は、特定の環境又はデバイス構成に限定されるものではない。特に、クライアント／サーバーの区別は本発明にとって必要ないが、説明上の枠組みを与えるために使用される。むしろ、本発明は、以下に詳細に述べる本発明の方法をサポートすることのできるいかなる形式のシステムアーキテクチャー又は処理環境において実施されてもよい。

【００４２】

ＩＩ．種々のデバイスの能力に対して仕立てられたメディアのオンラインレンダリング

Ａ．序文

今日、大量の種々の形式のコンテンツが多数のインターネットサイトにおいて入手できる。同時に、メディアをユーザに表示（レンダリング）することのできる広範囲なターゲットデバイスが存在する。これらのデバイスは、パーソナルコンピュータ、ラップトップコンピュータ、パーソナルデジタルアシスタンス（ＰＤＡ）、両方向ページャー及びセルラー電話を含む。しかしながら、これらのデバイスへメディアコンテンツを配送するには、多数の問題が存在する。能力が著しく異なる種々のターゲットデバイスが使用されると共に、異なる形式のイメージ及び他のメディアコンテンツがインターネットにおいて入手できるので、コンテンツプロバイダーは、現在、特定のターゲットデバイスのユーザに満足なやり方で表示（又はレンダリング）するのに適した仕方でメディアコンテンツを配送するという問題を有している。

【００４３】

これらの問題の１つの解決策は、インターネットサイトが多数の異なるフォーマットで情報を表示することである。しかしながら、この解決策は、インターネットコンテンツプロバイダーが、市場における種々の形式のデバイス及びそれらの広範囲な能力に対処するために同じメディアオブジェクトの非常に多数のコピーを表示することを必要とする。この解決策の欠点の中には、コンテンツプロバイダーが、サポートされるべきデバイスの数に基づいて、同じメディアの多数の予めレンダリングされたバージョンを種々のフォーマットで生成し、記憶し、そして管理することを必要とすることも挙げられる。

【００４４】

本発明は、コンテンツプロバイダーが、コンテンツをあるフォームで開発し、そしてそのコンテンツを要求するクライアントデバイスの能力に基づいてそのコンテンツを多数のフォームで配送するのを許す。本発明は、デバイスのメディア出力能力に適したメディアコンテンツをクライアントデバイスに供給するためのオンラインシステム及び方法を包含する。このシステムは、接続されたクライアントデバイスの能力を決定するためのクライアント能力モジュール（ＣＣＭ）と、適当なメディアをオンザフライでレンダリングしそしてそれを適当なフォーマットでクライアントデバイスに配送するメディア変換（又はトランスコード化）モジュール（ＭＴＭ）とを備えている。

【００４５】

本発明の動作は、特定のクライアントデバイスに対してデジタルイメージをレンダリングする以下の例により説明することができる。この例では、インターネットサイトにおけるメディアコンテンツのオリジナル項目が２４ビットカラーＪＰＥＧイメージであり、そしてこのイメージを要求するクライアントデバイスは、１６レベルのグレースケールをサポートするパームＰＤＡである。このクライアントデバイスは、インターネットサイトにワイヤレスで接続され、そしてこのＪＰＥＧイメージに対してＵＲＬを呼び出す。本発明を使用するインターネットコンテンツプロバイダーは、本発明のクライアント能力モジュールがインストールされたマシンを参照するためにこのイメージに対するユニフォーム・リソース・ロケータ（ＵＲＬ）を予め改訂している。その結果、このＵＲＬ要求は、ＣＣＭへルーティングされ、これは、要求されたイメージ及びクライアントデバイスをこの要求から識別する。ＣＣＭは、サーバーに対するクライアント要求を検査してクライアント

デバイスに関する情報を得、そしてこの情報を、そのデータ記憶デバイスに記憶された既知のデバイス特性及び能力と比較することにより、インテリジェントな形態でクライアントデバイスの能力を決定する。この場合に、ＣＣＭは、このデバイスを特定形式のパームＰＤＡとして確認し、そしてシステムのデータベースにおいてデバイスの能力をルックアップする。この情報に基づき、ＣＣＭは、ＪＰＥＧイメージを１６レベルのグレースケールフォーマットでこのパームデバイスへ供給しなければならないと決定する。

【００４６】

デバイスにより要求された適当なメディアフォーマットが決定された後に、ＣＣＭは、フロントサイドキャッシュを（任意に）チェックして、そのキャッシュがイメージのバージョンを要求されたフォーマットで既に記憶しているかどうか調べる。適当な変換されたイメージがキャッシュにない場合には、ＣＣＭは、この形式のパームＰＤＡデバイスへレンダリングするのに適した１６レベルのグレースケールフォーマットでメディア変換モジュールからイメージを要求する。ＭＴＭは、適当なイメージを得て、それを適当なフォーマットに変換し、そしてそれをクライアントデバイスへサービスする。システムは、特定のターゲットデバイスにより使用するのに適したフォーマットにイメージを（そのオリジナルフォーマットから）最適に変換するのを許すインテリジェンスを含む。全体的な変換又は変更プロセスは、システムに要望される性能及び拡張性基準を保つように実行される。

【００４７】

Ｂ．メディア配送システムの概要

１．基本的アーキテクチャー

図３は、本発明を実施するのに適したオンライン環境３００を示す。図示されたように、この環境３００は、インターネット（３１０で示す）を経て１つ以上のクライアントデバイス３０１に接続されたオンラインメディア配送システム３２０と、少なくとも１つのインターネットサイト（サーバー）３３０とを備えている。これら要素の各々を以下に詳細に説明する。

【００４８】

クライアントデバイス３０１は、インターネットを経て接続されてオンラインコンテンツにアクセスすることのできる種々のターゲットデバイス（又は「クライアント」）の１つを表わす。例えば、クライアントデバイスは、ワイヤレスデバイス（例えば、セルラー電話、ハンドヘルドＰＤＡ（パーソナルデータアシスタント）、及びページャー）と、有線デバイス（例えば、デスクトップコンピュータ、ラップトップコンピュータ及びビデオフォン）の両方を含むことができる。単一のクライアントデバイスが図示されているが、この環境３００は、通常、このようなデバイスが多数接続された状態で動作する。

【００４９】

インターネットサーバー３３０は、メディアコンテンツの項目（例えば、オーディオ、ビデオ、文書、ブロブ(blob)オブジェクト又は他の当該項目）が記憶されたウェブサーバーを表わす。オペレーション中に、インターネットサーバー３３０は、通常、広範囲なクライアントデバイスに使用できるようにされるべき多数の種々のメディアコンテンツ項目を記憶する。インターネットサーバー３３０とオンラインメディア配送システム３２０との間の実際の接続は、インターネットを経て行われてもよいし、或いは任意であるが、図中破線の接続線で示されたように非インターネット（例えば、ＷＡＮ）接続を経て行われてもよい。いずれの場合にも、インターネットサーバー３３０は、オンラインメディア配送システム３２０と同じ場所に収容されてもよいし、或いは必要に応じて離れた場所に収容されてもよい。

【００５０】

オンラインメディア配送システム３２０は、クライアントデバイスの能力を検出するように機能し、そしてその決定に基づいて、メディアコンテンツを変換して、クライアントデバイス３０１に適したフォーマットでそこに配送する。図示されたように、このメディア配送システム３２０は、クライアント能力モジュール（ＣＣＭ）３２２と、メディア変

換モジュール（ＭＴＭ）３２５と、デバイス能力データ記憶デバイス３２４とを備えている。図示されたように、クライアント能力モジュール３２２は、フロントサイドキャッシュ３２１及びＣＣＭログ３２３と直接通信し、メディア変換モジュール３２５も同様にバックサイドキャッシュ３２７及びＭＴＭログ３２６と直接通信する。

【００５１】

２．基本的オペレーション

基本的なシステムオペレーション中に、インターネットサーバー３３０におけるメディアコンテンツ（例えば、ウェブページ）を含む及び／又は参照する項目は、そのような項目を要求しているクライアントをシステム３２０へと導くＵＲＬでエンコードされる。又、インターネットサーバー３３０は、デジタルイメージ、ビデオ、オーディオ、文書、「ブロブ」オブジェクト等を含む任意の形式のコンテンツであるメディアコンテンツのオリジナル項目を含んでもよい。或いは又、メディアコンテンツのオリジナル項目は、システム３２０にローカルに記憶されてもよいし、或いはシステム３２０に接続された別のローカル又はリモートサーバーに記憶されてもよい。コンテンツの項目に対する要求（例えば、ＨＴＴＰ要求）がクライアント３１０によってなされたときには、その要求がメディア配送システム３２０のクライアント能力モジュール３２２へルーティングされる。クライアントデバイス３０１から受け取られた要求に応答して、クライアント能力モジュール３２２は、（クライアント）デバイスを識別し、そしてデバイス能力に関して使用可能な情報を得る。この識別に基づいて、クライアント能力モジュール３２２は、データ記憶デバイス３２４からのメディアを表示又は出力するためにクライアントデバイスの能力に関する付加的な情報を検索する。

【００５２】

データ記憶デバイス３２４は、種々のデバイスのメディア出力能力を含む。現在好ましいと思われる実施形態では、この情報を指示するために、対応するデバイス識別子が使用される。データ記憶デバイス３２４に記憶された能力は、スクリーンの解像度に関する情報と、スクリーンのカラー深度と、デバイスのスクリーンディスプレイに適合させるためにイメージを回転すべきかどうかと、以下に詳細に述べる他の情報とを含む。データ記憶デバイス３２４は、新規なデバイスが市場に導入されたときに、そのようなデバイスのプロフィール及びそれらの能力を追加できるように現場でアップグレードすることができる。クライアント能力ログ３２３は、識別することができないか又は能力を入手できないクライアントデバイスの記録を含む。これらのログ記録は、除去されるデバイスを識別できるようにし、これらデバイスに関する情報を得てデータ記憶デバイス３２４に追加できるようにする。

【００５３】

クライアントデバイス３０１の能力が決定された後に、クライアント能力モジュール３２２は、既に変換されたコンテンツを記憶しているフロントサイドキャッシュ３２１を（任意に）見て、オブジェクトが適当なフォーマットで得られるかどうか調べる。フロントサイドキャッシュ３２１は、既に変換されたメディアオブジェクトが、将来の要求に応答して供給するために保持された（任意な）最適なものである。フロントサイドキャッシュ３２１は、最小最近使用(least-recently used)（ＬＲＵ）技術を使用して実施されて、最小最近使用項目を「エージ・アウト(age out)」（即ち除去）するようにしてもよい。クライアント能力モジュール３２２は、適切なオブジェクトがフロントサイドキャッシュ３２１に得られないと決定すると、オンザフライ変換を実行してオブジェクトを供給するようにメディア変換モジュール３２５に要求する。

【００５４】

メディア変換モジュール（ＭＴＭ）３２５は、メディアコンテンツの特定項目の要求をクライアント能力モジュール３２２から特定のフォーマットで受け取る。メディア変換モジュール３２５は、要求されたメディアオブジェクトのオリジナルコピーを得て、それを要求されたフォーマットに変換し、そしてその変換されたメディアオブジェクトをクライアントデバイス３０１に返送する。バックサイドキャッシュ３２７は、高い効率を与える

ための最適なものであり、これも、ＬＲＵ技術を使用して実施されてもよい。インターネットサーバー３３０（又は別のソース）から受け取られたオリジナルオブジェクトは、各要求された項目のコピーを各要求に応答して検索する必要性を回避するためにバックサイドキャッシュ３２７に保持される。このバックサイドキャッシュの使用は、ネットワークを経て送られるコールの数を減少する。又、これは、リモートサーバーからの大きなオブジェクト（高クオリティのカラーイメージ）の検索を回避することにより、メディア変換モジュール３２５による使用可能なオブジェクトの変換及び返送を促進する。

【００５５】

Ｃ．デバイスの能力を検出し、適当なメディアオブジェクトを配送する方法

図４Ａ－Ｂは、接続されたクライアントデバイスの能力を検出し、そしてメディアオブジェクトをこのようなデバイスへ適切なフォーマットで配送するシステムのオペレーションの詳細な方法ステップを示す１つのフローチャートを構成する。ステップ４０１では、インターネットサイトにおけるウェブページのマルチメディアオブジェクトに対するＵＲＬは、このようなオブジェクトが本発明のメディア配送システムによりサービスされるように変更される。現在好ましいと思われる実施形態では、ＵＲＬは、メディア配送システムがインストールされるサーバー名がＵＲＬに予め添付される。例えば、加入者のサイトが、ＵＲＬ：ｈｔｔｐ：／／ｗｗｗ．ｓｕｂｓｃｒｉｂｅｒ．ｃｏｍ／ｉｍｇ／ｌｏｇｏ．ｇｉｆにより通常アクセスされるロゴを含む場合には、変更されたＵＲＬがｈｔｔｐ：／／ｅｓｗｉｔｃｈ．ｃｏｍ／ｗｗｗ．ｓｕｂｓｃｒｉｂｅｒ．ｃｏｍ／ｉｍｇ／ｌｏｇｏ．ｇｉｆとなる。

【００５６】

ステップ４０２では、これらの再書き込みされたＵＲＬを含むウェブページがオープンされるか、又はクライアントデバイスが再書き込みされたＵＲＬを選択（クリック）するときに、クライアントデバイスからのＨＴＴＰ要求がメディア配送システムへルーティングされる。ステップ４０３では、クライアント能力モジュール（ＣＣＭ）が、ステップ４０１において実行されたエンコードプロセスを逆転し、ソースイメージに対する全ＵＲＬを決定する。現在好ましいと思われる実施形態では、これは、要求ＵＲＬから「／ｅｓｗｉｔｃｈ．ｃｏｍ」を除去することより成る。

【００５７】

ステップ４０４では、ＣＣＭは、ＨＴＴＰユーザーエージェントヘッダをキーとして使用してデータ記憶デバイスからクライアント能力構成を検索する。この構成情報は、表示サイズ、カラー深度、オーディオチャンネル等のクライアントデバイスの再生能力を指定する。この構成情報は、クライアントデバイスの完全な能力を決定するために付加的なＨＴＴＰ要求ヘッダの検査を要求してもよい。このステップ中に収集される情報は、ターゲットデバイスに厳密にどんな能力がサポートされるかＣＣＭが理解できるようにする。特に、この情報は、オリジナルメディアオブジェクトをターゲットデバイスに適したフォーマットに変換するためにどんな特定の変換オペレーションが要求されるかをシステムに指示する。

【００５８】

ステップ４０５では、ＣＣＭは、メディア変換モジュール（ＭＴＭ）に特有のコマンドを含むＵＲＬを作成する。これらのコマンドは、ソースメディア文書又はオブジェクトを変換して、その文書を要求したクライアントデバイスの能力に合致させるようにＭＴＭに命令する。このＵＲＬは、構成ファイルで指定されたＭＴＭサーバーを指す。ＭＴＭモジュールは、ＣＣＭと同じサーバーにあってもよいし、それとは異なるサーバーにあってもよい。（任意の）ステップ４０６において、ＣＣＭは、作成されたＭＴＭ　ＵＲＬに合致するオブジェクトについてフロントサイドキャッシュに問い合わせる。換言すれば、この特定のターゲットデバイスに適するように変換されたメディアオブジェクトのバージョンをフロントサイドキャッシュが既に記憶しているかどうか見て調べる。

【００５９】

現在好ましいと思われる実施形態では、システムの内部で使用されるＵＲＬストリング

は、特定フォーマットの特定オブジェクトに対するインデックスとして働くようにエンコードされる。このように、エンコードされたＵＲＬストリングは、特定フォーマットの特定文書が特定位置に記憶されることを指示することができる。例えば、ＣＣＭは、特性：サイズ＝１００ピクセル、カラー深度＝８及びカラー＝偽をもつｌｏｇｏ．ｇｉｆの変換されたバージョンを含むＵＲＬをチェックすることができる。文書又はオブジェクト（ターゲットデバイスに対して変換された）がフロントサイドキャッシュに既に存在する場合には、ＣＣＭは、その文書をターゲットデバイスに単に返送すればよい。しかしながら、合致する項目がキャッシュに見当たらない場合には、この方法は、ステップ４０７に示すように続けられる。

【００６０】

ステップ４０７では、ＣＣＭがオリジナルクライアント要求を代理処理し、クライアントにより送信されたＵＲＬを、ＣＣＭにより生成された再構成されたＵＲＬに置き換える。このプロセスは、クライアントに対して完全に透過的であり、即ち要求を行なうクライアントデバイスには、要求がＭＴＭへ通されたことが通知されず又はそれに気付かない。むしろ、この転送は、ＣＣＭが、クライアントデバイスによりなされた要求を転送して、ＭＴＭにより満足させるバックエンドプロセスである。ステップ４０８では、ＭＴＭがその構成されたＵＲＬを受け取り、そしてオリジナルメディアオブジェクトに対してバックサイドキャッシュサーバーを経てＨＴＴＰ要求を行なう。オブジェクトがバックサイドキャッシュに存在する場合には、それがローカルディスクからサービスされる。もしそうでなければ、キャッシュサーバーは、オリジナルＵＲＬで識別されたインターネットサイトからオブジェクトを要求し、将来の使用のためにそれをキャッシュ記憶する。この点におけるＭＴＭのタスクは、メディアオブジェクトを、そのオリジナルフォーマットから、ターゲットデバイスに望ましい（ターゲットデバイスの能力に基づいて）フォーマットに変換することである。ステップ４０９では、ＭＴＭは、それが受け取った再構成されたＵＲＬで指定されたメディア変換を実行する。ＭＴＭがこのタスクを実行すると、ステップ４１０において、メディアオブジェクトの新たに変換されたバージョンがクライアントデバイスへ返送され、そして（任意であるが）フロントサイドキャッシュにもコピーされる。

【００６１】

Ｄ．クライアント能力を決定してその情報を供給するシステムの使用

上述した役割にサービスするのに加えて、本発明は、クライアントデバイスの能力を決定しそしてこのクライアント能力情報を他のシステム又はデバイスへ供給するのに使用されてもよい。セルラー電話のようなデバイスがそれらの能力をサーバーにいかに記述しなければならないかの更なる説明については、例えば、ＷＡＰフォーラムから入手できるＷＡＧ　ＵＡＰｒｏｆ（Wireless Application Group User Agent Profile Specification）、ワイヤレス・アプリケーション・プロトコル・フォーラム・リミテッド、プロポーズドバージョン３０－Ｍａｙ－２００１を参照されたい。この仕様のコピーは、インターネットのｈｔｔｐ：／／ｗｗｗ１．ｗａｐｆｏｒｕｍ．ｏｒｇ／ｔｅｃｈ／ｄｏｃｕｍｅｎｔｓ／ＷＡＰ－２４８－ＵＡＰｒｏｆ－２００１０５３０－ｐ．ｐｄｆにおいて現在見ることができる。同様の仕様が、ワールドワイドウェブコンソーシアム（Ｗ３Ｃ）から入手できる１９９９年7月２７日、Ｗ３Ｃノート、「Composite Capabilities / Preference Profiles (CC/PP): A user side framework for content negotiation」により説明されている。この仕様のコピーは、インターネットのｈｔｔｐ：／／ｗｗｗ．ｗ３．ｏｒｇ／ＴＲ／ＮＯＴＥ－ＣＣＰＰ／．において現在見ることができる。

【００６２】

これらの提案された規格の一方又は両方を、将来の新規なデバイス及びデバイスソフトウェアによりサポートできるが、このような規格に対する現在のサポートは非常に限定されている。これら規格のデバイスサポートが普遍的なものとなるまで、サーバーアプリケーションは、標準化されたデバイス情報の利点を取り入れることができない。この問題は、本発明のシステムを、非準拠のデバイスから到来するＨＴＴＰ要求に対するプロキシーとして働くように構成することにより対処することができる。ＵＡＰｒｏｆ又はＣＣ／Ｐ

Ｐ情報を一部分伴うか又は全く伴わずに要求がメディア配送システムへ転送されたときには、クライアント能力モジュールが、要求されたデータをルックアップし、この情報を要求にアタッチした後に、その要求を最終的な行先へ転送する。これは、システムが、非準拠のクライアントデバイスと、ＵＡＰｒｏｆ、ＣＣ／ＰＰ又は他の同等の規格との準拠を必要とするインターネット及びＷＡＰサイトとの間のブリッジとして働くことができる。

【００６３】

図５は、非準拠のデバイスから到来するＨＴＴＰ要求に対するプロキシーとして働く際のメディア配送システムのクライアント能力モジュールのオペレーションを示すフローチャートである。ステップ５０１では、非準拠のクライアントデバイスからのＨＴＴＰ要求がインターネット又はＷＡＰサイトからシステムへ転送される。説明上、「非準拠」のクライアントデバイスとは、デバイスにその能力を識別することを要求するＵＡＰｒｏｆ、ＣＣ／ＰＰ又は同様の規格に準拠しないデバイスである。

【００６４】

ステップ５０２では、システムのクライアント能力モジュール（ＣＣＭ）が、ＨＴＴＰユーザーエージェントヘッダをキーとして使用してデータ記憶デバイスからクライアント能力構成を検索する。この構成情報は、表示サイズ、カラー深度、オーディオチャンネル等のクライアントデバイスの能力を指定する。構成情報は、付加的なＨＴＴＰ要求ヘッダを検査してクライアントデバイスの完全な能力を決定することを要求する。このステップ中に収集された情報は、ターゲットデバイスにおいてどんな能力がサポートされるかＣＣＭが厳密に理解できるようにする。

【００６５】

ステップ５０３では、ＣＣＭは、特定のクライアントデバイスによりなされた要求を、以下に示すようにこのようなクライアントデバイスの特定の能力に関する情報で補足する。ステップ５０４では、ＣＣＭは、クライアントの能力に関する詳細を含む補足された要求を、このようなクライアント要求で指定された行先へ返送する。

【００６６】

このシステムをいかに使用してデバイス能力情報を要求に添付するかの一例を以下に示す。システムに転送される到来要求の一例は、次の通りである。

【００６７】

```
GET /index.wml HTTP/1.1
Host: www.lightsurf.com
Accept-Charset: ISO-8859-1
Accept-Language: en
x-up-subno: pegli_pegli-nt4.office.lightsurf.com
x-upfax-accepts: none
x-up-uplink: none
x-up-devcap-smartdialing: 1
x-up-devcap-screendepth: 1
x-up-devcap-iscolor: 0
x-up-devcap-immed-alert: 1
x-up-devcap-numsoftkeys: 2
x-up-devcap-screenpixels: 171,108
x-up-devcap-msize: 8,18
User-Agent: UP.Browser/3.1-UPG1 UP.Link/3.2
```

【００６８】

示されたように、到来情報は、例えば、カラーサポート（上記デバイスについては「０」即ちなし）及びスクリーンピクセル（上記デバイスについては１７１ｘ１０８ピクセル）を含むデバイス能力を報告する。

【００６９】

上記要求を受け取るのに続いて、クライアント能力モジュールは、上述したように特定のクライアントデバイスの能力を決定する。ＣＣＭは、次いで、この情報を要求にアタッ

チし、そしてその補足された要求をその最終的な行先に転送する。ＣＣＭにより添付された情報を示す要求の一例は、次の通りである。

【００７０】

```
GET /index.wml HTTP/1.1
Host: www.lightsurf.com
Accept-Charset: ISO-8859-1
Accept-Language: en
User-Agent: UP.Browser/3.1-UPG1 UP.Link/3.2
x-wap-profile: http://www.eswitch.com/profiles/0A3F362B.xml
```

【００７１】

この場合に、「０Ａ３Ｆ３６２Ｂ．ｘｍｌ」は、ＵＡＰｒｏｆ又はＣＣ／ＰＰプロフィール情報を含む発生された文書である。この要求に対するＵＡＰｒｏｆファイルの一例は、次の通りである。

【００７２】

```
<?xml version="1.0"/>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:prf="http://www.wapform.org/profiles/UAPROF/ccppschema-20010430#">
  <rdf:Description id="WAP-enabled cellular phone">
    <rdf:type
resource="http://www.wapform.org/profiles/UAPROF/ccpschema-
20010430#Hardware
Platform"/>
    <prf:ScreenSize>171x108</prf:ScreenSize>
    <prf:ColorCapable>No</prf:ColorCapable>
    <prf:NumberOfSoftKeys>2</prf:NumberOfSoftKeys>
    <prf:ScreenSizeChar>8x18</prf:ScreenSizeChar>
    ...
  </rdf:Description>
</RDF>
```

【００７３】

発生された文書は、デバイスのＨＴＴＰ要求、並びにメディア配送システムにより維持されたデータ記憶デバイス又は知識ベースからの情報でポピュレーションされる。システムは、クライアントデバイス特性の知識ベースを維持するので、ＨＴＴＰヘッダの情報を単にトランスコード化するサーバーよりも著しく完全なＵＡＰｒｏｆ又はＣＣ／ＰＰ文書を生成することができる。

【００７４】

Ｅ．ＣＣＭ及びＭＴＭモジュールの詳細な動作方法

１．クライアント能力モジュール

クライアント能力モジュール（ＣＣＭ）は、ＨＴＴＰ要求からクライアントデバイスを識別する。ＣＣＭは、要求において供給される情報を、データ記憶デバイスに記憶されたメディア出力能力情報と一緒に使用して、特定のクライアントデバイスのメディア出力能

力を決定する。これは、ＣＣＭが、当該項目（例えば、メディアオブジェクト）を要求している特定形式のクライアントデバイスに対して最適な送信サイズ及び再生フォーマットを決定できるようにする。例えば、ＨＴＴＰブラウザは、ブラウザ名（例えば、「ネットスケープナビゲータ」又は「マイクロソフトインターネットエクスプローラ」）、ブラウザバージョン、及びそれがサポートするグラフィック形式を指示することができる。この情報は、非常に限定されたグラフィックサポート（例えば、ＪＰＥＧサポートのみ）を有するセットトップボックスにおいて実行されるブラウザのようにグラフィックサポートに限度のある場合に有用である。ターゲットデバイスがその能力を指示することができない場合には、少なくとも、そのデバイスクラス、例えば、パームハンドヘルドデバイス、セットトップボックス、ＷＡＰブラウザをもつ電話等を指示する。

【００７５】

デバイスクラスに基づいて、ＣＣＭは、デバイス能力データ記憶デバイスからデバイスの能力に関する情報を得る。例えば、デバイスクラスは、特定モデルのパームハンドヘルドデバイスでよい。この情報に基づいて、ＣＣＭは、データ記憶デバイスに維持された知識ベースにおけるデバイスクラスをルックアップすることにより、デバイスの能力、例えば、表示能力（例えば、１６色）、デバイスメモリ（例えば、４－８ＭＢ）、及び表示サイズ（例えば、３００ｘ５００）を見分けることができる。

【００７６】

又、ＣＣＭは、未識別のクライアントをＣＣＭのログに記録すると共に、知識ベースができるだけ最新のものとなるよう確保するために規則的な間隔で通知を与える方法も含む。新たなクライアントデバイスが導入されたときには、知識ベースの構成情報を更新して、これら新たなクライアントデバイスに関する情報を追加することができる。ＣＣＭは、その構成ファイルを更新するための「プッシュ」又は「プッシュ／プル」機構をサポートする。「プッシュ」機構は、置き換え構成ファイルを含むデータ記憶デバイスに送信される機密ＨＴＴＰ　ＰＯＳＴ要求又はＳＭＴＰメッセージで構成される。「プッシュ／プル」機構は、ＨＴＴＰ　ＧＥＴ要求又はＳＭＴＰメッセージをデータ記憶デバイスへ送信することより成り、これは、サーバーがローカル構成ファイルの更新をスケジュールするようにさせる。

【００７７】

ＣＣＭは、主としてＨＴＴＰヘッダ、特にユーザーエージェントヘッダにおいてクライアントを識別するように応答する。又、ＣＣＭは、ＨＴＴＰより下のプロトコル層の情報も検査する（例えば、要求のＩＰパケットの原点）。デバイスが識別されると、ＣＣＭは、ハイアラーキー構成ファイル（データ記憶デバイスにおける）に問い合わせ、これは、デバイス能力に対するデフォールト値を供給すると共に、アクセプトヘッダのような付加的なヘッダを指定し、これを使用して、メディアコンテンツに対する適切な出力フォーマットを決定することができる。デバイスの能力が決定されると、ＣＣＭは、適切な再フォーマット情報を含む指定のメディア文書に対してＭＴＭへの要求を構成する。次いで、ＣＣＭは、再構成された要求と共にＭＴＭへクライアント接続を転送する。

【００７８】

現在好ましいと思われる実施形態では、ＣＣＭは、全ＨＴＴＰへのアクセス要求がクライアントによりなされるようなアパッチモジュールとして実施されてもよい。この要求に含まれた情報を使用して、構成ファイルに対する一連の問合せによりその要求を行なうクライアントデバイスを識別する。ＨＴＴＰユーザーエージェントヘッダは、要求を発しているクライアントの主たる指示子である。ユーザーエージェントは、ＨＴＴＰ／１．０及びＨＴＴＰ／１．１の両方における任意のヘッダであるが、実際には、全てのウェブクライアントソフトウェアが、各要求に応じてそのヘッダにおけるある識別情報を送信する。又、多数のクライアントが、それらのデバイスの物理的能力を記述するカスタムヘッダも送信する。例えば、移動電話に使用されるＷＡＰブラウザであるＵＰブラウザ（カリフォルニア州レッドウッドシティのオープンウェーブシステムズインクにより供給されるアンワイヤード・プラネットブラウザ）は、要求において「非標準」ヘッドを送出する。この

点について「非標準」とは、このようなヘッダがＨＴＴＰ仕様によってカバーされないことを意味する。これらヘッダの一例を以下に示す。

【００７９】

```
User-Agent: UP.Browser/3.04-SC02 UP.Link/3.2.3.8
x-up-devcap-charset: US-ASCII
x-up-devcap-immed-alert: 1
x-up-devcap-max-pdu: 1472
x-up-devcap-msize: 8,10
x-up-devcap-numsoftkeys: 2
x-up-devcap-screenchars: 15,5
x-up-devcap-screenpixels: 120,50
x-up-devcap-smartdialing: 1
x-up-devcap-softkeysize: 7
x-up-fax-accepts: none
x-up-fax-limit: 0
x-up-subno: RzOyzSrSj-ARAs01_up2.upl.sprintpcs.com
x-up-uplink: up2.upl.sprintpcs.com
```

【００８０】

上記ヘッダの「x-up-devcap-screenpixels」部分は、特に、クライアントデバイスが巾及び高さにどれほど多くのピクセルを表示できるか（１２０ｘ５０）指示する。上記例に示すヘッダは、この特定のクライアントデバイスの能力に関する多数の詳細を含む。しかしながら、多くの場合に、ヘッダは、これら全ての詳細を含むものではなく、従って、ＣＣＭは、デバイスの特性を得るためにデータ記憶デバイスに記憶された情報を参照しなければならない。しかしながら、ＣＣＭは、もし可能であれば、上記ヘッダの「x-up-devcap-screenpixels」部分のようなヘッダの情報を使用する。

【００８１】

現在好ましいと思われる実施形態では、データ記憶デバイスにおけるＣＣＭ構成ファイル（又は知識ベース）がＸＭＬに書き込まれ、その言語のハイアラーキー特徴の利点が取り入れられる。このファイルは、一連の＜ｕｓｅｒ－ａｇｅｎｔ＞エントリーで構成される。＜ｕｓｅｒ－ａｇｅｎｔ＞エントリーは、例えば、次のように現われる（行番号は、参照に過ぎない）。

【００８２】

```
 1    <user-agent header='User-Agent' pattern='UP.Browser' >>
 2      <device-class>wap</device-class>
 3      <content-type pattern='^image/'>
 4        <capability name='color-depth' default='8'/>
 5        <capability name='display-height' default='100'>
 6          <header name='x-up-devcap-screenpixels'
pattern='(\d+),\d+'/>
 7        </capability>
 8        <capability name='display-width' default='100'>
 9          <header name='x-up-devcap-screenpixels'
pattern='\d+,(\d+)'/>
10        </capability>
11        <capability name='output-format' default='image/bmp'/>
12      </content-type>
13    </user-agent>
```

【００８３】

一般に、タグ属性名パターンは、これら属性に与えられる値が規則的表現として処理されることを指示する。規則的な表現から情報を抽出しなければならない場合には、規格「マッチ・リメンバー(match remember)」シンタックス（かっこ）が使用される。例えば、上記９行目において、パターン属性は、ストリング「１２０、５０」に一致し、そしてサブストリング「５０」を使用して包囲能力値をセットすることを指示する。

【００８４】
ＣＣＭは、要求を受け取ると、構成ファイルを通って、ヘッダ属性で指定されたＨＴＴＰヘッダの値を、値属性で指定されたストリングと比較することにより、各＜ｕｓｅｒ－ａｇｅｎｔ＞タグを順次にチェックする。ほとんどの場合に、＜ｕｓｅｒ－ａｇｅｎｔ＞タグは、ＨＴＴＰユーザーエージェントヘッダに対して一致され、そしてＨＴＴＰユーザーエージェントヘッダに対してタグを一致することは、ヘッダ属性が削除された場合のデフォールト振舞いである。各＜ｕｓｅｒ－ａｇｅｎｔ＞ブロックは、それが構成ファイルに現われた順序で処理され、従って、限定性の高い値属性をもつ＜ｕｓｅｒ－ａｇｅｎｔ＞タグが、限定性の低い値属性をもつ＜ｕｓｅｒ－ａｇｅｎｔ＞タグの前に現われる。例えば、値＝「ＵＰ．Ｂｒｏｗｓｅｒ／３．１」と読まれる＜ｕｓｅｒ－ａｇｅｎｔ＞タグは、値＝「ＵＰ．Ｂｒｏｗｓｅｒ」と読まれる＜ｕｓｅｒ－ａｇｅｎｔ＞タグの前に配置される。＜ｄｅｖｉｃｅ－ｃｌａｓｓ＞要素は、異なるクライアントをそれらの能力に基づいて任意のグループに分離するのに使用される。＜ｄｅｖｉｃｅ－ｃｌａｓｓ＞値の幾つかの例は、「ＷＡＰ」、「ｉ－モード」、「ＨＤＭＬ」及び「ｊ－フォン」である。

【００８５】
一致する＜ｕｓｅｒ－ａｇｅｎｔ＞タグが見つかると、ＣＣＭは、一般に、ＨＴＴＰ ＨＥＡＤ要求を文書ソースへ発行することにより、ＭＩＭＥ形式のメディア文書が要求されていると決定する。ＭＩＭＥ形式が分かると、ＣＣＭは、＜ｕｓｅｒ－ａｇｅｎｔ＞ブロック内の適当な＜ｃｏｎｔｅｎｔ－ｔｙｐｅ＞ブロックをルックアップする。上述した例では、「ｉｍａｇｅ／」でスタートするＭＩＭＥ形式に対するいかなる要求も、第１の＜ｃｏｎｔｅｎｔ－ｔｙｐｅ＞タグのパターン属性に定義された規則的表現に一致する。実際の構成ファイルは、各々のサポートされるメディア形式又はサブ形式に対して個別のブロックを有してもよい。

【００８６】
クライアント能力は、１つ以上の＜ｃａｐａｂｉｌｉｔｙ＞タグにより＜ｃｏｎｔｅｎｔ－ｔｙｐｅ＞タグ内に定義される。各メディア形式は、異なる能力を要求してもよい。イメージを適切に表示するために、表示巾、高さ、及びカラー深度を知ることが必要である。適当なオーディオ流を供給するために、帯域巾と、デバイスが多数のチャンネルを再生できるかどうかとを知る必要がある。最も単純なケースでは、構成ファイルは、強制的なデフォールト属性により各能力に対して既に記憶された値を与える。上記例の４行目及び１１行目は、このケースを示し、全ＵＰ．Ｂｒｏｗｓｅｒユーザエージェントに対するカラー深度及び出力フォーマットを、８ビット／ピクセル及びｉｍａｇｅ／ｂｍｐに各々セットする。上述したように、あるユーザエージェントは、非標準的ＨＴＴＰヘッダの形態でサーバーへ付加的な情報を与える。＜ｈｅａｄｅｒ＞タグを＜ｃａｐａｂｉｌｉｔｙ＞タグの内側で使用して、これらのヘッダをパーズすると共に、ヘッダ値に基づいてデバイス能力をセットするようにＣＣＭに命令することができる。この例では、６行目に示されたように、非標準的な「x-up-devcap-screenpixels」ヘッダがもしあればこれを使用して、パターン属性に与えられた規則的表現をそのヘッダ値に適用することによりデバイス表示巾をセットしなければならない。ヘッダ値の部分を分離して能力を指定するためにかっこが使用される。

【００８７】
上記では特に示さなかったが、デバイスのクラス又は形式からその基礎となる通信搬送を推測してもよい。例えば、ターゲットデバイスがセルラー電話である場合には、システムは、基礎となる通信搬送がワイヤレスであると推測してもよい。別の例として、ターゲットデバイスが、ＷＡＰを使用して通信するページャーである場合には、システムは、ターゲットデバイスが、限定された帯域巾で（セルラー電話に比して）ワイヤレス通信を使用すると推測してもよい。デバイスクラス及び到来する要求に基づいて、システムは、通常、通信搬送がワイヤレスであるか有線であるか見分けてもよい。更に、ワイヤレス及び有線の両方の能力を有するデバイスは非常に僅かである。典型的な有線接続は、Ｔ１、ＤＳＬ、ケーブルモデム、及びダイヤルアップ接続を含む。ワイヤレスサイドでは、典型的

な接続は、９６００ボーの回路交換データコール、９６００ボーのパケット交換データコール、又は新たな６４ＫボーのＧＰＲコールを含む。

【００８８】

又、クライアント能力モジュールは、第三者ではなく、許可された参加サイトのコンテンツの再フォーマットにしかシステムを使用できないように、オリジナルコンテンツのソースを照合する役割も果たす。コンテンツを変換すべきところの指定のサーバーの名前を含む特定のＵＲＬに応答してＣＣＭをアクチベートするだけとすることにより、セキュリティが施行される。

【００８９】

ＣＣＭは、特定のクライアントデバイスの能力に関してそれが導出した情報を使用して、要求されたメディア文書に対しメディア変換モジュールへの要求を構成する。このＣＣＭは、適切な再フォーマット情報及びクライアント接続を含むこの構成された要求をＭＴＭへ転送する。クライアント能力モジュールは、変換されたメディア文書のためにフロントサイドキャッシュを（任意に）実施する。このフロントサイドキャッシュは、構成された要求がＭＴＭへ転送される前に、変換された文書がその要求の基準を満たすことについて問い合わせがされる。このフロントサイドキャッシュの目的は、ＭＴＭモジュールの負荷を最小にすることである。

【００９０】

２．メディア変換モジュール

メディア変換（又はトランスコード化）モジュール（ＭＴＭ）は、フォーマット命令を要求パラメータとして含むメディア文書に対するＨＴＴＰ要求を受け入れて、メディアをそれらの命令に基づいて再フォーマットする。ＭＴＭは、イメージ又はビデオのような単一メディア形式に特有なものであってもよいし、又は多数の形式のメディアをサポートしてもよい。基本的なオペレーションにおいて、メディア変換モジュールは、像を次のＭＩＭＥ形式へ及びそれら形式から変換し：ｉｍａｇｅ／ｊｐｇ、ｉｍａｇｅ／ｂｍｐ、ｉｍａｇｅ／ｇｉｆ、ｉｍａｇｅ／ｔｉｆｆ、ｉｍａｇｅ／ｗｂｍｐ、ｉｍａｇｅ／ｉｆｆ、ｉｍａｇｅ／ｐｃｘ、及びｉｍａｇｅ／ｐｎｇ、更に、イメージの寸法を減少すると共にイメージの寸法を増加し、クライアント表示のアスペクト比に適合するようにイメージの回転をサポートし、そしてイメージのカラー深度の減少及びイメージのカラー深度の増加をサポートすることにより、イメージの再フォーマットをサポートする。又、ＭＴＭは、オーディオファイル及びストリームの両方を次のＭＩＭＥ形式へ及びそれら形式から変換し：ａｕｄｉｏ／ａｉｆｆ、ａｕｄｉｏ／ａｕ、ａｕｄｉｏ／ｍｐｅｇ、ａｕｄｉｏ／ｗａｖ、そしてオーディオビットレートを減少することにより、オーディオの再フォーマットをサポートする。

【００９１】

ＭＴＭは、ビデオファイル及びストリームの両方を次のＭＩＭＥ形式へ及びそれら形式から変換することにより、ビデオの再フォーマットをサポートする：ｖｉｄｅｏ／ｍｐｅｇ、ｖｉｄｅｏ／ｑｕｉｃｋｔｉｍｅ、ｖｉｄｅｏ／ｘ－ｍｓｖｉｄｅｏ（ＡＶＩ）、ｖｉｄｅｏ／ｘ－ｍｓ－ａｓｆ、ｖｉｄｅｏ／ｒｍ、及びｖｉｄｅｏ／ｍｊｐｅｇ。又、ＭＴＭは、ＲＦＣ２０４６、マルチパーパス・インターネット・メール・エクステンション（ＭＩＭＥ）により定義された付加的なマルチメディアコンテンツ形式及びストリームの再フォーマットもサポートできる。ＲＦＣ２０４６のコピーは、インターネットのｈｔｔｐ：／／ｗｗｗ．ｉｅｔｆ．ｏｒｇ／ｒｆｃ／ｒｆｃ２０４６．ｔｘｔ．において現在入手できる。

【００９２】

メディア変換モジュールのオペレーションを示す一例を以下に示す。この例では、ＭＴＭ受信器は、ＪＰＥＧイメージを変換し、入力として次のものを受信する。

出力の大きさ（巾及び高さ）

出力デバイスの形式

カラースペース（例えば、ＲＧＢ又はグレースケール）

カラーパレット（例えば、真のカラー又はインデックス）、及び
圧縮技術
これらの入力から、ＭＴＭは、指定サイズの指定出力フォーマットでピクチャーを出力することができる。あるデバイスは、それらのネイティブな特性とは異なるように特徴付けされてもよいことに注意されたい。例えば、１６色のＬＣＤディスプレイを伴うデバイスは、真のカラーのデバイスとして特徴付けられるのが好ましい。従って、デバイスの役割は、ＭＴＭにより送信される真のカラーモードイメージを、内部ソフトウェア／ハードウェアを使用してその内部１６色モードに変換することである。所有権又は非所有権構成を含む適当な圧縮技術を使用することができる。例えば、ＪＰＥＧ、ＪＢＩＧ、ＧＩＦ等が含まれる。例えば、１９９５年７月及び１９９５年８月、ドクタードブズジャーナル、「JPEG-like Image Compression」（パート１及び２）（カリフォルニア州サンマテオのドクタードブズジャーナルからドクタードブズ／ＣＤリリース６としてＣＤＲＯＭで入手できる）を参照されたい。

【００９３】

上述したＪＰＥＧイメージを変換するＭＴＭの特定のオペレーションは、次の通りである。第１に、使用されたカラースペースにおけるビットマップを発生するために入力ピクチャーが圧縮解除される。例えば、Ｃｌｉｋｐｉｘは、ＧＵＶカラースペースを使用し、ＪＰＥＧは、ＹＵＶ及びＲＧＢを含むマルチカラースペースをサポートする。ＧＵＶカラースペースは、２０００年１月２１日に出願された共通所有の特許出願第０９／４８９，５１１号に説明されており、工業規格カラースペースの説明もこの特許出願に見ることができる。次いで、ピクチャーは、通常、工業規格のカラースペースの１つにおいて「標準」中間フォーマットへと変換される。例えば、次のものを含む。

Ｌ、ａ、ｂ　１６ビット／ピクセル／チャンネル（例えば、Ａｄｏｂｅ　Ｐｈｏｔｏｓｈｏｐ（登録商標）で使用される）；
ＳＲＧＢ　８ビット／ピクセル／チャンネル（例えば、マイクロソフト、ＨＰ及びその他により使用される）；及び
ＹＵＶ。

【００９４】

次いで、中間フォーマットが、次の処理により出力デバイスで要求されるフォーマットにマップされる。

１．イメージスケーリング：希望の出力サイズにイメージをスケーリングする。
２．モノクロ情報だけが望まれる場合には、標準変換方法を使用して（例えば、Ｒ、Ｇ、Ｂ信号からルミナンス信号Ｙを発生するためにインターナショナル・テレコミュニケーション・ユニオン（ＩＴＵ）推奨勧告を使用して；例えば、ＩＴＵ－推奨勧告ＢＴ．６０１－１　「Encoding parameters of Digital Television for studio」を参照）、イメージのモノクロバージョンが発生される。
３．ビット／ピクセルが８未満である場合には、ディザ技術が使用される（例えば、エラー拡散、ブルーノイズマスキング等。例えば、Recent Progress in Digital Halftoning, 1994, The Society of Imaging Science and Technology, compiled and edited by Reiner Eschbach, ISBN 0-892080-181-3を参照）。
４．出力デバイスがカラーパレット（例えば、２５６カラーのみをサポートする）を有する場合には、カラーディザ機構が使用される。このような機構は、Computer Graphics Principles and Practice, Second Edition, 1990, Foley, van Dam, et al., Addison Wesley, Reading, MA, ISBN 0-201-12110-7にある程度詳細に説明されている。
５．データがストリーミングアウトされる前に圧縮が任意に実行される。真のカラーイメージについては、好ましい圧縮機構がＪＰＥＧである。インデックスされたイメージＧＩＦについては、ＰＮＧが好ましい方法である。ハーフトーンイメージについては、好ましい解決策は、ＪＢＩＧ圧縮である。

【００９５】

最終的に、発生されたピクチャーが出力され、ターゲットデバイスへストリーミングし

てそのデバイスのディスプレイで最終的にレンダリングする準備ができる。
メディアコンテンツの項目を再フォーマットするメディア変換モジュールのオペレーションの一例を、以下の「AutoRotateOp」ファンクションにより示す。

【００９６】

```
 1: #include "autorotateop.h"
 2:
 3: AutoRotateOp::AutoRotateOp()
 4: {
 5: }
 6:
 7: AutoRotateOp::~AutoRotateOp()
 8: {
 9: }
10:
11: const char *AutoRotateOp::Name()
12: {
13:   return "autorotate";
14: }
15:
16: const char *AutoRotateOp::Args()
17: {
18:   return "displayWidth(160),displayHeight(120),clockwise(1)";
19: }
20:
```

【００９７】

```
21: const char *AutoRotateOp::Example()
22: {
23:   return "autorotate=118,157";
24: }
25:
26: void AutoRotateOp::Process(IMG_image *img)
27: {
28:   int32 w, h, cw;
29:   float displayAspect, photoAspect;
30:
31:   GetArg(&w, 160);
32:   GetArg(&h, 120);
33:   GetArg(&cw, 1);
34:
35:   displayAspect = (float)w / (float)h;
36:   photoAspect = (float)(img->width) / (float)(img->height);
37:
38:   if ((displayAspect > 1.0f && photoAspect < 1.0f) ||
39:         (displayAspect < 1.0f && photoAspect > 1.0f))
40:   {
41:         if (cw)
42:               IMG_RotateRight(img);
43:         else
44:               IMG_RotateLeft(img);
45:   }
46: }
47:
```

【００９８】

上記「AutoRotateOp」ファンクションは、特定デバイスの使用可能な表示に良好に適合するようにイメージを自動的に回転する。上記２６行目に示すように、このファンクションは、イメージに対するポインタをパラメータとして受け取る。２８行目から３９行目において、デバイスの表示特性が得られる。３８行目から３９行目の状態は、デバイスの表示に良好に適合するために、イメージをポートレート向きに表示すべきか（即ちイメージを垂直方向に表示するか）、風景向きに表示すべきか（即ちイメージを水平方向に表示するか）評価する。この評価の結果に基づいて、４１行目から４４行目に示すように、ＩＭＧ　ＲｏｔａｔｅＲｉｇｈｔ又はＩＭＧ　ＲｏｔａｔｅＬｅｆｔへのコールがなされ、そして特定デバイスの表示に適合するようにイメージが時計方向又は反時計方向に回転される。ＩＭＧ　ＲｏｔａｔｅＲｉｇｈｔ又はＩＭＧ　ＲｏｔａｔｅＬｅｆｔファンクションの更なる詳細を示すソースコードリスティングが添付資料Ａとして添付されている。

【００９９】

ＭＴＭは、可能なときに、オリジナルメディアオブジェクトのキャッシュ記憶バージョンを使用する。ＭＴＭは、ソースメディアを読み取るのに必要な時間を短縮すると共に、メディア配送システムによるインターネット帯域巾消費を減少するために、バックサイドキャッシュにソースメディアオブジェクトをキャッシュ記憶する。このバックサイドキャッシュ記憶は、ＭＴＭにより行うこともできるし、又は中間の逆プロキシーキャッシュに

より行うこともできる。ソースオブジェクトは、ＨＴＴＰキャッシュ制御ヘッダで指定された命令に基づいてキャッシュ記憶される。現在好ましいと思われる実施形態では、逆プロキシーキャッシュとして構成されたアパッチＨＴＴＰサーバーがインターネットとＭＴＭモジュールとの「間」に配備され、従って、ソースマルチメディア文書に対する要求が最初にローカルディスクキャッシュと比較される。アパッチ逆プロキシーキャッシュモジュールは、優れた拡張性及び保守性を得るためにキャッシュタスクをメディア変換タスクからデカップルする。

【０１００】

Ｃ／Ｃ＋＋プログラミング言語でソースリスティングを含むコンピュータプログラムリスティング添付資料Ａを添付し、本発明を更に説明する。Ｃ／Ｃ＋＋プログラムを生成しそして構築するのに適した環境は、種々の売主から入手でき、カリフォルニア州スコットバレーのボーランド・ソフトウェア・コーポレーションから入手できるＢｏｒｌａｎｄ（登録商標）Ｃ＋＋ビルダー、及びワシントン州レドモンドのマイクロソフト・コーポレーションから入手できるＭｉｃｒｏｓｏｆｔ（登録商標）ビジュアルＣ＋＋を含む。

【０１０１】

１つの好ましい実施形態及び幾つかの変形態様を特に参照して本発明を詳細に説明したが、この特定の実施形態又は特定の変形態様に本発明を限定するつもりはない。例えば、当業者であれば、本発明の教示から逸脱せずに好ましい実施形態に対して種々の変更がなされ得ることが明らかであろう。

【０１０２】

コンピュータプログラミングリスティング添付資料A

IMGXFORM

```
#include "vimage.h"
#include "imglib.h"

IMGLIB_API void IMG_FlipH(IMG_image *img)
{
	if (img->pixelBytes != 4)
	{
		DEBUG_IMG_FORMAT;
		return;
	}

	int32 x, y, w;
	uint8 *rowPtr;
	uint32	*pixPtr, *pixPtr2, tempPix;

	w = img->width >> 1;
	rowPtr = img->baseAddr;
	for (y = 0; y < img->height; y++)
	{
		pixPtr = pixPtr2 = (uint32 *)rowPtr;
		pixPtr2 += img->width - 1;
		for (x = 0; x < w; x++)
		{
			tempPix = *pixPtr;
			*pixPtr++ = *pixPtr2;
			*pixPtr2-- = tempPix;
		}
		rowPtr += img->rowBytes;
	}
}

IMGLIB_API void IMG_FlipV(IMG_image *img)
{
	img->baseAddr += img->rowBytes * (img->height - 1);
	img->rowBytes = -img->rowBytes;
}

IMGLIB_API void IMG_RotateLeft(IMG_image *img)
{
	if (img->pixelBytes != 4)
	{
		DEBUG_IMG_FORMAT;
		return;
	}

	IMG_image	tempImg;

	if (!IMG_DuplicateImage(img, &tempImg))
		return;

	int32 x, y;
	uint8 *rowPtr, *srcRow, *srcPix;
	uint32	*pixPtr;

	if (img->rowBytes < 0)
	{
		img->baseAddr += (img->height - 1) * img->rowBytes;
		img->rowBytes = -img->rowBytes;
	}
	img->width = tempImg.height;
	img->height = tempImg.width;
	img->rowBytes = img->pixelBytes * img->width;
	rowPtr = img->baseAddr;
	srcRow = tempImg.baseAddr + tempImg.pixelBytes * (tempImg.width -
1);
	for (y = 0; y < img->height; y++)
	{
		pixPtr = (uint32 *)rowPtr;
		srcPix = srcRow;
		for (x = 0; x < img->width; x++)
		{
			*pixPtr = *((uint32 *)srcPix);
```

【０１０３】

```
                pixPtr++;
                srcPix += tempImg.rowBytes;
            }
            rowPtr += img->rowBytes;
            srcRow -= tempImg.pixelBytes;
        }
        IMG_FreeImage(&tempImg);
}

IMGLIB_API void IMG_RotateRight(IMG_image *img)
{
        if (img->pixelBytes != 4)
        {
            DEBUG_IMG_FORMAT;
            return;
        }

        IMG_image   tempImg;

        if (!IMG_DuplicateImage(img, &tempImg))
            return;

        int32 x, y;
        uint8 *rowPtr, *srcRow, *srcPix;
        uint32      *pixPtr;

        if (img->rowBytes < 0)
        {
            img->baseAddr += (img->height - 1) * img->rowBytes;
            img->rowBytes = -img->rowBytes;
        }
        img->width = tempImg.height;
        img->height = tempImg.width;
        img->rowBytes = img->pixelBytes * img->width;
        rowPtr = img->baseAddr;
        srcRow = tempImg.baseAddr + tempImg.rowBytes * (tempImg.height -
1);
        for (y = 0; y < img->height; y++)
        {
            pixPtr = (uint32 *)rowPtr;
            srcPix = srcRow;
            for (x = 0; x < img->width; x++)
            {
                *pixPtr = *((uint32 *)srcPix);
                pixPtr++;
                srcPix -= tempImg.rowBytes;
            }
            rowPtr += img->rowBytes;
            srcRow += tempImg.pixelBytes;
        }
        IMG_FreeImage(&tempImg);
}

IMGLIB_API void IMG_Rotate180(IMG_image *img)
{
        IMG_FlipV(img);
        IMG_FlipH(img);
}

IMGLIB_API void IMG_Resize(IMG_image *img, int32 newWidth, int32
newHeight, bool highQuality)
{
        if (img->pixelBytes != 4)
        {
            DEBUG_IMG_FORMAT;
            return;
        }

        if (img->width == newWidth && img->height == newHeight)
            return;

        IMG_image   tempImg;

        if (!IMG_AllocateImage(&tempImg, newWidth, newHeight, img-
>imgFormat))
            return;
```

【０１０４】

```
//IMG_StretchBlit(img, &tempImg, 0, 0, newWidth, newHeight, true);
      if (highQuality)
      {
            VImage            vImg;
            vImg.Import(img);
            vImg.Scale(newWidth, newHeight, CImageServer::CUBIC);
            vImg.Export(&tempImg);
      }
      else
            IMG_StretchBlit(img, &tempImg, 0, 0, newWidth, newHeight,
false);

      IMG_CopyTags(img, &tempImg);
      IMG_FreeImage(img);
      *img = tempImg;
}

IMGLIB_API void IMG_ClampResize(IMG_image *img, int32 maxWidth, int32
maxHeight, bool highQuality)
{
      float f, aspect, newAspect;
      int32 width, height;

      aspect = (float)(img->width) / (float)(img->height);
      newAspect = (float)maxWidth / (float)maxHeight;
      if (aspect > newAspect)
      {
            width = maxWidth;
            f = (float)width / aspect;
            height = ROUND(f);
            height = CLAMP(height, 1, maxHeight);
      }
      else
      {
            height = maxHeight;
            f = (float)height * aspect;
            width = ROUND(f);
            width = CLAMP(width, 1, maxWidth);
      }
      IMG_Resize(img, width, height, highQuality);
}
```

Apache Module

```
/**
 * This function creates a device capabilities object
 * and constructs the URL that is passed to the
 * Media Transcoding Module.
 *
 * @param r
 * @return
 */
static int uts_rewrite_uri(request_rec *r) {
    int status;

    // skip if already a proxy
    if (r->proxyreq != NOT_PROXY) {
        return OK;
    }

    // skip all but GET requests
    if (strcasecmp("GET", r->method) != 0) {
        return OK;
    }

    // get config file
    uts_dir_config      *cfg =  (uts_dir_config *)
ap_get_module_config(r->per_dir_config, &uts_module);
    uts_server_config   *scfg = (uts_server_config *)
ap_get_module_config(r->server->module_config, &uts_module);

    // (subrequest for content type)
    string contentType = "image/jpeg";
```

【０１０５】

```
    /**
     * first, parse the full incoming URI.  We're going to treat the
path+args
     * of this URI as the full URI for the proxy request
     */
    uri_components uc;
    status = ap_parse_uri_components(r->pool, r->uri, &uc);
    if (status != HTTP_OK) {
        ap_log_error(APLOG_MARK, APLOG_ERR | APLOG_NOERRNO, r->server,
"Bad input URI: %s", r->uri);
        return DECLINED;
    }

    char *path = ap_pstrdup(r->pool, uc.path);

    // pass in headers, get device capabilities back
    DeviceIdentifier::DeviceCapabilities devcap;
    status = dev_ident->getDeviceCapabilities(&devcap, r, contentType);
    if (status != DI_SUCCESS) {
        ap_log_error(APLOG_MARK, APLOG_ERR | APLOG_NOERRNO, r->server,
"Device capabilities lookup failed");
        ap_log_error(APLOG_MARK, APLOG_ERR | APLOG_NOERRNO, r->server,
"Recording headers") ;
        logHeaders (r) ;
        return DECLINED;
    }

    ap_log_error(APLOG_MARK, APLOG_DEBUG | APLOG_NOERRNO, r->server,
"%s", devcap.toString().c_str());

    char *sub;
    // pop off initial slash
    sub = ap_getword_nc(r->pool, &path, '/');   // initial slash

    // if we have an additional subscriber ID, pop it off, too
    if (cfg->subscriber_id_on_url) {
        sub = ap_getword_nc(r->pool, &path, '/');   // subscriber ID
    }

    //
    // Build option list.
    // TODO:  This should really be a function that returns a string.
    //
    char *cfg_clamp, *cfg_mimetype, *cfg_path, *cfg_quality,
*cfg_argparam, *cfg_rotate, *cfg_scale, *cfg_filesize ;
    char blank = '\0' ;

    // Clamp or scale?
    if (!devcap.scale_to_width)
    {
      cfg_clamp = ap_psprintf (r->pool, "%sclampsize=%d&",
        (strchr (cfg->media_transcoder_uri, '?') != NULL ? "&" : "?"),
        (devcap.display_height > devcap.display_width ?
devcap.display_height : devcap.display_width));
      cfg_scale = ap_psprintf (r->pool, "%c", blank) ;
    } else {
      cfg_clamp = ap_psprintf (r->pool, "%s",
        (strchr (cfg->media_transcoder_uri, '?') != NULL ? "&" : "?"));
      cfg_scale = ap_psprintf (r->pool, "limitsize=%i,0&",
devcap.display_width) ;
    }

    // Path
    if (path[strlen(path)-2] == '.')
    {
      switch (path[strlen(path)-1])
      {
        case 'b':
          cfg_path = ap_psprintf (r->pool, "in=\"http://%smp\"&", path)
;
          break ;
        case 'g':
          cfg_path = ap_psprintf (r->pool, "in=\"http://%sif\"&", path)
;
          break ;
        case 'j':
```

```
            cfg_path = ap_psprintf (r->pool, "in=\"http://%spg\"&", path)
;
            break ;
          case 'p':
            cfg_path = ap_psprintf (r->pool, "in=\"http://%sng\"&", path)
;
            break ;
          case 't':
            cfg_path = ap_psprintf (r->pool, "in=\"http://%sif\"&", path)
;
            break ;
        }
      } else {
        cfg_path = ap_psprintf (r->pool, "in=\"http://%s\"&", path) ;
      }

      // Mime Type
      cfg_mimetype = ap_psprintf (r->pool, "outformat=%s&",
devcap.mime_type.c_str()) ;

      // Quality
      if (devcap.image_quality != -1)
        cfg_quality = ap_psprintf (r->pool, "outquality=%i&",
devcap.image_quality) ;
      else
        cfg_quality = ap_psprintf (r->pool, "outquality=%i&",
devcap.color_depth) ;

      // Rotate?
      if (devcap.rotate_to_fit)
        cfg_rotate = ap_psprintf (r->pool, "autorotate=%i,%i,0&",
          devcap.display_width,
          devcap.display_height) ;
      else
        cfg_rotate = ap_psprintf (r->pool, "%c", blank) ;

      // Arguments
      if (r->args)
        cfg_argparam = ap_psprintf (r->pool, "%s&", r->args) ;
      else
        cfg_argparam = ap_psprintf (r->pool, "%c", blank) ;

      // Filesize
      if (devcap.maximum_file_size)
        cfg_filesize = ap_psprintf (r->pool, "filesize=%i&",
devcap.maximum_file_size) ;
      else
        cfg_filesize = ap_psprintf (r->pool, "%c", blank) ;

      //
      // End option list build
      //

      // create URL for lsps
      //
[1]URL[2]Clamp[3]path[4]format[5]quality[6]rotate[7]argparam[8]scale[9]f
ilesize
      char *url = ap_psprintf (r->pool, "%s%s%s%s%s%s%s%s%s",
        cfg->media_transcoder_uri,
        cfg_clamp,
        cfg_path,
        cfg_mimetype,
        cfg_quality,
        cfg_rotate,
        cfg_argparam,
        cfg_scale,
        cfg_filesize) ;

      ap_log_error(APLOG_MARK, APLOG_DEBUG | APLOG_NOERRNO, r->server,
"proxying: %s", url);

      // double-check that this is a valid URL
      status = ap_parse_uri_components(r->pool, url, &uc);
      if (status != HTTP_OK) {
          ap_log_error(APLOG_MARK, APLOG_ERR | APLOG_NOERRNO, r->server,
"Bad proxy URI: %s", url);
```

【0107】

```
        return DECLINED;
    }

    // internal redirect won't work for remote servers, so
    // set this up as a proxy
    /* example of working request member variables
    r->proxyreq     = PROXY_PASS;
    r->uri          = "/lsps";
    r->filename     = "proxy:http://pegli-
nt4.office.lightsurf.com:10104/lsps";
    r->args         =
"clampsize=100&in=http://www.lightsurf.com/images/misc/pk_sunset.jpg";
    r->handler      = "proxy-server";
    */

    r->proxyreq     = PROXY_PASS;
    r->uri          = ap_pstrdup(r->pool, uc.path);
    r->filename     = ap_pstrcat(r->pool, "proxy:", uc.scheme, "://",
uc.hostinfo, uc.path, NULL);
    r->args         = ap_pstrdup(r->pool, uc.query);
    r->handler      = "proxy-server";

    return DECLINED;
}
```

DeviceIdentifier

```
/**
 * DeviceIdentifier.cpp
 * implementation of DeviceIdentifier and DeviceCapabilities classes
 * 4/23/2001 pae
 */

#include <string>
#include <strstream.h>
#include <stdlib.h>

#include "httpd.h"
#include "http_log.h"

#include "Regex.hpp"
#include "DeviceIdentifier.hpp"

static Regex regex;

DeviceIdentifier::DeviceIdentifier(const char * filename) {
    try {
        config = new XMLConfigFile(filename);
    }
    catch (...) {
    }
}

DeviceIdentifier::DeviceIdentifier() {
    config = new XMLConfigFile("/lsurf/uts/conf/devices.xml");
}

DeviceIdentifier::~DeviceIdentifier() {
    if (config != NULL) {
        delete config;
    }
}

/**
 * Retrieve a named capability value from the configuration
 * file.  This function will first find the &lt;capability&gt;
 * node identified by <code>capabilityName</code>, then
 * determine if any &lt;header&gt; tags apply.  The
 * value is returned as a string, which can then be
 * converted to the appropriate type.
 *
```

【0108】

```
 * @param r
 * @param baseXPath
 * @param capabilityName
 * @return
 */
string DeviceIdentifier::getCapability(request_rec *r, string baseXPath,
string capabilityName) {

    /**
     * The device capabilities data storage is implemented as an XML
file.
     * A member variable named "config" allows us to make XPath queries
     * against the XML file.
     */
    string cquery =
    baseXPath +
    string("/capability[@name=\"") +
    capabilityName +
    string("\"]");

    string value;

    // first, look through headers (if any)
    XMLConfigFile::StringVectorType headernames = config->query(cquery +
string("/header/@name"));
    for (unsigned int i=0; i < headernames.size(); i++) {
        const char *headerval = ap_table_get(r->headers_in,
headernames[i].c_str());
        if (headerval != NULL) {
            // found a matching header, so grab the pattern and run the
regex
            XMLConfigFile::StringVectorType patterns = config-
>query(cquery + string("/header[@name=\"") + headernames[i] +
string("\"]/@pattern"));
            if (regex.group(patterns[0], string(headerval), &value)) {
                // yes, there is a match between the <header pattern="">
and the header value
                if (value.size() > 0) {
                    // value now contains the matched substring
                    break;
                } else {
                    // there was a pattern match, but no parenthesized
substring, so use the value of the "default" attribute
                    XMLConfigFile::StringVectorType defaults = config-
>query(cquery + string("/header[@name=\"") + headernames[i] +
string("\"]/@default"));
                    value = defaults[0];
                }
            }
        }
    }

    // if we didn't get anything from the headers, use the default value
    XMLConfigFile::StringVectorType defaults = config->query(cquery +
string("/@default"));
    if (defaults.size() > 0) {
        value = defaults[0];
    }

    // TODO: if there isn't a default, die a horrible death
    // this may not be necessary if we publish a schema for the config
file

    return value;
}


/**
 * Convert the result of <code>getCapability()</code>
 * to an integer.
 *
 * @param r
 * @param baseXPath
 * @param capabilityName
 * @return
```

【０１０９】

```
 */
int DeviceIdentifier::getCapabilityInt(request_rec *r, string baseXPath,
string capabilityName) {
    string value = getCapability(r, baseXPath, capabilityName);
    if (value.size() > 0) {
        return atoi(value.c_str());
    } else {
        return -1;
    }
}


/**
 * Convert the result of <code>getCapability()</code>
 * to a boolean.
 *
 * @param r
 * @param baseXPath
 * @param capabilityName
 * @return
 */
int DeviceIdentifier::getCapabilityBool(request_rec *r, string
baseXPath, string capabilityName) {
    string value = getCapability(r, baseXPath, capabilityName);
    return(!value.compare("true") || !value.compare("1"));  // anything
but true or 1 is considered false
}


/**
 * Fill in a DeviceCapabilites structure based on the
 * HTTP request headers and the content type of the
 * requested document.  The main job of this function
 * is to find the correct node in the config file for
 * the given User-Agent header, then call
 * <code>getCapabilityStr()</code>, <code>getCapabilityInt()</code>,
 * and <code>getCapabilityBool()</code> to fill in
 * the device capabilities.
 *
 * @param devcap
 * @param r
 * @param contentType
 */
int
DeviceIdentifier::getDeviceCapabilities(DeviceIdentifier::DeviceCapabili
ties *devcap, request_rec *r, string contentType) {

    if (devcap == NULL) {
        // sorry, no can do
        ap_log_error(APLOG_MARK, APLOG_ERR, r->server, "Cannot call
getDeviceCapabilities() with a null DeviceCapabilities parameter");
        return DI_ERROR;
    }

    XMLConfigFile::StringVectorType xpath_results;
    string base_xpath_query;

    // find the correct user-agent node
    const char *ua_header = ap_table_get(r->headers_in, "User-Agent");

    xpath_results = config->query("//user-agent/@pattern");
    int found = FALSE;
    for (unsigned int i=0; i < xpath_results.size(); i++) {
        if (regex.match(xpath_results[i], string(ua_header))) {
            base_xpath_query.append("//user-agent[@pattern=\"");
            base_xpath_query.append(xpath_results[i]);
            base_xpath_query.append("\"]");
            found = TRUE;
            break;
        } else if (regex.last_error.size() > 0) {
            // this will complain a lot if there are regex errors
            ap_log_error(APLOG_MARK, APLOG_ERR, r->server, "Regular
expression error: %s", regex.last_error.c_str());
```

【０１１０】

```
            ap_log_error(APLOG_MARK, APLOG_DEBUG, r->server, "Regex:
%s", xpath_results[i].c_str()) ;
        }
    }
    if (!found) {
        ap_log_error(APLOG_MARK, APLOG_DEBUG, r->server, "Could not find
configuration entry for User-Agent \"%s\"", ua_header);
        // TODO: log or send email
        return DI_NOTFOUND;
    }

    // within that node, find the correct mime-type node
    xpath_results = config->query(base_xpath_query + string("/mime-
type/@pattern"));

    found = FALSE;
    for (unsigned int i=0; i < xpath_results.size(); i++) {
        if (regex.match(xpath_results[i], contentType)) {
            base_xpath_query.append("/mime-type[@pattern=\"");
            base_xpath_query.append(xpath_results[i]);
            base_xpath_query.append("\"]");
            found = TRUE;
            break;
        } else if (regex.last_error.size() > 0) {
            // again, here's another big complainer
            ap_log_error(APLOG_MARK, APLOG_ERR, r->server, "Regular
expression error: %s", regex.last_error.c_str());
            ap_log_error(APLOG_MARK, APLOG_DEBUG, r->server, "Regex:
%s", xpath_results[i].c_str()) ;
        }
    }
    if (!found) {
        ap_log_error(APLOG_MARK, APLOG_DEBUG, r->server, "Could not find
<mime-type> entry in config file for document MIME type \"%s\"",
contentType.c_str());
        // TODO: log or send email
        return DI_NOTFOUND;
    }

    ap_log_error(APLOG_MARK, APLOG_DEBUG, r->server, "Successfully
identified device, User-Agent: \"%s\"", ua_header);

    // fill in the DeviceCapabilities structure
    // note defaults are:  string="", int=-1, bool=false
    devcap->mime_type           = getCapability(r, base_xpath_query,
"output-mime-type");
    devcap->display_width       = getCapabilityInt(r, base_xpath_query,
"display-width");
    devcap->display_height      = getCapabilityInt(r, base_xpath_query,
"display-height");
    devcap->color_depth         = getCapabilityInt(r, base_xpath_query,
"color-depth");
    devcap->image_quality       = getCapabilityInt(r, base_xpath_query,
"image-quality");
    devcap->color_capable       = getCapabilityBool(r, base_xpath_query,
"color-capable");
    devcap->rotate_to_fit       = getCapabilityBool(r, base_xpath_query,
"rotate-to-fit");
    devcap->scale_to_width    = getCapabilityBool(r, base_xpath_query,
"scale-to-width");
    devcap->palette           = getCapability(r, base_xpath_query,
"palette");
    devcap->maximum_file_size = getCapabilityInt(r, base_xpath_query,
"maximum-size");
    devcap->video_frame_rate    = getCapabilityInt(r, base_xpath_query,
"video-frame-rate");
    devcap->audio_encoding_rate = getCapabilityInt(r, base_xpath_query,
"audio-encoding-rate");
    devcap->audio_channels      = getCapabilityInt(r, base_xpath_query,
"audio-channels");

    return DI_SUCCESS;
};
```

【０１１１】

```
/*************************************************************************
********/

// utility functions to print out the DeviceCapabilities object

void DeviceIdentifier::DeviceCapabilities::print(ostream *os) {
    *os << this->toString();
};

string DeviceIdentifier::DeviceCapabilities::toString() {
    std::ostrstream buffer;
    buffer << "DeviceCapabilites {"
    << " mime-type: " << mime_type
    << "; display width: " << display_width
    << "; display height: " << display_height
    << "; color depth: " << color_depth
    << "; color capable? " << (color_capable ? "yes" : "no")
    << "; rotate to fit? " << (rotate_to_fit ? "yes" : "no")
    << "; video frame rate: " << video_frame_rate
    << "; audio encoding rate: " << audio_encoding_rate
    << "; audio channels: " << audio_channels
    << " }"
    << "\0";
    return buffer.str();
};

ostream &operator<<(ostream &os, DeviceIdentifier::DeviceCapabilities
&devcap) {
    devcap.print(&os);
    return os;
};
```

【図面の簡単な説明】

【０１１２】

【図１】本発明のソフトウェア実施プロセスが具現化されるコンピュータシステムのブロック図である。

【図２】コンピュータシステムのオペレーションを制御するためのソフトウェアシステムのブロック図である。

【図３】本発明のオンラインメディア配送システムを示すブロック図である。

【図４Ａ】ターゲットデバイスの能力を決定し、そしてコンテンツを変換して適当なフォーマットでターゲットデバイスへ配送するシステムの詳細な方法ステップを示すフローチャートである。

【図４Ｂ】ターゲットデバイスの能力を決定し、そしてコンテンツを変換して適当なフォーマットでターゲットデバイスへ配送するシステムの詳細な方法ステップを示すフローチャートである。

【図５】非準拠のデバイスから到来するＨＴＴＰ要求に対してプロキシーとして働く本発明のクライアント能力モジュールのオペレーションを示すフローチャートである。

【図１】

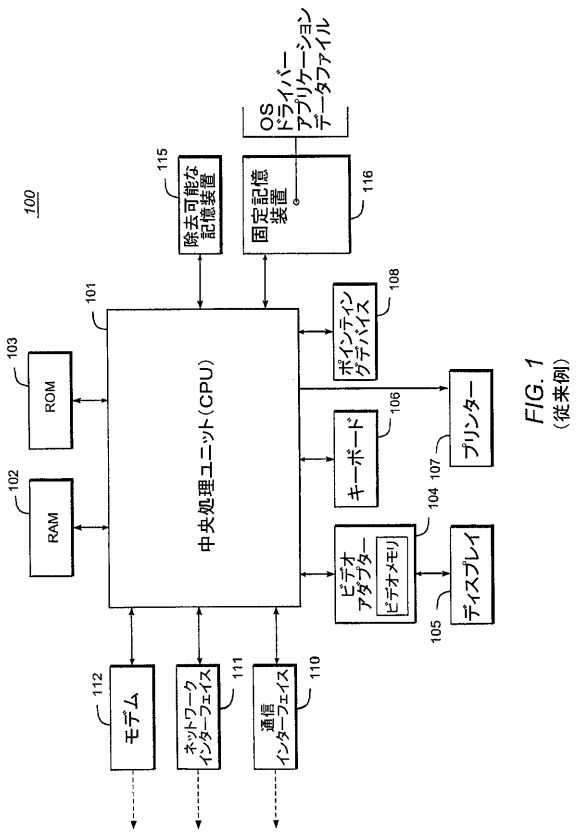


FIG. 1
（従来例）

【図２】

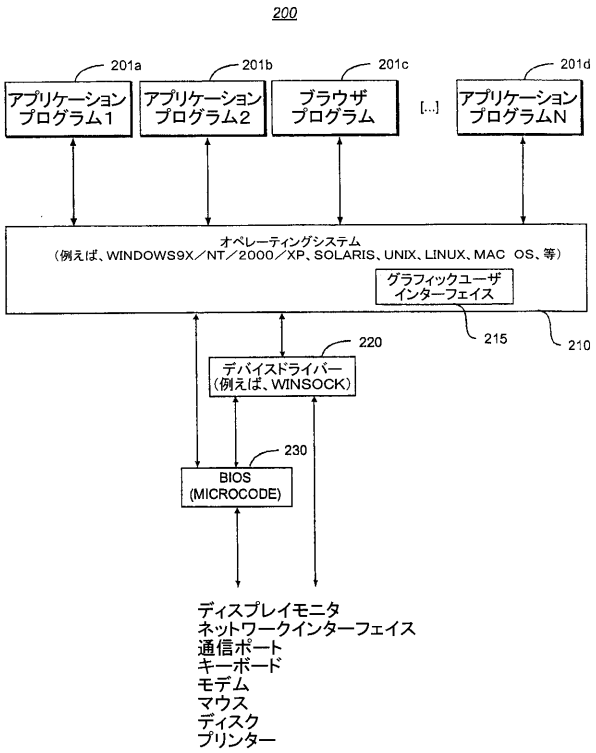


FIG. 2

【図３】

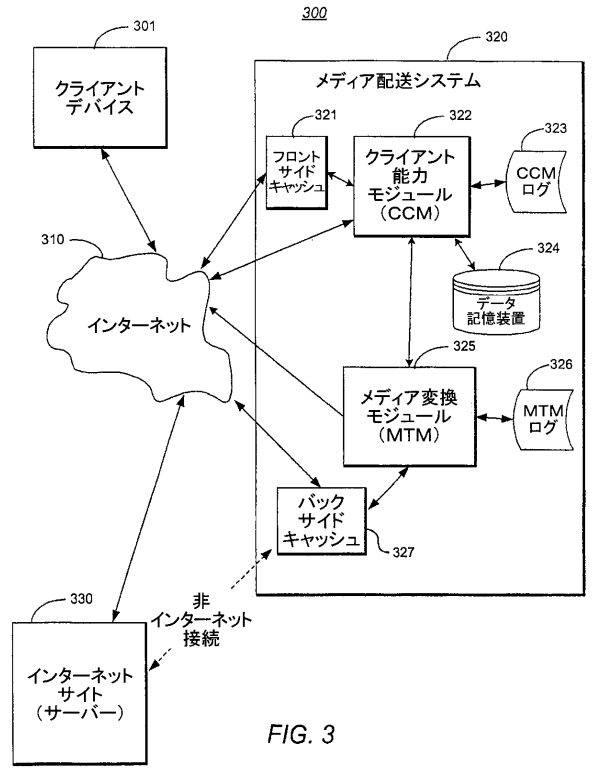


FIG. 3

【図４Ａ】

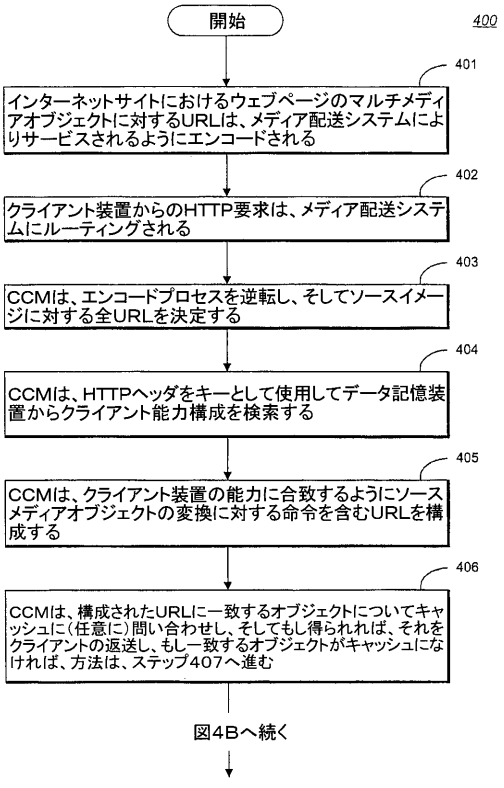


FIG. 4A

【図４Ｂ】

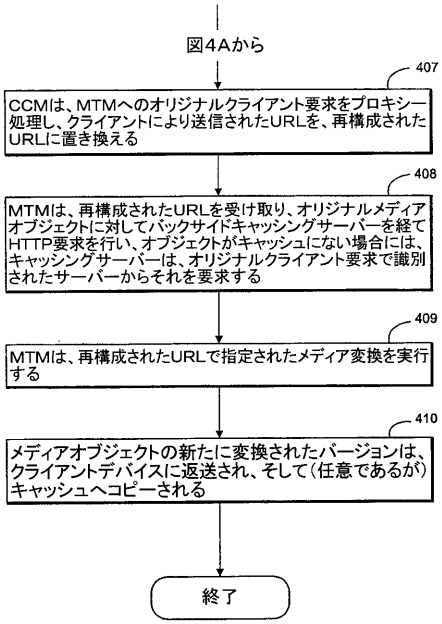


FIG. 4B

【図５】

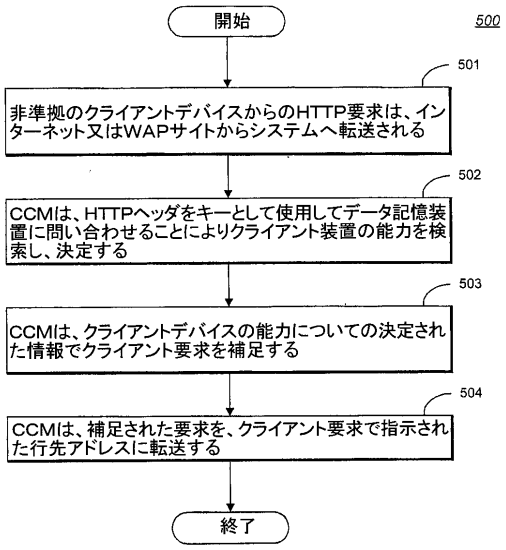


FIG. 5

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Int ernal Application No
PCT/US 02/36064

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 278 449 B1 (ZHOU JOE ZEXUAN ET AL) 21 August 2001 (2001-08-21) column 1, line 19 -column 5, line 8	1-5, 39-43
Y		6
X	EP 1 109 371 A (RES IN MOTION LTD) 20 June 2001 (2001-06-20) the whole document	1-5, 39-43
Y	US 6 300 947 B1 (KANEVSKY DIMITRI) 9 October 2001 (2001-10-09) the whole document	6
X	DE 100 50 172 A (IBM) 26 April 2001 (2001-04-26) the whole document	77-82

-/--

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 October 2003	27 11. 2003
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Kalabic, F

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PC I/US 02/36064

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	DE 199 34 787 A (DEUTSCHE TELEKOM MOBIL) 8 February 2001 (2001-02-08) the whole document	77-82
A	US 6 167 441 A (HIMMEL MARIA AZUA) 26 December 2000 (2000-12-26) the whole document	77-82
A	US 6 161 140 A (MORIYA KOJI) 12 December 2000 (2000-12-12) the whole document	77-82
A	FRANKLIN REYNOLDS; JOHAN HJELM: "Composite Capability/preference Profiles (CC/PP): A user side framework for content negotiation" INTERNET, 'Online! 27 July 1999 (1999-07-27), XP002257669 Retrieved from the Internet: <URL:www.w3.org/TR/NOTE-CCPP> 'retrieved on 2003-10-14! cited in the application the whole document	77-82

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

national application No.
PCT/US 02/36064

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 7-38, 44-76
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☒ No protest accompanied the payment of additional search fees.

International Application No. PCT/US 02/36064

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-6, 39-43

   Distillation of the informations according capabilities of a target device

2. Claims: 77-82

   Informing a destination

International Application No. PCT/US 02 /36064

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box I.2

Claims Nos.: 7-38, 44-76

In view of the large number and also the wording of the claims presently on file, which render it difficult, if not impossible, to determine the matter for which protection is sought, the present application fails to comply with the clarity and conciseness requirements of Article 6 PCT (see also Rule 6.1(a) PCT) to such an extent that a meaningful search is impossible. Consequently, the search has been carried out for those parts of the application which do appear to be clear (and concise), namely claims 1-6, 39-43 and 77-82.

The applicant's attention is drawn to the fact that claims, or parts of claims, relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure.

INTERNATIONAL SEARCH REPORT

Int oral Application No
PCT/US 02/36064

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6278449 B1	21-08-2001	AU 5695199 A	27-03-2000
		EP 1116096 A1	18-07-2001
		JP 2002524791 T	06-08-2002
		WO 0014625 A1	16-03-2000
EP 1109371 A	20-06-2001	US 6477529 B1	05-11-2002
		CA 2326875 A1	06-06-2001
		EP 1109371 A2	20-06-2001
US 6300947 B1	09-10-2001	GB 2339374 A ,B	19-01-2000
		JP 2000066868 A	03-03-2000
		KR 2000011423 A	25-02-2000
DE 10050172 A	26-04-2001	CA 2317082 A1	15-04-2001
		DE 10050172 A1	26-04-2001
DE 19934787 A	08-02-2001	DE 19934787 A1	08-02-2001
		AT 239272 T	15-05-2003
		AU 6685000 A	13-02-2001
		CA 2391322 A1	01-02-2001
		CZ 20020307 A3	15-05-2002
		WO 0108097 A1	01-02-2001
		DE 50001993 D1	05-06-2003
		EP 1200933 A1	02-05-2002
US 6167441 A	26-12-2000	CN 1225479 A	11-08-1999
		GB 2331600 A ,B	26-05-1999
		JP 3184802 B2	09-07-2001
		JP 11194983 A	21-07-1999
		TW 449707 B	11-08-2001
US 6161140 A	12-12-2000	AU 709158 B2	26-08-1999
		AU 4397997 A	24-04-1998
		CA 2238320 A1	09-04-1998
		EP 0870388 A1	14-10-1998
		WO 9815091 A1	09-04-1998
		JP 3350553 B2	25-11-2002
		JP 2000514582 T	31-10-2000
		US 6345298 B1	05-02-2002

Form PCT/ISA/210 (patent family annex) (July 1992)

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,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,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,OM,PH,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,UZ,VN,YU,ZA,ZM,ZW

（特許庁注：以下のものは登録商標）
Ｍａｃｉｎｔｏｓｈ
Ｌｉｎｕｘ
マッキントッシュ
Ｓｏｌａｒｉｓ
ＵＳＢ

(72)発明者 ミーツ イーグリ ポール
アメリカ合衆国 カリフォルニア州 ９５０６６ スコッツ ヴァリー ブルーベリー ドライヴ １１６
(72)発明者 キラニ シェカー
アメリカ合衆国 カリフォルニア州 ９５０１０ カピトラ ウォッシュバーン アベニュー １０９
(72)発明者 イースワー ヴェンカット
アメリカ合衆国 カリフォルニア州 ９５０１４ クパーティノ リンダ ヴィスタ ドライヴ １０７３６