社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－DBS－143（89）
2007／7／4

非定型Webコンテンツ管理のための軽量ラッピング言語

澤 菜津美† 森嶋 厚行† 杉本 重雄† 北川 博之††

† 筑波大学大学院図書館情報メディア研究科 〒305-8550 茨城県つくば市春日1-2
†† 筑波大学大学院システム情報工学研究科 〒305-8573 茨城県つくば市天王台1-1-1
E-mail: †{sawa,mori,sugimoto}@slis.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

あらまし 本稿では，HTMLで記述されたWebコンテンツから構造データを抽出するためのラッピング言語Parseletを提案する．Parseletは，特に非定型Webコンテンツから，構造データを抽出する事を考慮して設計されたものである．そのため，人手による規則の記述が容易になるよう，簡易な構文やライブラリの工夫を行っている．本稿では，Parselet開発の動機と設計について述べ，実Webサイトへの適用可能性に関する予備実験の結果を示す．
キーワード Webサイト管理，情報統合，ラッピング言語

A Lightweight Wrapping Language for the Management of Non-Template-Based Web Contents

Natsumi SAWA†, Atsuyuki MORISHIMA†, Shigeo SUGIMOTO†, and Hiroyuki KITAGAWA††

† Grad. Sch. of Info. and Media Studies, Univ. of Tsukuba. Kasuga 1-2, Tsukuba, Ibaraki, 305-8550 Japan
†† Grad. Sch. of Sys. and Info. Eng., Univ. of Tsukuba. Tennohdai 1-1-1, Tsukuba, Ibaraki, 305-8573 Japan
E-mail: †{sawa,mori,sugimoto}@slis.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

Abstract This paper proposes Parselet, a wrapping language for extracting structured data from Web contents written in HTML. Parselet is designed especially for extracting structured data from non-template-based Web pages and for maintaining the content integrities among such Web pages. Its simple syntax and the library of useful patterns help the user write wrapping descriptions by hand. This paper explains the motivation of its development and the language design and then shows the result of a preliminary experiment about applicability of the language to real Web sites.
Key words Web-site Management, Information Integration, Wrapping Languages

1. はじめに

本稿では，HTMLで表現されたWebコンテンツから構造データを抽出するためのラッピング言語Parseletを提案する．既に，Webコンテンツをラッピングするための仕組みは数多く研究されてきた[1][2]．これらは，主に，複数のWebコンテンツの統合利用や，Webコンテンツに対してDBライクな問合せを実行することを念頭に研究が進められてきたものである．それに対し，Parseletは，特に非定型Webコンテンツの一貫性管理に応用することを念頭に設計されているため，これらとはやや異なる特徴を持っている．Parseletの特徴をまとめると次のようになる．(1) 簡易な構文やライブラリなどの工夫により，人手でラッピングのための記述を書き下ろすことが比較的容易である．(2) HTMLの中に組み込んで利用できる．(3) HTMLデータの論理構造を考慮したパースが可能である (詳細は4章で説明する)

本稿では，Parselet開発の動機，その設計，実用性検証のための予備実験の結果について述べる．構成は次の通りである．まず，2章で，Webサイト構築方式とWebコンテンツの一貫性管理の問題について議論する．3章では，この問題に関して我々が進めているプロジェクトと，提案するParseletの位置づけについて述べる．4章で，Parseletの設計について説明する．5章では，Parselet評価のための実験について述べる．6章では，関連研究について述べる．7章はまとめと今後の課題である．

2. Webサイト構築方式とWebコンテンツの一貫性管理の現状

現在，Webサイトを構築する方法には，大きく分けて次の二つの手法がある．

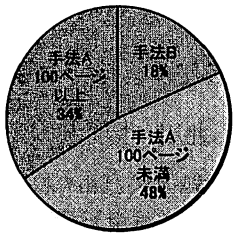

図 1　tsukuba.ac.jp 内の Web サイト調査結果
(ページ収集期間は 2006 年 12 月 22-25 日)

(手法 A) 各ページのコンテンツの直接作成: 例えば，テキストエディタを用いて HTML ドキュメントを直接作成する方法，HTML 作成支援ツール等を用いる方法，Wiki などを通じて作成する方法，などがある．コンテンツの更新は各ページを更新することにより行われる．

(手法 B) ページとは別の情報源からページを作成するシステムを構築: 例えば，バックエンドに DB システムを配置し，DB に格納されているデータから Web ページを作成する方法．コンテンツの更新は DB の更新により行われる．

一般に，Web サイトに含まれるコンテンツは互いに関連していることが多い．例えば，大学の研究室の Web サイトでは，各人のページに研究室の名前，住所，電話番号が含まれており，これらは一致するはずである．また，研究室メンバの発表論文の一覧は，研究室の発表論文一覧のサブセットであることが一般的である．このような関連を表す制約を，本論文では**コンテンツ一貫性制約**と呼ぶ．Web サイトのコンテンツの変更があった場合には，これらのコンテンツ一貫性制約が保持されるように更新されることが望ましい．しかし，Web サイトの規模が大きくなるにつれ，手法 A ではコンテンツ一貫性の維持が困難になる．したがって，ある程度大規模な Web サイトは手法 B で構築される．

予備調査として，我々は tsukuba.ac.jp 内の Web サイトの調査を行った (図 1) [3]．これは，クローラを用いて収集した tsukuba.ac.jp 内のサイトから無作為に選んだ 300 個の Web サイトを対象としたものである．その結果，82%のサイトが手法 A で構築された Web サイトと考えられる物であった．また，それらのうち 34%が 100 ページ以上の Web ページを持っていた．以上の結果から，ある程度多くの Web ページを持つ Web サイトであっても，手法 A で構築されている Web サイトが多いことが強く推測される．このような状況である原因はいくつか考えられる．例えば，(1) 手法 B の Web サイトを構築するためのリソースが存在しない，(2) サイトの内容が非定型であり，手法 B に適さない，(3) 最初は少ないページであったので手法 A で構築していたが，いつの間にか規模が大きくなった，等である．手法 A で作成された Web サイトのコンテンツの一貫性を保持するためには，各ページの入念なチェックとページ毎の更新が必要であるが，多数の Web ページについて行おうとすると，非常に労力がかかることは明白である．

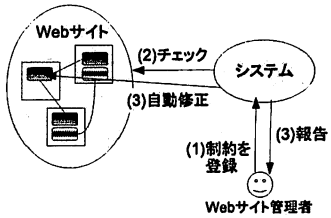

図 2　コンテンツ一貫性制約を用いた Web サイト管理手法

3. 明示的なコンテンツ一貫性制約を用いた Web コンテンツ管理

そこで我々は，明示的なコンテンツ一貫性制約を用いた Web サイト管理手法を提案している [3]．図 2 はコンテンツ一貫性制約を用いた Web サイト管理の仕組みを表したものである．以下にその手順を述べる．まず，利用者がコンテンツ一貫性制約を登録する (図 2 (1))．登録の際には利用者が直接コンテンツ一貫性制約を作成しても良いが，既存のコンテンツからコンテンツ一貫性制約の候補を自動発見させて，適切と考えられるものを一部採用しても良い [3]．制約が登録されると，システムは定期的もしくは Web サイトの更新が行われた際などに Web サイトのチェックを行い，先に発見しておいた制約と照らし合わせて，制約が破られていないかどうか調べる (図 2 (2))．その際，もし制約違反を発見したら，Web サイト管理者に報告もしくは自動修正を行う (図 2 (3))．

本論文での問題. 提案管理手法を実現するためには，コンテンツ一貫性制約を記述する必要がある．コンテンツ一貫性制約に関する議論の一部は [3] にあり，本稿では省略するが，効果的に制約を記述するためには，HTML データの論理構造を適切に把握できなくてはならない．その鍵となる技術が，HTML データからの構造データの抽出である．一般に，HTML データはブラウザから見たときの見やすさなどを優先して記述されているためそこに含まれているデータ間の関係が明示的に現れておらず，それらの関係を入手するためには何らかの仕組みを用意する必要がある．本稿では，HTML データから構造データを抽出するための仕組みとして Parselet を提案する．

3.1 Parselet 利用のシナリオ例

コンテンツ一貫性制約を用いた Web サイト管理手法のシナリオ例について述べる．

(シナリオ 1) コンテンツ一貫性制約違反の警告: ある研究室の Web サイトでは，図 3 のような論文リストを教員 A および学生 C,D がそれぞれ自分の Web ページに掲載している (図 4)．各人のページのコンテンツに関しては，次のような制約がある:

制約 1: $\forall s \in$ 学生 (教員 A の論文集合$\supseteq$学生 s の論文集合)

各 Web ページを個人が管理していると，各学生が論文リストを更新したにも関わらず，教員 A のページはなかなか更新されない，といった状況が起こりうる．このため，上記制約が満たされない状態が生じる．このような場合，コンテンツ一貫性制約を用いた，ページ間の制約の指定およびチェックが有効である．上記制約を，満たすべきコンテンツ一貫性制約として指

図 3　論文リストの Web ページの例

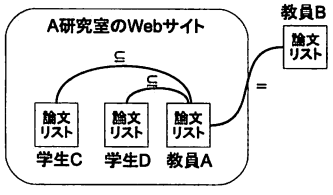

図 4　コンテンツ一貫性維持の例

```
<li>
飯田敏成，澤菜津美，森嶋厚行，杉本重雄，北川博之
『WWW のリンク切れで困っていませんか? -The WISH Project-』
電子情報通信学会第 17 回データ工学ワークショップ (DEWS2006)，沖縄コンベンション
センター，2006 年 3 月.
</li>
```

図 5　論文リストの例

```
<papers>
  <paper>
    <authors>
      <auth>飯田敏成</auth>
      <auth>澤菜津美</auth>
      <auth>森嶋厚行</auth>
      <auth>杉本重雄</auth>
      <auth>北川博之</auth>
    </authors>
    <title>WWW のリンク切れで困っていませんか? -The WISH Project-</title>
    <info>
      <_undef>電子情報通信学会第 17 回データ工学ワークショップ
              (DEWS2006)</_undef>
      <_undef>沖縄コンベンションセンター</_undef>
      <_undef>2006 年 3 月</_undef>
    </info>
  </paper>
</papers>
```

図 6　パース結果

定すると，システムは自動的に制約違反を発見する．制約違反になった際には，教員 A に，メールで報告するようにしておけば，教員 A は学生の論文が更新された事にすぐに気付いて修正できるので，同一 Web サイト内でのコンテンツ一貫性維持に役立つ．

問題は，HTML データから「教員の論文集合」を適切に同定する事が困難であるため，そのままでは，これらのコンテンツ一貫性制約が成立しているかどうかの判定が自明でないことである．Parselet は，このコンテンツ一貫性制約の利用を効果的にするための鍵となる．Parselet を利用すれば，例えば図 5 の HTML データから図 6 のような XML で表現された構造データを出力できる．

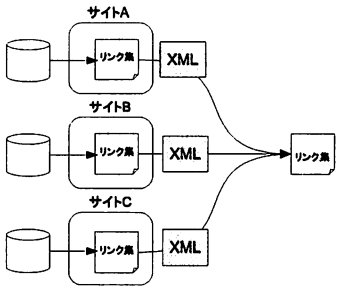

図 7　動的ページ生成の例

(シナリオ 2) 既存 Web コンテンツからの動的なページ生成: シナリオ 1 と異なり，本シナリオでは，制約のチェックではなく，既存の Web コンテンツから別の Web ページのコンテンツを作成する (図 7)．Web ページ A,B,C が，Web コンテンツとしてリンク集をそれぞれ持つとする．各リンク集に対して，Parselet を用いて XML 形式のデータを取得し，XQuery などを使ってこれらをまとめた一つのリンク集を生成する．

以上 2 つのシナリオにおいては，Web コンテンツを管理するサイトが 1 つであったり，Web サイト管理権限が必要であるといった制限がないことに注意して欲しい．例えば，教員 A の論文リストと他大学の教員 B の論文リスト間に，次のような制約が存在するシナリオも考えられる．これは，シナリオ 1 に似ているが，複数サイトにまたがった例である．

制約 2: 教員 A のページに存在する教員 B との共著論文集合= 教員 B のページに存在する教員 A との共著論文集合.

4. Parselet

本章では Parselet について説明する．Parselet は，HTML データからどのように構造データを抽出するかを記述するための言語である．Parselet を用いて構造データを抽出するためには，Parselet で記述された式 (Parselet 式) と HTML データを入力として，構造データの抽出を行うソフトウェア (パーサ) を利用する (図 8)．パーサは，構造データを XML の形式で出力する．以降では，出力する XML データを木の用語を用いて表現することがある．例えば要素を**ノード**とよび，要素の入れ子関係を親ノード，子ノードなどで表現することがある．また，要素名をノードの**ラベル**と呼び，要素が直接含む CDATA をその要素 (ノード) の**値**と呼ぶ．

Parselet の特徴は，次の通りである．(1) 簡易な構文やライブラリなどの工夫により，人手でラッピングのための記述を書き下ろすことが比較的容易である．(2) HTML データの中に組み込んで利用できる．具体的には，`<ul parselet=Parselet 式>`というように，タグ内に組み込んでおくと，そのタグの範囲を対象として Parselet 式を指定したことになる．この機能により，Web ページの作成者自身が，構造データの抽出方法をあらかじめ内部に埋め込んでおくことが出来る．(3) HTML データの論理構造を考慮したパースが可能である．これに関しては後述する．

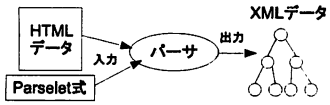

図 8　パースの流れ

(a) 果物在庫リスト

```
<ul>
  <li>りんご,10</li>
  <li>みかん,20</li>
  <li>桃,30</li>
</ul>
```

(b) Parselet 式でパースした結果

```
<在庫>
    <果物><名前>りんご</名前><数量>10</数量></果物>
    <果物><名前>みかん</名前><数量>20</数量></果物>
    <果物><名前>桃</名前><数量>30</数量></果物>
</在庫>
```

図 9　Parselet 式の適用例

簡単な例. 図 9(a) のような果物在庫を表現する HTML ページがあり，それに対応する構造データが図 9(b) であるとする．このとき，その構造データを出力するための Parselet 式は次のようになる．

```
在庫:/{ 果物:#li/[名前:_val(#v)\,, 数量:_val(#v)]}
```

Parselet 式は，“ラベル:パターン/子ノードのための (一般には複数の)Parselet 式” という入れ子構造で表現される．ラベルは出力する構造データのノードラベルを指示する．例の場合では，在庫，果物，名前，数量がラベルである．パターンは，そのノードに対応する HTML データの部分を指定するのに使われる．これは，文字の正規表現や，あらかじめライブラリとして用意しているパターン部品で指定される．パターン部品は，よく使うパターンや複雑なパターンに名前を付けたものであり，「#パターン部品名」と表す．例の場合では，`#li` と `_val(#v)\,` と `_val(#v)` がパターンであり，そのうち`#li` と `#v` がパターン部品である．`#li` は `li` 要素内の文字列にマッチし，`#v` はパターンにおける前後の文字を含まない文字列にマッチする．また，パターンは，後述する特殊指示子を含むことが出来る．上の例では`_val()` が特殊指示子である．これは，パターンのその部分にマッチした文字列を該当ノードの値とする指示子である．在庫の子ノードの式は {...} でくくられているが，後述するようにこれは繰返し構造を表す．

Parselet パーサの動作. パーサは，Parselet 式が与えられると，次のように動作する．すなわち，入れ子構造の外側からパターンマッチを行い，マッチする部位を発見する毎に XML 木のノードを生成する．さらに，マッチした部位を対象として，子ノードのための Parselet 式を評価する．ただし，ノードにパターンが指定されていない場合には，無条件にその該当ノードを生成する．先の例の場合は，次のように動作する．

$$E \rightarrow [label]\ ':'\ [pattern]\ ['/'\ C]$$
$$C \rightarrow '\{'\ E\ '\}'\ [\ '('\ pattern\ ')'\ ]\ |\ '['\ E\ \{\ ','\ E\ \}\ ']'$$

図 10　Parselet の構文

`_val(p)`	p の部分を値にする
`_before(p)`	p にマッチした文字列の先頭位置を，次に実行するパターンマッチの開始位置とする．
`_skip(p)`	p にマッチした文字列の次の文字を，次に実行するパターンマッチの開始位置とする．

図 11　特殊指示子

（1） 在庫:にはパターンが存在しないため，無条件にルートノードである在庫ノードを作成する．

（2） 図 9(a) の果物在庫リストに対して，パターン部品`#li` にマッチする文字列を順に抽出する．図 9(a) の場合，最初にマッチする文字列は，「りんご,10」である．パターンマッチに成功すると，果物ノードが生成される．このパターンには特殊指示子`_val()` が存在しないため，値は生成されない．

（3） 果物ノードが作成されると，マッチしたパターンそれぞれに対して子の Parselet 式が評価される．最初の果物ノードでは「りんご,10」が対象になる．名前ノードには，「りんご」が，数量ノードには「,10」が，それぞれパターンマッチし，これらのノードが作成される．どちらにも`_val()` が存在するため，「りんご」と「10」がそれぞれの値となる．

Parselet の構文. Parselet の構文は図 10 のようになる．ラベルは省略可能であるが，その場合は，ラベル_undef が存在するとみなされる．子ノードのための Parselet 式の書き方には，列，繰り返し，の 2 種類がある．[···] は列，{···} は繰り返しである．繰返しには，while 条件をつけることが出来る．これは，{···}(繰り返し条件) のように記述する．

特殊指示子. パターンには，図 11 の特殊指示子を挿入することが出来る．`_val(p)` については既に説明した．`_before(p)` は，主に繰返し構造と一緒に利用される．例えば，論文の書誌情報から著者とタイトルを抽出したいとき，`[{auth:...}(_before("))`, `title:" ... "]` などというパターンが用いられる．この場合，「"」にたどり着くまで著者情報を抽出するが，次のタイトルの情報を抽出するためには改めてその場所 (「"」) からパターンマッチが行われる．それと異なり，`_skip(p)` は，パターンマッチした文字列の次の文字から，次のパターンマッチを適用する．これはデフォルトの解釈であるため，通常は指定しない．

パターン部品ライブラリ. 現在検討しているパターン部品ライブラリの一部を図 12 に示す．`#name` は，人名にマッチするためのパターン部品である．例えば，”Sawa, N.”など，氏名の間にカンマ等が入っている場合でも，適切に氏名を取得する．

Parselet の特徴の一つは，必ずしも HTML 要素の並び順に依存しないパターン部品を用意していることである．表のパースを例に説明する．HTML では，表は行の並びとしてエンコードされているが，`#column` を利用することにより，列を単位と

パターン名	機能
#li	... にマッチする.
#name	氏名にマッチする.
#v	直後に指定されたパターンを含まない文字列にマッチする.
#row	テーブル行にマッチする.
#column	テーブル列にマッチする.
#combination	2 次元の表を 1 次元化する.

図 12 パターン部品ライブラリの一部

(a) リーグ表

	A	B
A	-	2-0
B	0-2	-

(b)1 次元化したデータ

```
<group>
  <game><t>A</t><t>A</t><r>-</r></game>
  <game><t>A</t><t>B</t><r>2-0</r></game>
  <game><t>B</t><t>A</t><r>0-2</r></game>
  <game><t>B</t><t>B</t><r>-</r></game>
</group>
```

図 13 #combination の使用例

```
papers:/{paper:#li/[authors:/ [auth:_val(#name),
{auth:,_val(#name)}(_before(『))], title:『_val(#v)』,
info:/[{:_val(#v),}, :_val(#v)\.] ]}
```

図 14 Parselet 式

したパースが行われる．また，#combination は，2 次元の表を 1 次元化してパースする．例えば，図 13(a) のリーグ表の結果に使用すると，同図 (b) のように，1 次元化されたデータを抽出することができる．

4.1 記 述 例

図 5 の HTML データから図 6 の XML データを抽出するための Parselet 式を図 14 に示す．このとき，パーサの動作は次のようになる．まず，ルートノードとして，papers を作成する．その子供として，paper ノードを複数作成する．paper ノードのパターンは，#li なので，li タグ内の文字列にマッチする．次に，paper の子供として，authors, title, info を作成する．authors の子供には，複数の auth を作成する．最初の auth は #name にマッチする文字列である．その後の auth は,「,」をスキップし，#name にマッチする文字列を値とする事を繰り返す．この繰返しを「『」にマッチするまで行う．title は,『』内の文字列を値とする．info の子供には，ラベルの指定がないため，_undef が作成される．_undef は「,」の手前にマッチした文字列を値とし,「,」をスキップする事を繰返す．最後の_undef は,「.」の手前の文字列にマッチした文字列を値とする．

5. 予 備 実 験

Parselet の適用可能性を評価する予備実験として，Web 上の論文リストを対象として，Parselet により構造データの抽出が可能かどうかの実験を行った．

実験方法. Web に存在する，日本のデータベースシステム研究の領域における論文リストから無作為に 10 ページ選択し，それぞれのページからさらに無作為に選択した論文のデータに対して Parselet による論文リストの抽出を試みた．Web からの論文データの選択は下記のように行った．

（1） データベース関連研究機関のリンク集(注1)から無作為に 10 個の大学研究室の URL を選ぶ．

（2） それぞれの研究室 Web サイトの論文リストを掲載した Web ページを選択する．無い場合は，教員の論文リストのページを選択する (もし複数の教員がいる場合には，無作為に一つ選択)．

（3） 選択したページから無作為に 10 個の論文データを選ぶ (10 個未満の場合は，全ての論文データ)．

選択した論文数は合計 90 である．これらの各論文データに対して，図 6 のように，タイトル，著者名，その他の情報 (雑誌名，ページ数など) にパースするための Parselet 式を記述可能かどうか調査した．

実験結果. 90 論文のうち，Parselet によるパースが可能な論文リストは，85 個，Parselet によるパースができない論文リストは 5 個であった．Web ページ毎に分類した結果を，図 15 に示す．

ページ ID	1	2	3	4	5	6	7	8	9	10	合計
パース可	10	10	7	8	10	5	10	9	7	9	85
パース不可	0	0	3	0	0	0	0	1	0	1	5
合計	10	10	10	8	10	5	10	10	7	10	90

図 15 調 査 結 果

Parselet によるパースができない論文リストとは，正規表現と現在用意しているパターンライブラリの組み合わせだけでは，パースできない論文リストである．例えば，図 16 の論文リストをパースする際，次のような問題があった．

- タイトルと著者名は,「:」記号で区切られているので，タイトルのパターンを，_val(#v):と書くことが考えられる．しかし，これではタイトル中に「:」記号が含まれているため，正しくタイトルを抽出できない．逆にこの記号を含んでよいことにすると，終了条件を指定できない．論文タイトルに含まれる「:」の個数に関する一般的な規則はない．
- 著者名とその他の情報は，どちらも「:」記号で区切られた文字列の繰返しで表されているため，著者名の終了位置がわからない．

考察. このように，単純なパターンマッチだけでは難しい場合，現在の Parselet の枠組みおよびパターン部品ライブラリでは対応できない．これらに対処するためには，辞書を用意してパターン部品ライブラリに組み込むことが考えられる．例えば，人名辞書を用いてパターン部品#name を設定し，人名か否かを判定することが出来るようになると，タイトルを抽出するためのパターンは_val(#v)_before(#name) と書く事ができ，名前の手前までがタイトルであると指定することが出来る．

また，今回の実験では，論文リストを，タイトル，著者名，その他の情報の 3 つのデータ構造に分解した．もし，今回は一律にその他の情報としたものを，日付やページ数など細かくタ

(注1)：http://alpha.c.oka-pu.ac.jp/ yokota/db/db-dorg.html

<td>The WISH Project: Web Integrity management by Self-Healing mechanisms:Atsuyuki Morishima, Akiyoshi Nakamizo, Toshinari Iida, Tomohiro Ariyama, Shigeo Sugimoto, Hiroyuki Kitagawa, University of Tsukuba, (2006)</td>

図 16 論文データ例

グ付けしたい場合，Parselet 式が長くなり，またパターンが複雑になることによって人手での記述が大変になる．この問題については，ライブラリに含まれるパターン部品を充実させることによって，Parselet 式中で書かなければならないパターンの記述を単純化することが有効であると考えられる．

6. 関連研究

既に，Web コンテンツをラッピングするための仕組みは数多く研究されてきた．XWRAP [1] は，HTML データから構造データを抽出するラッピング記述を，ユーザとの対話により半自動生成する．XWRAP は抽出規則の記述が，Parselet に比べると複雑で長くなるため，ユーザが直接記述することは難しい．それに対して，Parselet はライブラリの充実や簡易な文法など，人手による直接記述を意識した設計になっていることや，HTML データに簡単に組み込むことが出来ることから，より非定型 Web コンテンツ管理に向いていると考えられる．Arasu らの論文 [2] では，DB をバックエンドにした Web サイトの定型コンテンツから，テンプレートとデータを分離する手法を提案している．具体的には，複数の定型ページをサンプルとして比較を行うことにより，ページ生成に使われたテンプレートを推測し，データだけを抽出するものである．この手法は定型のページを多量に持つような Web サイトからの構造抽出には向いているが，我々の想定する応用である非定型 Web コンテンツ管理には向いていない．

GRDDL [4] や microformats [5] は，どちらもセマンティック Web 実現の支援を目的に開発された技術であり，HTML データにセマンティクスを埋め込むための記法を提案している．これらは，重要な値に明示的にタグ付けを行うことによって意味を明確にする，というアプローチをとる．したがって，構文解析的な側面は持たず，個々のインスタンスレベルで意味の指定を行うことになる．それに対し，Parselet は構文解析のためのヒントを与えるというアプローチであるため，半構造的な性質を持つページ (論文リストなど) へのセマンティクスの付加を簡潔に行えるという利点がある．

7. まとめ

本稿では，HTML で記述された Web コンテンツから構造データを抽出するためのラッピング言語 Parselet を提案した．Parselet は，特に非定型 Web コンテンツから，構造データを抽出する事を考慮して設計されたものである．Parselet の特徴は次の通りである．(1) 簡易な構文やライブラリなどの工夫により，人手でラッピングのための記述を書き下ろすことが比較的容易，(2) HTML の中に組み込んで利用可能，(3) HTML データの論理構造を考慮したパースが可能．本稿では Parselet の設計と，適用可能性調査のための簡単な予備実験を行った．今後の課題としては，適用可能性と記述容易性をより高めるためのパターン部品の開発や，非定型 Web コンテンツの一貫性管理への Parselet の効果的な応用手法の開発などがある．

謝辞

Parselet の表現力に関して議論をいただきました筑波大学大学院図書館情報メディア研究科の中井央准教授に御礼申し上げます．また，ゼミなどでコメントいただきました筑波大学大学院図書館情報メディア研究科の阪口哲男准教授，永森光晴講師に感謝致します．本研究の一部は科学研究費補助金特定領域研究（#19024006）による．

文献

[1] L. Liu, C. Pu, and W. Han. XWRAP: *An XML-enabled* wrapper construction system for web information sources. International Conference on Data Engineering (ICDE), pp. 611-621, 2000.

[2] Arvind Arasu, Hector Garcia-Molina. Extracting Structured Data from Web Pages. ACM SIGMOD International Conference on Management of Data, pp.337-348, 2003.

[3] 澤菜津美, 森嶋厚行, 飯田敏成, 杉本重雄, 北川博之, コンテンツ一貫性制約を用いた Web サイト管理手法の提案. 電子情報通信学会第 18 回データ工学ワークショップ (DEWS2006), 7 pages, 2007 年 3 月.

[4] GRDDL. http://www.w3.org/TR/grddl/

[5] microformats. http://microformats.org/