

# 2U-06 WWW 環境におけるメタアプリケーション構築支援

倉垣 公一<sup>†</sup> 石川 佳治<sup>††</sup> 北川 博之<sup>††</sup>

<sup>†</sup> 筑波大学 工学研究科 <sup>††</sup> 筑波大学 電子・情報工学系

## 1 はじめに

近年の Web の普及に伴い、Web 上で利用可能なアプリケーションが増大している。例えば、Web 上の検索や電子商取引などであるが、そのサービス内容も様々である。

このような背景のもと、Web 上の複数のアプリケーションを統合して1つの上位アプリケーションを構築することがしばしば必要となる [1]。簡単な例としては、amazon.com 等のサイトで書籍情報を調べた後、図書館の検索システムを用いて当該書籍の有無を調べたいというようなことが考えられる。このため、これらの個別アプリケーションを統合した上位アプリケーションの構築を支援する仕組みが必要となる。

本研究では、その様な上位アプリケーション構築を、メタアプリケーション (メタ AP) とアプリケーションラッパー (APW) という2つの層から成るシステムで支援するアーキテクチャを提案する。

APW は Web サーバとの具体的な要求のやり取りを隠蔽し、アプリケーションを1つの抽象的なオブジェクトとして表現する。従って、その上位にあるメタアプリケーションを構築する際には、個別アプリケーションの詳細とは独立に処理を記述することが可能となり、個別アプリケーションの多目的利用が容易になる。本稿では、対象とするアプリケーションとして、フォームを用いてユーザからの入力を受け取り、CGI (Common Gateway Interface) を用いて HTML ページを作成しそれを返すものに限定する。

## 2 提案システム

### 2.1 APW の導入

CGI のフォームへ要求を提出するための具体的な処理手順を、メタ AP から隠蔽するためのモジュールが APW である。これによって、Web 上のアプリケーションが抽象化可能となる。しかし、アプリケーションから返される実際の HTML ページには要求の結果としては無関係な情報まで含まれていることが多い。従って、メタ AP に結果を返す上では、このページから必要な部分を抽出する必要がある。こうすることによって、メタ AP から見た場合に、要求内容を表す引数を与えると要求の結果が返ってくるオブジェクトとして、アプリケーションが抽象化される。APW は、アプリケーションへの問合せ要求の提出と、検索結果の分解及び抽出による結果の作成の具体的な処理内容をメタ AP から隠蔽するものである。

### 2.2 提案システム概要

本研究で提案するシステム概要を図1に示す。このシステムでは、各アプリケーションの操作内容毎に1つの APW を設ける。また、APW もメタ AP も本研究ではオブジェクト指向プログラミング環境を想定し、オブジェ

Meta-application development support on WWW.  
Kouichi Kuragaki<sup>†</sup>, Yoshiharu Ishikawa<sup>††</sup> and Hiroyuki Kitagawa<sup>††</sup>  
<sup>†</sup> Doctoral Program in Engineering, University of Tsukuba  
<sup>††</sup> Institute of Information Sciences and Electronics, University of Tsukuba

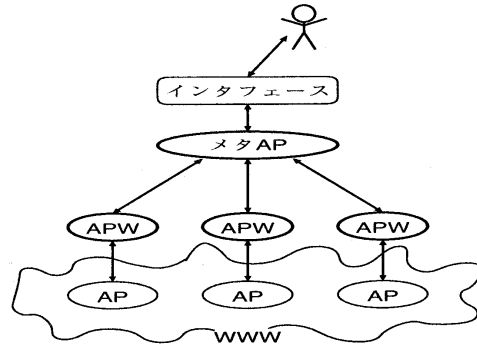


図 1: 提案システム概略

クトとして記述する。また、各 APW の入出力オブジェクトのインタフェースは統一を図る。

それでは、amazon.com の書籍検索の結果得られた書籍が、筑波大学の附属図書館にあるかどうかを Tulips<sup>1</sup> を使用して調べるといった処理を例にして、APW とメタ AP の役割を説明する。メタ AP はユーザから受け取ったキーワードを用いて、amazon.com の APW への入力オブジェクトを生成する。このオブジェクトを受け取った APW は CGI に問合せ要求を提出し、結果の分解及び要求されている結果部分の抽出を行ない、出力オブジェクトを生成する。ここで、出力オブジェクト1つは、1書籍分の結果を表すものとする。一般的には、1回の検索に対して数件の検索結果が存在すると考えられるので、APW からメタ AP へは出力オブジェクトのリストが返される。次に、メタ AP は amazon.com 側の APW から返ってきた結果と Tulips 側の APW への入力オブジェクトとのマッピングを行ない、入力オブジェクトを生成する。このオブジェクトを受け取った Tulips 側の APW は amazon.com 側の APW と同様にして、検索結果をメタ AP へ返す。メタ AP は検索結果を受け取ると、Tulips 側の APW へ次の入力オブジェクトを渡す。

この例では、メタ AP は APW を利用することによって、amazon.com の出力オブジェクトから Tulips への入力オブジェクトを生成する処理を主に記述するだけで良いことになる。

## 3 APW の内部処理

APW における処理は、以下のように3つの部分に分割できる。(1) 第1フェーズ: 入力オブジェクトを受け取り、要求をアプリケーションに送り結果を受け取る、(2) 第2フェーズ: 受け取った結果を解析し、必要な部分を抽出する、(3) 第3フェーズ: 抽出した結果から出力オブジェクトを生成する。

第1フェーズでは、メタ AP から入力オブジェクトが渡されると、APW はこのオブジェクトが持つデータから、name 値と value 値を連結した CGI へのリクエスト文字列を作成する。リクエスト文字列の作成が終了したら、リクエストメソッドによって決められているリクエスト方法で問合せ要求を提出する。要求結果に対応した

<sup>1</sup> 筑波大学附属図書館蔵書目録 (OPAC)

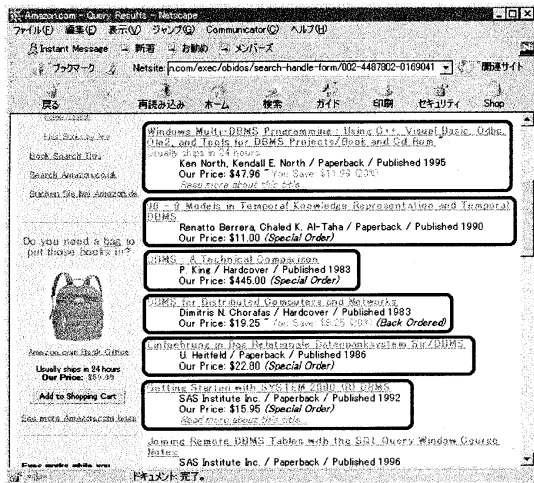


図 2: amazon.com の検索結果

HTML ページが CGI スクリプトから返されたらそれを受け取る。

第 2 フェーズでは、問合せ要求の結果から、結果として必要な部分と結果に無関係の部分とを分離する [2]。例えば、図 2 では矩形部分が必要な部分であり、その他は必要ない部分である。そして、結果に無関係の部分の構造は要求内容によらず一般的に固定であり、結果として必要な部分は結果の件数によって長さが変わると考えられる。しかし、結果として必要な部分の構造は、件数によって構造そのものが変わるのでなく、同様の構造の反復から成るリストを構成していることが多い。例えば、図 2 の 1 つの矩形部分が結果 1 件に対応するが、同様の構造が繰り返されていることが分かる。よって、HTML ページを解析し要求結果から 1 件ずつの情報を抽出するために、HTML のタグ構造を情報として用いる。

次に、amazon.com の書籍検索の結果を例にとると、1 件の結果の中にもタイトルや著者や出版年等の情報がある。そこで、これらの分解と抽出を行なうことを考える。まず、このレベルまで分解されると、HTML のタグにあまり構造が無く、この情報だけでは分解できなくなる。そこで、テキスト内のパターン情報を元に分解し、結果として意味のある文字列を抽出する。

第 3 フェーズでは、第 2 フェーズで抽出された結果から、出力オブジェクトを生成する。ここで、出力オブジェクトは 1 件分の情報を表すオブジェクトである。問合せ要求の結果は、一般的にこのような結果が数件集まったものであるため、得られたこれら全ての出力オブジェクトを要素とする配列を作り、これをメタ AP へ返す。

#### 4 APW 構築支援システム

本研究での APW は Web 上のアプリケーションサービスの内容 1 つに対して 1 つ構築する必要がある。しかし、Web 上に提供されているサービス数は非常に多いため、メタ AP の構築に至るまでに非常に時間がかかる。このため、本研究では APW (入出力オブジェクトを含む) を半自動的に構築する仕組みを提供する。本研究では、APW を構築する際にユーザとインタラクティブな情報の交換を行なうことで、ユーザが直接プログラムを記述することから解放されることを目標としている。このシステムモデルを図 3 に示す。図 3 の APW 構築器が、ユーザやフォーム解析器や APW 構築用パラメタ設定器から得られた情報を元に、APW のプログラムを生成するという

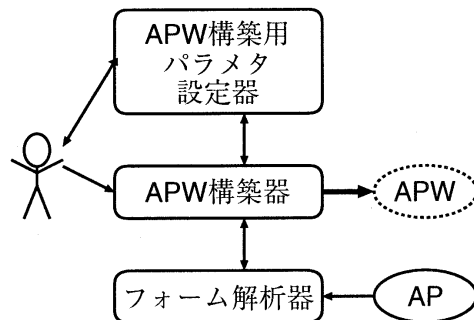


図 3: APW 構築支援システムモデル

仕組みである。

以下では、前節の処理内容で分けたそれぞれの部分に対して、APW の半自動的構築を行なう際に、APW 構築器にどのような情報が必要となるかを説明する。

第 1 フェーズでは、APW を構築する対象となるアプリケーションの URL が分からなければ、フォームを解析することができない。そのため、APW の構築対象であるアプリケーションの URL をユーザから受け取る。

第 2 フェーズでは、2 つの手法を用いて結果の HTML ページを解析する。そのため、各手法でそれぞれどのような構造に基づいて解析を行なうかという情報が必要である。つまり、アプリケーションから渡された HTML ページの解析では、ページ内のどこに現れる構造が結果として必要な構造であるかを知る必要がある。また、1 件分の情報の抽出では、上記の構造中でどこからどこまでの構造が 1 件となるのかを知る必要がある。最後に、出力オブジェクトのデータとなる 結果の抽出では 1 件分のテキストのどのようなテキストパターンが必要な 結果を抽出するための手がかりになるかを知る必要がある。

第 3 フェーズでは、結果の集合を持つ出力オブジェクトを生成するが、これらの情報は APW から与えられるので、プログラムを記述する段階では情報を必要としない。

#### 5 まとめ

ラッパーを用いて Web 上に存在するアプリケーションをオブジェクトとして抽象化することにより、これらを統合利用するメタアプリケーションの構築を容易にするようなモデルを提案した。

今後の課題としては、各機能の詳細な仕様の決定、結果ページの解析及びデータ抽出のための最適な手法の考察、モデルの実装が挙げられる。また、現在の APW はアプリケーションからの結果の状態によって処理内容を選択することができない。このため、APW の処理手順に状態遷移の概念を導入したモデルの設計も考える必要がある。

#### 参考文献

- [1] Hasan Davulcu, Juliana Freire, Michael Kifer, and I.V.Ramakrishnan, A Layered Architecture for Querying Dynamic Web Content, SIGMOD Conf., 1999, pp.491-502.
- [2] D.W.Embley, Y.Jiang, and Y.-K.Ng, Record-Boundary Discovery in Web Documents, SIGMOD Conf., 1999, pp.467-478.