

3 T-9 多種端末向けWebアプリケーション・フレームワークDharmaのHTML変形プロクシー・サーバーへの適用の試み

広瀬 紳一 近藤 豪

日本アイ・ビー・エム株式会社 東京基礎研究所

はじめに

我々は、クライアント端末の機能の違いに対応して様々な出力を生成することを可能にする、JavaによるWebアプリケーション構築のためのフレームワークDharma [1][2]の研究を行なっている。Dharmaは、基本的には、Webサーバー上で動くサブレットのためのフレームワークであり、新規のアプリケーション開発を対象としている。しかし、現在のWeb上の種々の資源を有効に活用することを考えると、既存のHTML文書等に対しても、同様の技術を適用できることが望ましい。これは、例えば、画面の狭いクライアント端末のために、長いWebページを分割してその一部のみを送る、といったHTML文書の変形処理となる。本発表では、このような機能を実現するプロクシー・サーバーの構築を、Dharmaのフレームワークを拡張することにより行なった試みについて報告する。

アプリケーション・フレームワークDharma

Dharma においては、アプリケーションはWebサーバー上のサブレットとして動作する。これは、LCMLというXMLベースの言語で記述されたアプリケーションが汎用のサブレットによって読み込まれるという形で構成されている。ここで言う、Webアプリケーションとは、一般にはユーザーとのやりとりが複数回行なわれるものであり、単独のHTTPセッションで完結するものではない。従って、ユーザーからのリクエストによりサブレット内に読み込まれた

LCMLは、アプリケーション・オブジェクトと呼ばれるオブジェクトの木として、アプリケーションの終了時まで保持される。アプリケーション・オブジェクトは、自身で必要なデータを持つ場合もあるし、更にバックエンドのデータベース・システム等と通信を行なうことによってデータを管理する場合もある。

アプリケーション・オブジェクトの木の持つデータを必要に応じてユーザーに提示するために、ビュー・オブジェクトと呼ばれるものが用いられる。すなわち、ユーザーからのリクエストがあると、アプリケーション・オブジェクトのうちで、現在、ユーザーに提示すべき部分に対してビュー・オブジェクトが生成され、それが、アプリケーション・オブジェクトの持つデータからレスポンスとなるHTML文書を生成する。

このとき、クライアント端末の機能に応じて生成するビュー・オブジェクトを調整することによって、その端末に適合した出力を得ることが可能になる。また、生成されたビュー・オブジェクトは、次のリクエストまで保持され、ユーザーが入力したフォームのデータを中継してアプリケーション・オブジェクトの状態の更新を行なうという働きも持っている。

プロクシー・サーバーの構成

プロクシー・サーバーの基本的な考え方は、上記のフレームワークにおける、アプリケーションを定義するLCMLの代わりに、HTMLからアプリケーション・オブジェクトを生成することによって、ビュー・オブジェクトが持つ、端末の機能に応じたHTML文書の出力の機能を利用する、というものである。今回の試作では、IBM AlphaWorks (<http://www.alphaWorks.ibm.com>)に登録されている、WBI (Web Intermediaries)を利用した。

Application of Dharma, a Framework for Web Applications for Pervasive Terminals, to HTML Transforming Proxy Servers

Shin-ichi Hirose, Go Kondo
Tokyo Research Laboratory, IBM Japan, Ltd.

WBIでは、MEG (Monitor, Editor, Generator)と呼ばれる部品群を組み合わせて目的の機能を実現する。Generatorは、リクエストに対して自身の中でレスポンスを生成することによって、通常のWebサーバーと同様の処理を行なうものであり、Editorは、リクエストやレスポンスに変更を加えるものである。また、Monitorは、統計情報の収集等のために、リクエストやレスポンスを観察するが、それらに変更を加えたりすることはない部品である。これらの部品には、固有の起動条件を与えることができる。

ここでは、レスポンスのEditorと、Generatorをそれぞれひとつずつ用いる。まず、レスポンスがHTML文書を含む場合にEditorが作動して、文書からDharmaのアプリケーション・オブジェクトを生成し、クライアント端末に応じたビュー・オブジェクトを用いて、新たなHTML文書を出力する。このとき、例えば、画面が狭い端末であれば、元文書の一部のみからなる文書や目次の文書が生成されたりすることになる。また、この文書は、元文書の他の部分を閲覧するための特殊なアンカーを含んでいる。このアンカーを選択した場合のリクエストに対してはGeneratorが作動し、先に生成したアプリケーション・オブジェクトを利用して、別の部分を含むHTML文書が生成される。

ページの分割処理の概要

今回、実装したWebページの分割は、元文書のDOM (Document Object Model) の構造に基づいたものである。すなわち、処理対象のHTML文書は、まず、[3]のパースャーによって、Javaで実装されたDOMに変換され、更に、その構造を反映したDharmaのアプリケーション・オブジェクトが生成される。分割に際しては、適当な大きさのDOMの部分木に対応する要素が選択され、出力のページを形成する。これは、元文書が<DIV>等を用いて文書の内容に従った構造化がなされていることが前提になっているが、現実には、既存のHTML文書は、そのような構造を持たず、むしろ、<BODY>の下に要素並んでいるだけ、という場合がほとんどである。

基本的な対処法としては、対象となる文書を編集し、<DIV>等を挿入することによって、文書の論理構造を反映したマークアップを追加する、ということになるが、元文書へのアクセスがいつでも可能とは限らず、更に、<DIV>は、スタイルの適用範囲の指定等、別の目的のためのグルーピングに用いられることが多くある、という問題もある。これを解決するために、我々は、HTML文書に対して、その論理的構造等々についての注釈を別のファイルとして与える、という方式を提案している[4]。

おわりに

今回、試作したプロクシー・サーバーは元文書の構造を利用してページの分割を行なうが、実用化のためには、要素が一行に並んだような文書に対して、注釈なしでもある程度要素間の関係を推測して処理を行なえるようなものにしていきたい。また、現在の構成では、元文書の情報がアプリケーション・オブジェクトとして保持されるため、レスポンスを返したあとでも資源を使い続けている。この問題に対しては、セッションの記憶管理を強化して対応し、逆に、アプリケーション・レベルのセッションを有効に利用する、という方向で検討したいと考えている。

参考文献

- [1] 北山, 広瀬, 久世: 多種端末向けWebアプリケーション構築システム: Dharma - システム概要とアプリケーションオブジェクト, 情報処理学会第57回全国大会
- [2] 広瀬, 北山, 久世: 多種端末向けWebアプリケーション構築システム: Dharma - ビューオブジェクト生成とHTML生成機構, 情報処理学会第57回全国大会
- [3] 近藤, 広瀬, 北山, 久世: DOMインターフェースに基づくSGMLパースャー, 情報処理学会第58回全国大会
- [4] M. Hori, et al.: "Annotation of Web Content for Transcoding," W3C Note (<http://www.w3.org/TR/annot>)