

Dharma

さまざまな インターネット端末に コンテンツを適応させる ソフトウェア技術

北山文彦* 広瀬紳一**

日本アイ・ビー・エム(株)東京基礎研究所

* ktyrn@jp.ibm.com ** hiroses@jp.ibm.com

銀行や証券会社が行うオンライン取引や、商品・チケットの販売など、インターネットや関連技術を用いた新しいサービスが続々と開始されている。一方、iモードのような非PCのインターネット端末が普及しはじめている。このようなWebを用いたサービスを提供する側は、PCのように規格化が進み利用形態もほぼ一定しているものだけではなく、画面サイズやユーザインタフェース、通信能力などが異なるさまざまな端末を相手にしなければならなくなってきている。本稿では、このようなサービスを提供するWeb上のコンテンツをソフトウェアととらえ、さまざまなインターネット端末の能力に適応した形で表示・実行を行えるようなソフトウェアの作成技術および実行環境を紹介する。

インターネット端末の多様化—いま何が問題か

ここ数年、多くの企業がインターネット上で、新しい顧客サービスや種々の取引をはじめている。たとえば、物品や保険商品等の購入・問合せ、支払いなどの銀行取引など、生活や仕事にかかわるあらゆることがインターネット上で可能となりつつある。このような情報処理を実装するにあたっては、従来よりも高度なWeb関連技術が必要となる。

たとえば、Webサーバは文書を送り出すファイルサーバとしてだけでなく、より柔軟で動的な情報提示や処理を指示するためのソフトウェア、あるいは、より大きな情報システムの中での処理の一翼を担うアプリケーションサーバとしての位置付けが重要になってきている。また、コンテンツ記述言語としてのHTMLに加えて、データを交換するためのXMLなども標準として定着しつつある。したがって、このようなインターネット上のサービスの作成は、単なるWebコンテンツの編集ではなくアプリケーションソフトウェアの開発であり、データや処理をどう作っていくかというソフトウェア開発技術として考えていく必要がある。インターネットに接続されたWeb端末を対象としたこのようなソフトウェアは、Webアプリケーションと呼ばれる(図-1)。

一方、インターネットの普及を背景に、PC以外の携帯情報機器、携帯電話、オフィス・家庭用機器などがインターネットに接続され使用されるようになってきた。Web閲覧可能な携帯電話の爆発的な普及は記憶に新しい。このような非PC系のインターネット端末からもWebアプリケーションを使用することは一般的になりつつある。では、主にPC向けに作成されたWebアプリケーションがこれらの端末に対してもサービスを提供することに問題はないのだろうか。

図-2は本学会のWebページを通常のPCのブラウザで閲覧したものである。図-3は同じページをiモードの携

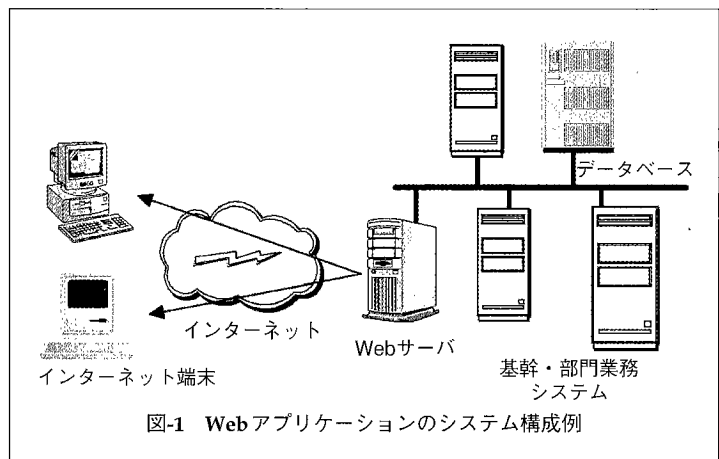


図-1 Webアプリケーションのシステム構成例



図-3 iモード電話による
情報処理学会のページ表示例

図-2 通常のPCによる
情報処理学会のページ表示例

帯電話から閲覧したものである。画面が小さいためすべてを表示しきれないということもあるが、図-3に示したページは基本的に判読困難であり、さらに、容量制限のため受信が途中で打ち切られている。他のページでもよほど単純なものを除いて同様な現象が起きる^{☆1}。

問題の原因は2つある。1つは、技術的・物理的制約から、端末がインターネットに接続される機器として一般に想定されている基準を満たしていないためである。もともと、このようなインターネット上のサービスは、インターネット上の名前付け (URI^{☆2})、プロトコル (HTTP)、および、書式 (HTML) といった標準とそれに従った端末やソフトウェアを使うことにより、特定の技術や製品に依存せずに情報交換や処理が行えるように考えられている。さらに、通常使われる端末の画面サイズや入力装置なども一般的には一定の基準があったといってもよい。ところが、端末の多様化に伴って、この基準に合致しないものが現れるようになってきたのである。たとえば、使用できないHTMLのタグがあったり、バッファ容量が少なかったり、といった現時点における技術的な制約もあれば、画面サイズが小さいというような物理的な要因に基づくものもある。

画面が出ないもう1つの理由はコンテンツそのものにある。インターネットが広く普及するにつれてももとの技術が想定していなかったコンテンツ表現の多彩化が要請されるようになり、マルチメディア関連など種々の拡張や特定の機能に依存した使い方がされるようになってきている。このため、端末としては通常求められる一般基準を満たしているにもかかわらず、コンテンツの方がそれ以上の機能を仮定して作成されているために、す

べての端末で同じ処理を行うことができないのである。

たとえば、現実のWebページには、内容に必須ではない画像の多用や本来の意味を持たないような段組みを行うためだけのTABLEタグの使用などが多数見られる。しかし、このようなページは、画像の表示に制約のあるデバイスや画面サイズ・形状に著しく差異のあるものでは作成者の意図に反した表示結果となり、端末によっては表示不能となる。ただし、インターネットが普及しはじめた当初は、限られたワークステーションやPCのみからアクセスされるだけであり、デバイス能力の多様性の問題は表面化することがなかったのである。

特に、複雑な開発を要し動的に実行処理を行わなければならないWebアプリケーションにとって、このような端末の多様化とコンテンツ表現の多彩化に対応することは単純ではない。能力や性能としてわずかな差異しかないような多くの種類の端末に合わせてアプリケーション開発者がすべて対応するとすると、組合せの爆発によりその開発・保守コストは莫大なものになる。さらに、インターネットというオープンな環境では、どのようなデバイスの端末からアクセスされるかや利用される環境などはあらかじめ予測不可能な面があり、利用状況の変化も激しい。このため、実行ソフトウェアはより動的で柔軟な処理が求められている。本稿で紹介する研究の目的は、このような複雑さをソフトウェア技術を用いて低減することである。

☆1 iモード用のページ記述言語はコンパクトHTMLという通常のHTMLのサブセットであり、(株)NTTドコモのガイドにしたがってWebページを作成する必要がある。

☆2 IETF RFC1630/2396で定められたリソースの共通標記方法。その一部であるインターネット上の位置を表すURLはよく知られている。

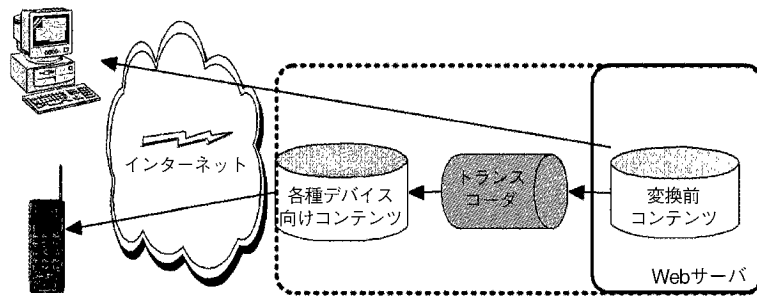


図-4 トランスコーダを使用したWebシステム

サービス (URL)	説明
iモード	無線ネットワーク-インターネット接続
EZweb	無線ネットワーク-インターネット接続, WML/HTMLからHDMLへの変換
J-Sky ウェブ	無線ネットワーク-インターネット接続, HTMLから変換
ドットi	無線ネットワーク (PHS) -インターネット接続
H*Link	無線ネットワーク (PHS) -インターネット接続, メールプロトコル-HTTP間の変換
WebTV	TV受像機を用いたWeb端末向けプロバイダ. TV受像機に合わせたコンテンツ変換
モネ	独自プロトコルによるカーナビ向けプロバイダ・情報サイト. 情報・メールの音声変換
Yahoo! Mobile (http://mobile.yahoo.com/home)	マルチデバイスサイト (英文, 日本語あり http://mobile.yahoo.co.jp/) ただし, iモードとJ-Skyのみ, 携帯電話, PDA, Pager, 音声接続あり (USのみ)
AOL Anywhere (http://mobile.aol.com/)	Webまたは専用端末 (ハードまたはソフト) によるマルチデバイスサイト (英文), 携帯電話 (WMLのみ), PDA, Pager, TV, 音声接続あり (USのみ)
Google (http://www.google.com)	検索サイト. 他サイトも含む携帯電話 (iモード, WML) 向けコンテンツ変換
Excite (http://k.excite.co.jp/hp/)	各種携帯電話向け検索サイト
7dream.com (http://phone.7dream.com/)	各種携帯電話向けショッピングサイト (HTML, HDML)

表-1 各種変換を行うインターネット接続サービスと多種端末に対応したWebページの例

コンテンツを端末に適応させるしくみ

Web上のコンテンツを各種端末に合わせて適応させる方法について、詳しく見てみる。まず、既存のWebコンテンツをそのデバイス能力に合わせて変換をするということが考えられる。この場合は、すでに存在するWebサーバとWebブラウザを持つ端末の間に変換器を設置し、端末に合わせて変換された文書を出力する。ページを1対1で変換するだけでなく、1ページで受け取ったものを複数ページにして出力してもよい。なお、コンテンツそのものも解析し変換することからセキュリティや著作権等の問題によりWebサーバと一体となった構成をとることが多い。このような変換器は、もともとコンテンツの記述・符号化方式(コード)を主に変換することから、トランスコーダと呼ばれる(図-4)。

このような方式のソフトウェアは各社から製品としていくつか出荷されている。また、各種端末からインターネットに接続するためのゲートウェイサービスやインターネット上で利用できるサイトなどがいくつかある。PC以外の端末に対応して各種変換等を行うサービスを表-1に挙げておく^{☆3}。

- このような変換器で考えられるものとして、
- (1) HTTP等のプロトコル変換、
 - (2) 言語や機能、容量などの制限回避ための変換、
 - (3) 異なるマークアップ言語や画像・音声データ等の記述・符号化方式の変換、

- (4) 操作性向上のための入力・操作方法の変換・追加、
- (5) 画面表示能力や対話方式等に合わせた画面分割・再構成、
- (6) 情報の抽出、要約、注釈などの意味情報処理や自然言語翻訳、

などがある。なお、変換には、変換先端末に関する情報だけでなく、(1)、(2)、(3)に関しては対象元コンテンツのプロトコルや記述・符号化方式に関する情報が必要であり、(4)、(5)に関しては構造情報が必要になる。(6)に関しては、コンテンツの意味内容が必要になる。たとえば、もとのHTML文書に文書構造の情報が正しく含まれていなければ、(5)の構造を再構成するような変換は難しい。なお、プロトコルは各ネットワークに固定的であり、記述・符号化方式に関する情報はコンテンツにあらかじめ付加されているため、(1)、(2)、(3)の変換に必要な情報は既知であることが多い。

トランスコーダによる変換方式は、既存システムをほとんど改変することなく実現可能であるので、主に情報提示を行うようなWebサイトや単純なメールを多種端末向けに変換するような場合に適した方式であるといえる。ただし、もとのコンテンツにその構造や意味内容に関する情報が含まれていることが一般的には期待できないので^{☆4}、構造や意味内容を反映したより柔軟な変換

^{☆3} 利用するには契約が必要なものがある。

^{☆4} 注釈と呼ばれる情報をコンテンツに付加することによりこのような情報を含めることができる¹⁾。

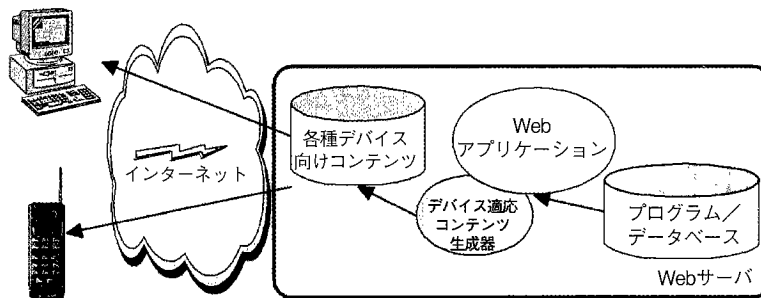


図-5 コンテンツ生成器を含んだWebアプリケーションサーバ

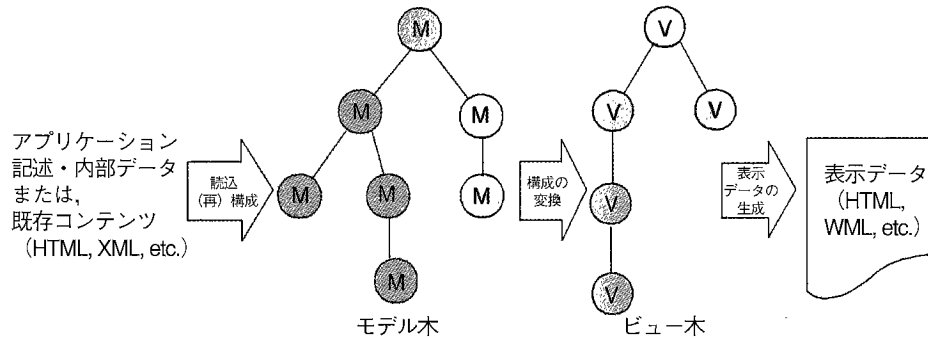


図-6 Dharmaの基本構造と動作

には向かない。すなわち、トランスコードはプロトコルや記述・符号化方式の変換や種々の制限回避のような比較的単純な変換に向いているといえる。

一方、トランスコードとは異なり、最初からコンテンツを端末に適合させて動的に生成する方式も考えられる(図-5)。この場合は従来のWebサーバに組み込まれて、サブレットなどのWebアプリケーションから呼び出されるというのが一般的なシステム構成である。トランスコードとの大きな違いは、変換元のコンテンツが既存の記述・符号化方式による外部文書ではなく、コンテンツの構造や意味内容を参照できるような内部データから変換を行うということである。変換内容はトランスコードと同様に記述・符号化方式の変換、構造の変換などが考えられるが、コンテンツの構造情報や意味情報を取得できる可能性が高く、より高度な変換が可能である。

Dharmaとは

Dharma²⁾は、Web上のコンテンツやWebアプリケーションの表示を、さまざまな端末向けに最適化するための技術である。その基本的なアイデアは、『開発者は個々の端末の属性等に依存しない形でデータや処理の記述を行い、Dharmaの実行時システムが端末の情報を用いて適当な形式の表示データを生成する』というものであり、多種の端末を対象としたアプリケーションの開発や保守のコストの削減を目的にしている。Dharmaは、前章で取り上げた変換器と生成器の両方に応用できる技術である。

図-6にDharmaの基本的な動作を示す。Dharmaが端末に適應した表示データを生成するには、次の3つのステップを行う。

- (1) アプリケーション等の記述から、表示すべきオブジェクト(データとその処理)の木構造であるモデル木を作成する。
- (2) モデル木の現時点で表示されるべき部分に対して、ビュー木を生成する。
- (3) ビュー木の各ノードが端末に適應したHTML文書等の出力を行う。

以下ではこれらのステップの詳細を説明する。

まず、第1のステップは、アプリケーション実行時に必要となるアプリケーションのオブジェクトからなるモデル木を作成する。これは、端末の物理的制約等から独立したものであり、どのような端末に対しても同じものとなる。また、モデル木は開発者によって設計されたアプリケーションの記述や実行時データから作られるものであり、動的に決まる表示を扱うための仕組みが備わっている。たとえば、データベースにアクセスするようなアプリケーションであれば、個々の数値データはもちろん、表示すべきデータの個数自体が実行のたびに異なる可能性がある。

モデル木はある一時点でのユーザインタフェースを表現するだけのものではなく、アプリケーション全体の処理の流れを含んだものであるという点にも注意されたい。すなわち、モデル木のルートからの構造は、アプリケーションを構成する処理の単位(プロセス)の関係を示すものになっている。基本的には、あるアプリケー

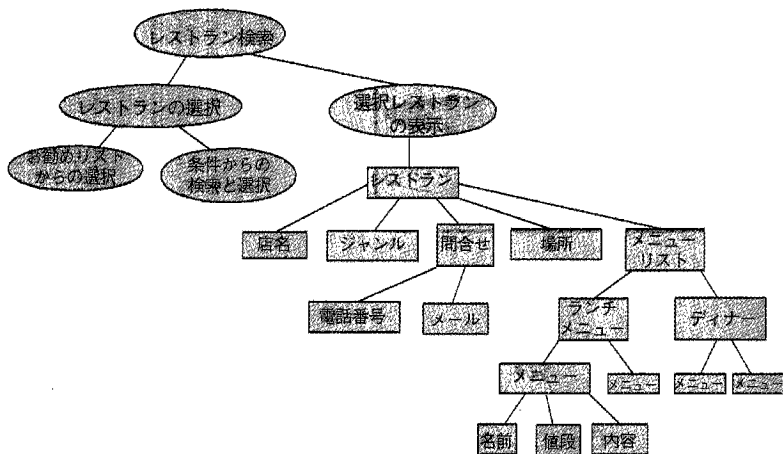


図-7 モデル木の例—レストラン検索アプリケーション

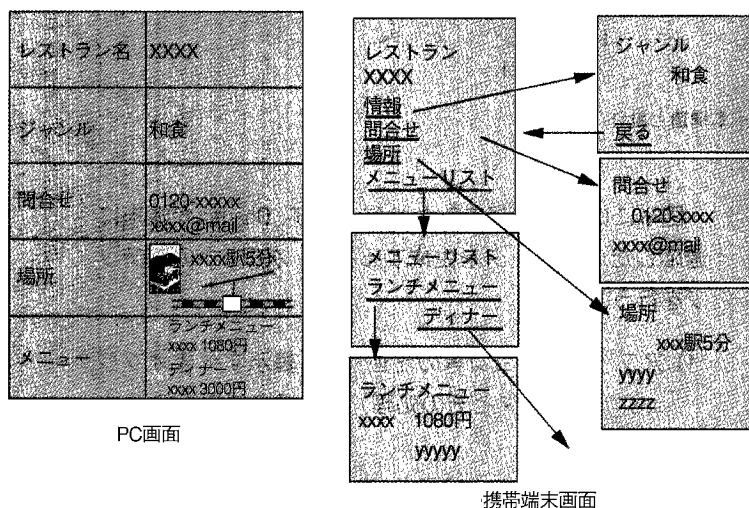


図-8 レストラン情報の表示例

ションは子ノードであるプロセスを順番に実行するものとして解釈される。たとえば、図-7に示したレストラン検索アプリケーションのモデル木では、「レストランの選択」と「選択されたレストランの表示」という2つのプロセスが存在している。さらに、「レストランの選択」は2つのサブプロセスから構成されることも表現している。図-7では楕円でこれらのプロセスを示す。

モデル木の残りの部分は、ある一時点でのアプリケーションデータやユーザインタフェースの論理構造を表現するものである(図-7では長方形で示す)。ここでは、関連の深い要素やデータほど近接するように木を構成する。これは、後のステップにおいて最終的な出力の単位を決定する際のヒントとして用いられる。

以上のようにモデル木が生成されると、出力生成の次のステップとして、現在実行中のプロセスの部分木に対応するビュー木を作る^{☆5}。Dharmaの実行時システムは、ライブラリの中から出力文書の記述言語や望ましい出力ページの大きさ等に応じてビューオブジェクトを選

択することにより、さまざまなビュー木を生成することができるようになってきている。したがって、一般にビュー木は端末の種類ごとに異なったものが生成されることになる。

たとえば、画面が小さくスクロールも遅い端末や、受信できる文書の大きさに制限がある端末では、出力される文書を小さくする必要があるが、これは、モデル木のうち表示対象の一部分については、ビューオブジェクトを生成しないことによって実現することができる。このとき、出力されない子ノードを持つビューオブジェクトは、実際の出力へのハイパーリンクを自動的に付加するといった処理も行う。図-8は、異なる端末によってレストラン検索アプリケーションの画面が異なるよう出力される例である。この例で、モデル木としては同一の状態であるにもかかわらず、分割された携帯端末画面において次々と異なる画面がユーザに表示されるのは、ビュー木がユーザとのインタラクションを制御しているためである。

最後のステップでは、ビュー木の各ノードが対応するモデル木のノードを参照しながら、自身に対応する部分の出力を生成していく。マークアップ言語の文書出力する場合、基本的には、各ノードは、1つ(場合によっては、2つ以上)のマークアップ要素に対応する出力を担当している。このステップにおいても、たとえば、端末の能力にあったイメージデータを参照させるために、出力文書中のURLを異なったものにする、もしくは、イメージデータを埋め込むのではなくハイパーリンクで参照させる、といった適応処理が行われる。

以上に述べてきたようなモデルとビューを分離させたシステムの構成は、MVCモデルとしてよく知られたものである^{☆6}。さらに、ビューの割り当ては実行時システムが自動的に行うため、開発者は個々の端末の差異を意識せずにアプリケーションを開発することが可能になっている。また、モデル木やビュー木のためのコンポーネントはクラスライブラリとして提供されており、アプリケーション特有の要求によってカスタマイズされたものも含めて、コードの再利用性を高めることができるようになってきている。この結果、端末の多様性に起因する開発や保守のコストを低減させることができると考えている。

☆5 これは、Dharmaの実行時システムが自動的に行うものであるが、既定の動作をカスタマイズするためのしくみも開発者に提供されている。

☆6 ビューオブジェクトはコントローラを兼ねている。

適用事例－携帯電話向けオンライン銀行取引アプリケーション

Dharmaのフレームワークの実装はJava言語を用いて行われているが、そのベースとなる実行機構にはいくつかのものが存在する。たとえば、図-7のようなWebアプリケーションを実現するには、Webサーバ上のサーブレットとしての実装が適している。また、既存のHTML文書等をモデル木に再構成することによって、トランスコーダとして実装することもできる。ここでは、Webサーバ上にサーブレットして組み込む形で実装されたDharmaについて紹介する。この実装では、Webアプリケーションのモデル木をXML文書で記述し、プログラムとして実行するシステムとなっている。

アプリケーションプログラムは実行システムに読み込まれ、エンドユーザがアプリケーションを開始するごとにモデル木が作成される。エンドユーザからの処理要求(HTTPリクエスト)に含まれる端末上のブラウザプログラム識別情報^{☆7}を用いて、その端末のデバイス情報を端末データベースから得る。その情報を用いて端末に最適なビュー木がそのつど構成され、HTML文書などの出力データが生成・送出される。ユーザからのHTTPリクエストに、入力データやアプリケーションに対する処理要求などが含まれる場合は、出力データ生成の前に必要な処理が行われる。

多種端末向けのソフトウェア技術として実現されたDharmaは、いくつかの都市銀行のiモード向けオンラインバンキングの実装に採用され、運用が行われている。図-9に示すように、システムのフロント部分であるWebサーバ部にDharmaのコアシステムとそのアプリケーションプログラム(XMLとJavaによるプログラム)が位置するシステム構成となっている。およそ1日あたり数千件~1万件のトランザクションをこなしながらも、1年~2年にわたって安定的に動作を続けており、実使用に耐える技術であることが実証されている。さらに、iモード以外の携帯電話やPDAなどにも同一のシステムによるサービスが開始、または予定されている。

これらの適用例においては、多くの種類の端末により早く低コストで対応できるという目的でDharmaが使われ、新しい種類の追加端末にいち早く対応するのに貢献したと思われる。さらに、ネットワークやインターネット端末の多様化がさらに進む今後、Dharmaのような技術の需要が増えていくと期待される。

☆7 HTTPヘッダフィールドの一つであるUser-Agentの値に含まれる。

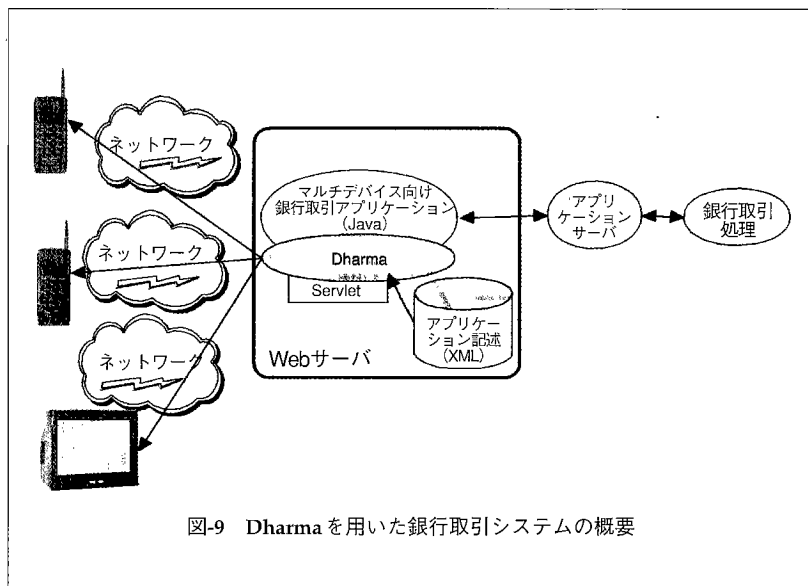


図-9 Dharmaを用いた銀行取引システムの概要

今後の課題

アプリケーションを多くの種類の端末に合わせてどのように作成するかという研究はまだ始まったばかりであり、ソフトウェアとしてどのように構成して実行していくかといったアプリケーションモデルはさらに吟味していく必要がある。たとえば、Dharmaによって自動生成された出力結果をどのように調整するかという問題がある。Dharmaでは、アプリケーション開発者は端末に非依存の論理的なプログラムのみを用意することによって、端末の多様性を吸収しようとしている。しかし、実システムの開発では各端末の画面デザインを開発者が直接行いたいという要求があり、各端末画面固有の調整を行う手段の必要性が認識されている。ソフトウェア開発の複雑さをうまく管理したままで、なおかつ、個々の端末に合わせて柔軟にかつ簡単に調整できるようなプログラミングモデルが求められている。

さらに、日本ではJavaプログラムが実行可能な携帯端末やサービスが出現しつつあり、また、欧米では通常の電話を用いて音声でインターネットにアクセスする技術が実用に供されるようになってきている。さらに、Web上のサービスに関しても新しい道具立てが提唱され標準化されるような動きがある。このように表示装置や入力手段、実行環境の多様化は今後ますます進展すると思われ、これらに柔軟に対応するようなソフトウェア技術はこれから求められていくだろう。

参考文献

- 1) Hori, M., Kondoh, G., Ono, K., Hirose, S. and Singhal, S.: Annotation-based Web Content Transcoding, Proc. WWW9 (2000).
- 2) Kitayama, F., Hirose, S., Kondoh, G. and Kuse, K.: Design of a Framework for Dynamic Content Adaptation to Web-Enabled Terminals and Enterprise Applications, Proc. APSEC'99 (1999).

(平成13年4月27日受付)