WWW情報統合のための協調型アーキテクチャ

A cooperative architecture for WWW information integration

野田知哉，北村泰彦，辰巳昭治
Tomoya Noda, Yasuhiko Kitamura and Shoji Tatsumi
{tnoda,kitamura,tatsumi}@kdel.info.eng.osaka-cu.ac.jp

大阪市立大学工学部情報工学科*
Department of Information and Communication Engineering
Faculty of Engineering, Osaka City University

Abstract

The Internet has spread over our society, and the WWW has been widely used as a means of information sharing. There are a number of information integration systems like search-engines on the WWW. However as the number of the Internet users increases,the number and the complexity of their demands for information integraion will increase. In this paper, to satisfy these demands we propose a flexible WWW information integration system which can run even on a PC. Our agent-based cooperative architecture makes the integration tasks flexible and robust.

1 はじめに

今日インターネットは社会に急速に普及し，我々の生活に欠かすことのできない基盤の１つになりつつある．中でもWWWは，学術研究，企業広告や電子商取引など 様々な目的で利用されている．これらのWWW情報源はブラウザを介してアクセスし，単独で利用するのが一般的であったが，複数の情報源を組み合わせて利用することができれば，その利用価値が高められる．このような情報統合の試みはすでにいくつかなされてきた．

MetaCrawler[1]は複数の検索エンジンに同じ条件で検索を行い，得られた複数の結果をまとめるメタ検索エンジンである．その検索結果の質は，検索エンジンを単独で使うときと比較して向上している．またShopBot[1]は，インターネット上にある複数のバーチャルショップから価格などの商品情報を抽出し，それらをまとめて表示する．

学術研究の分野では，ヒトゲノム計画のデータベース間で統合が行われている．ヒトゲノム計画では，文献，タンパク質のアミノ酸配列と立体構造，免疫系などの細胞構造といった多種多様なデータがデータベース化され，WWWで公開されている[2]．これらのデータベースは単独で使用するより，互いに参照するとより有用であり，ハイパーリンクによるデータの相互参照が行われている．

人間がブラウザを介して上記と同様の操作をするには，多大な手間と時間がかかる．しかし，情報統合システムによって，検索から結果をまとめるまでの一連の作業を自動的に行うことができれば，利用者はわずかな操作で必要な情報を取得できる．このように情報統合は，利用者の時間の節約と手間の軽減ができる．

Metacrawlerやゲノムデータベースでは，ハイパーリンクによって該当するページを利用者に示すページ単位の統合を行っている．一方，ShopBotは統合の対象となるページから必要とする部分だけを取り出して，それらをまとめている．ShopBotの様なページより細かい単位での統合は，利用者の不要な部分が取り除かれ，ページ単位の統合と比べて情報がより明確になる．

*連絡先：大阪市立大学工学部情報工学科知識情報工学講座
〒558-8585 大阪市住吉区杉本3-3-138

[1] http://www.go2net.com/search.html

[2] http://www.genome.ad.jp

ところで，先に紹介したシステムは，情報提供者，情報利用者以外の第三者が統合を行っているために，それらの仕様はシステム開発者の決定した固定されたものである．そのため，利用者の要求する統合とシステムの仕様が異なる場合，利用者の要求に応じることができない．このように第三者による情報統合は，様々な利用者の要求に応じられないという点で限界がある．

今後，インターネットの利用者は急増することが予想され，利用者の情報統合に関する要求が多岐に渡るのは避けられない．したがって，様々な利用者の要求に応じることができる情報統合システムが必要になる．

そこで本稿では，クライアント側で柔軟にWWW情報源の統合を行ったり，利用者によって統合可能なWWW情報源を追加することができる，エージェントベースの協調的なWWW情報源統合システムを提案する．

２章では，WWW情報統合と我々の提案する情報統合システムの概要を述べ，３章ではWWWエージェントの機能について説明する．４章ではメッセージの参照や，統合について述べる．５章では，統合可能なWWW情報源を増やす方法について提案する．

2 WWW情報統合

本論文において，WWW情報統合とは，以下の操作によって利用者の要求する情報を得ることを指す．

抽出: WWW情報源にアクセスし，そこから必要な情報を取り出す．

統合: 複数の情報源で得られた情報を，利用者の指定した条件で１つにまとめる．あるいは，あるWWW情報源で得た情報を用いて，別のWWW情報源にアクセスする．

保存・表示: 抽出や統合の結果をファイルに保存したり表示する．

このような情報統合を考えると以下の問題点がある．

自律分散した情報源: WWW情報源はインターネット上に分散して存在し，その運営や管理は個別に行われている．情報源へのアクセスは，ブラウザによるアクセスを想定しており，情報抽出や統合を前提としている情報源は少ない．また，WWWページの構成はぞれぞれのWWW情報源独自のものであり，１つのWWWページに様々な情報が混在する．他には，FORMの入力パラメータの仕様もそれぞれ独自のものであり，統一性がない．

半構造化情報: WWW情報源はHTMLを用いて情報を表現するが，HTMLは視覚構造を表現する言語であり，文書の意味的な構造を表現するのには適さない．そのため，機械的な解析によって必要な情報を取り出すことは困難である．

動的なWWW情報源: WWW情報源の発信する情報は，時間とともに内容や構造が更新される．

様々な統合要求: 情報統合の対象とするWWW情報源や，それらの組合せがそれぞれの利用者で異なる．

柔軟な情報統合を行うには，これらの問題に対処できるシステムが必要になる．

まず，WebL[2]やMetaCommnader[3]の様なスクリプト，あるいはJavaの様な言語による情報統合処理の記述を考えてみる．スクリプトはその自由な表現力で，ページより細かい単位での抽出が可能で，個々の利用者の要求に応えることができる．しかし，以下のような欠点を持つ．

- 利用者の要求が変化すると，スクリプト全体を修正する必要があり，柔軟性を持たない．
- 動的なWWW情報源が原因で，スクリプトの想定しているページ構造と実際の構造にずれが生じると，そのWWW情報源からの抽出ができなくなる．
- 利用者がスクリプトに関する知識を持たないと，抽出処理を記述できない．

このように，スクリプトによるWWW情報統合は柔軟性や頑健性に問題があり，他の手段を考える必要がある．

そこで我々は，図１に示されるエージェントベースのシステムを提案する．まず，情報統合の処理機能を独立性の高い機能に分解し，それぞれの機能を担当するエージェントを用意する．エージェントの交換を可能にするために，エージェントの入出力インタフェースを統一する．エージェントは独立性が

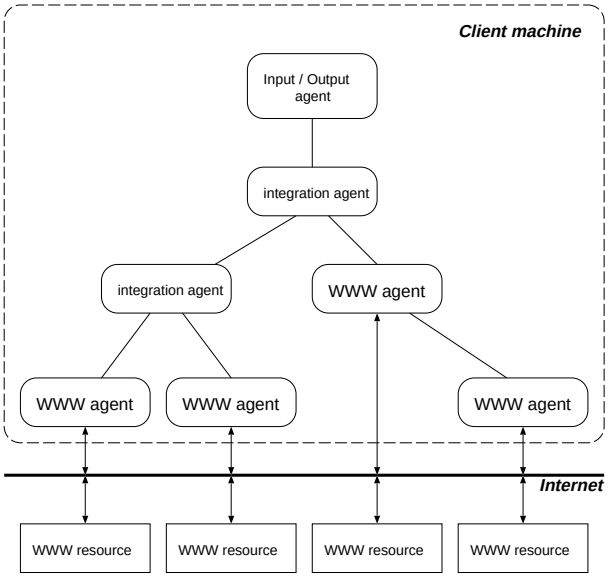

図 1: エージェントによる情報統合

高く，あるエージェントの交換は他のエージェントに影響しにくく，エージェントの再利用性が高まる．こうして，機能を分担したエージェントを自由に組み合わせることができ，様々な統合要求に応えられる．

次に，どのようなエージェントが必要かを考える．情報統合は，WWW 情報源からの情報抽出，利用者の指定した条件による複数の抽出情報の統合，統合結果のファイルへの保存の手順を組み合わせて実現される．そこで，統合処理機能を抽出，統合，保存の独立性の高い 3 つの機能に分解し，それぞれのエージェントを提供する．

WWW エージェント：HTTP を介して WWW 情報源にアクセスし，得られた WWW ページから必要な情報を抽出する．抽出結果は，意味情報を付加して構造化を行う．

統合エージェント：WWW エージェントの収集した複数の情報を，利用者の指定した条件で 1 つにまとめる統合を行う．

入出力エージェント：統合結果のファイルへの保存や，利用者の要求の送信などを行う．

WWW 情報源からの情報抽出を考えると，WWW 情報源は，先に述べたように自律分散的であり，それぞれの WWW 情報源は独立性が高い．本稿では，WWW 情報源からページより細かい単位で情報抽出を目的としているが，HTML 構造はそれぞれの WWW 情報源で異なり，機械的な抽出は困難である．よって，それぞれの WWW 情報源に対し，個別に WWW エージェントを提供するのが適当といえる．

```
<result>
  <hits>10</hits>
  <item>
    <title>大阪市立大学ホームページ</title>
    <url>http://www.osaka-cu.ac.jp</url>
    <score>94%</score>
    <size>4369</size>
    <update>98/11/18</update>
    <description>2月2日より大学の郵便番号が558-
     8585(杉本キャンパス), 545-...</descrition>
  </item>
  <!-- 2件目以降の記述 -->
</result>
```

図 2: メッセージの例

次に，エージェントの出力する情報の表現形式について考える．統合を行うには，出力されたデータがそれぞれどのような意味を持つかを指定する必要がある．例えば，あるエージェントが数字を出力しても，受信側ではそれが何を意味しているか (値段，重さ，得点など) を知ることはできない．そのため，データに意味情報を付加する必要がある．そこで，出力情報はその表現形式を XML とし，以降これをメッセージと呼ぶ．XML は要素をタグを用いて表す汎用的なデータ記述言語である [4]．メッセージの例を挙げると，典型的な検索エンジンを対象とする WWW エージェントでは，図 2 の様に検索エンジンの検索結果を構造化したメッセージを出力する．

メッセージの送信先 (あるいは受信元) のエージェントを特定するために，リンクを定義する．リンクは 2 つのエージェント間に接続され，メッセージは単方向のみに流れる．また，1 つのエージェントから複数のエージェントへリンクを接続することも可能で，このときメッセージは複製されて各々の出力先のエージェントへ送られる．エージェントはメッセージを受信すると，それを参照してそれぞれ固有の処理を行う．その結果，新たなメッセージが作成されると，新しいメッセージは更にリンクを介して別のエージェントへ送られる．このようにメッセージの送受信を繰り返して，統合処理を行う．

こうして，WWW 情報源に個別に対応した WWW エージェント，統合エージェント，入出力エー

ジェントから構成される，情報統合を行うシステムを提案する (図１)．また，エージェントとそれらを結ぶリンクの集まりを，統合ネットワークと呼ぶ．

3　情報抽出

3.1　抽出手段

ある WWW 情報源からの情報抽出の手段として，

- 正規表現によるパターンマッチング
- WWW ページを木構造とみなし，ツリーマッチングを行う
- 文脈自由文法
- スクリプト言語

などが考えられる．しかし，正規表現では表現不可能なデータ構造がある．例えば，$a^i b^i | (i > 1)$ の様に反復回数の記憶を必要とするデータ構造を表現することはできない．ツリーマッチングや文脈自由文法についても，その表現能力に限界があり，これらの問題点は [2] で指摘されている．WebL や MetaCommander といった，HTML 文書からの情報抽出を目的としたスクリプト言語が存在するように，WWW エージェントが情報抽出を行うには，スクリプト言語を使用するのが適当といえる．

3.2　抽出手順

WWW エージェントは，以下の手順で WWW 情報源から情報抽出を行う．

1. WWW 情報源にアクセスする．
2. 情報抽出を行う．
3. 抽出された情報の型を確認する．
4. 最終的に得られた結果を XML で表現し，メッセージとして出力する．

1.で，対象とする WWW 情報源に HTTP を介してアクセスし，WWW ページを得る．ところが，ネットワークの混雑や，サーバの混雑などで，WWW 情報源にアクセスできないときがある．このとき，WWW エージェントは，再度アクセスを試みる．もしミラーサーバがあれば，そちらのアクセスも試みる．

2.では，あらかじめスクリプトで記述された抽出手順に従って情報の抽出を試みる．しかし，WWW 情報源は時間とともにその内容や構造が変化する．そのため，WWW エージェントの抽出手順と実際の WWW ページの構造にずれが生じて，一部 (あるいはすべて) の情報が抽出できなくなることがある．

3.では 2.で抽出された情報の型を確認する．なぜなら，抽出で得られた結果文字列に意図しない文字列が含まれていたり，まったく別の部分を取り出してしまうことがあるからである．このような誤った抽出に対応するために，抽出情報の型を確認する．型には，文字列，符号つき整数，符号なし整数，符号つき小数，符号なし小数を用意し，抽出情報の型があらかじめ指定しておいた型に一致するかを確認する．もし型が一致しなければ，その情報は出力しない．

4.ではこうして得られた情報を XML に変換してメッセージを作成し，WWW エージェントの出力とする．

3.3　差分の利用

利用者の本システムの利用方法の１つに，固定された統合ネットワークで，定期的にその処理結果を確認する利用方法が考えられる．例えば，複数の新聞社の WWW ページから毎朝トップ記事を集める処理が挙げられる．このような場合，利用者は前回にアクセスしたときに得られた情報と，今回得られた情報で，異なっている部分 (更新された部分) に特に注目すると考えられる (以降，差分と呼ぶ)．しかし，大規模なデータベースで検索すると，大量の件数の結果を出力することがあり，人間がここから差分を見付けるのは時間がかかり，見落とす可能性も高い．そこで出力結果の内，差分のみを利用者に提示すると，利用者の時間を節約したり，手間を軽減できる．この処理は，利用者が差分の取得を希望する WWW エージェントで，先に述べた抽出手順の 3. と 4. の間で行われる．

4　統合操作

メッセージは，エージェント間での情報伝送に用いられる．メッセージは各種エージェントの出力する意味構造を保持するために，その形式を XML とする．検索エンジンを対象とする WWW エージェント

であれば，図2の様なメッセージを出力する．各種エージェントは受信したメッセージを参照したり，要素の追加や削除といった処理を行う必要があり，その表現手段としてSQLを用いることとする．

4.1 SQLによる記述

最初に特定の要素の指定方法について述べる．XMLは木構造をしていると見ると，タグ名を根から順に指定することで要素の特定が出来る．ここでは，その書式を$tag_1.tag_2$の様にする．図2の例でurlを指定するには$result.item.url$と記述し，10件の要素を得る．こうして特定の要素を指定できる．

次にSQLによる参照について下の簡単な例を挙げて述べる．

$$\begin{aligned} SELECT &\quad result.item.url, result.item.title \\ FROM &\quad AGENT \\ WHERE &\quad result.item.score \geq 75 \end{aligned}$$

FROM行ではメッセージを指定するために入力元となるエージェントを指定する．上の例では，エージェント$AGENT$から送られて来たメッセージを指す．次にWHERE以下の条件を満たす部分木を探す．該当する部分木の中で，SELECTで指定した要素を取り出す．

複数のメッセージをまとめる統合は，結合質問を行うことに相当する．2つの検索エンジンAとBで，両方の結果に含まれ，両方のスコアが50以上のWWWページのURLとページタイトルを取り出したい場合には，次のSQLで実現できる．

$$\begin{aligned} SELECT &\quad A.result.item.url, A.result.item.title \\ FROM &\quad A, B \\ WHERE &\quad A.result.item.url = B.reult.item.url \\ AND &\quad A.result.item.score \geq 50 \\ AND &\quad B.result.item.score \geq 50 \end{aligned}$$

また，入れ子質問の様なより複雑なクエリーや，更新(INSERT)や削除(DELETE)も同様に表現することが出来る．

4.2 データ照合に関する問題

複数のメッセージの統合を行うには，メッセージに含まれる特定の要素を指定するためにタグ名を参照したり，ある要素の持つデータを参照する必要がある．ここでは，複数のメッセージを統合するときに発生する2つの問題について述べる．

代表タグ	派生タグ
price	cost, fare, 価格, 値段 ...
university	univ, college, 大学, ...
car	motorcar, automobile, 自動車, ...

図 3: 同義タグリスト

4.2.1 タグ名の不一致

XMLはタグにより構造を管理し，タグ名ですべての要素を参照できる．ところで，後述する利用者によるWWWエージェントの開発が原因で問題が発生する．WWWエージェントの出力要素の名前は開発者が自由に決定するので，同じ意味のタグであってもWWWエージェントによりそのタグ名が異なる可能性がある．例えば，値段を表すタグに”price” ,”sales” ,” 値段” ,” 価格” といった名前を付加することは何ら不自然ではない．ところが，統合エージェントでタグを参照する際に，これらのタグは別のタグとして扱われ，利用者の意図する結果を得ることが出来なくなる．そこでタグ名が異なっていても対処できるように，同義タグリストを用意する (図3)．同義タグリストの各項は代表タグと，複数の派生タグから構成され，派生タグは代表タグと同じ名前とみなす．図３の例で，メッセージに$item.sales$という項目があれば，それは$item.price$と読み換える．同義タグリストによるタグ名の読み換えはSQLによる統合処理の直前に行う．またXMLでは，タグ名の大文字と小文字を区別するので，“price” と”Price” は区別される．これらも同一とみなすのが望ましいので，アルファベットは小文字に統一する．以上の結果，同じ意味のタグでその名前がWWWエージェント毎に異なっていても統合等の処理が可能になる．

4.2.2 略称等の問題

WWWエージェントの抽出する情報の書式は，抽出元のWWW情報源の表現方法に従い，XMLではすべてのデータを文字列として扱う．そのため同じ意味の文字列であっても，その表現が異なることが

ある．例えば，1000円を表す文字列に，”1000円”, ”1,000yen” ,”千円”などが考えられ，文字列として見るとこれらはすべて異なるが，数字(値段)として見ると同じ意味を持つ．他には，会社や大学名は正式名称の他に複数の略称があるが，これらを同一のものとして扱う必要がある．略称等の問題を解決する案の１つとして，同義タグリストと同様に代表名と，派生名から構成されるリストを用いる方法が考えられる．

5　エージェントの共有と再利用

あるWWW情報源から情報抽出をするには，その情報源に対応したWWWエージェントが必要である．逆に言うと，対応するWWWエージェントがなければ，その情報源は統合の対象にできない．本システムの様にWWW情報源に個別に対応するシステムでは，いかに多くのWWW情報源を統合の対象にできるかが，重要な課題の１つと言える．そこで，WWWエージェントの利用者による開発と，それらの共有を提案する．

利用者が自由にWWWエージェントを開発できると，統合可能な範囲が拡大し，利用者に利便性をもたらす．しかし，利用者がWWWエージェントを開発できても，個人の力では開発できるWWWエージェントには限りがある．そこで，個々の利用者が開発したWWWエージェントを，すべての利用者の間で共有すると，統合可能なWWW情報源は個人のときのそれと比べて増加する．

6　まとめ

クライアントマシンでWWW情報統合を行うアーキテクチャについて述べた．WWW情報統合に必要な機能をエージェントに分解し，それらを組み合わせることでWWW情報統合を行うことができる．またエージェントは再利用可能であり，エージェントの組合せを変えたり，参照する入力パラメータを変化させることで，様々な統合要求に応えることが出来る．利用者によるWWWエージェントの開発とそれらの共有は，統合可能な範囲を拡大させる．

今後の課題としては，WWWエージェント開発支援が挙げられる．本システムでは，スクリプトを用いて抽出処理を記述するが，計算機の非専門家にとって，スクリプトの記述はやさしいとはいえない．そのため，WWWエージェントを開発できる人間が限られてしまう．そこで，スクリプトの知識を持たなくても，WWWエージェントを開発できる環境が必要になる．[5]では利用者がシステムにどの情報を抽出するかを例示することで，抽出の記述を試みている．この方法は”Demonstration-oriented User Intersace”(DoUI)と呼ばれている．

また派生タグリストによって，タグ名の読み換えを行っているが，すべての組合せを羅列するのは無理がある．この対策としては，類義語辞書を用いたり，２つの文字列が同じ意味を持つかを推測したりする機能が考えられる．

謝辞

本研究の一部は，文部省科学研究補助金特定領域研究A(1)「ゲノムサイエンス」(課題番号08283103)によるものである．

参考文献

[1] Robert B. Doorenbos, Oren Etzioni and Daniel S. Weld, A Scalable Comparison-Shopping Agent for the World-Wide Web, Proc. 1st International Conference on Autonomous Agents,39-48(1997).

[2] Thomas Kistler, Hannes Marais, WebL - a programming language for the Web, Proc. of 7th WWW Conference(1998).

[3] 北村泰彦，野崎哲也，辰巳昭治，スクリプトに基づくWWW情報統合支援システムとゲノムデータベースへの応用，電子情報通信学会論文誌，J81-D-I(5):451-459(1998).

[4] XML/SGMLサロン.標準XML完全解説,技術評論社(1998).

[5] Craig A. Knoblock, Steven Minton, Jose Luis Ambite etc, Modeling Web Sources for Information Integration, Proc. 15th National Conference on Artificial Intelligence, 211-218(1998).