Web コンテンツを対象としたラッピング言語の設計と利用に関する研究*

石井悠太 (学籍番号 200921725)
研究指導教員: 森嶋厚行
副研究指導教員: 鈴木伸崇

1. はじめに

近年，Web コンテンツを通じた情報発信が広く普及したことに伴い，大量の Web コンテンツの管理や再利用が重要な問題となっている．これまで，Web コンテンツの効率的な再利用を実現するための方法のひとつとして，ラッピング技術に関する様々な研究が行われてきた．ラッピング技術とは，ある形式のデータから他の形式のデータへの変換処理を行うための技術であり，ラッピングを行うソフトウェアはラッパーと呼ばれる．

ラッパーを構築するアプローチの一つに，ラッピング言語を用いてラッパーを記述するというアプローチ [1, 2] がある．ラッピング言語とは，ラッパーの処理を具体的に記述したラッピング規則を記述するための専用の言語である．

本研究では，既存のラッピング言語ではサポートされていなかった (1) **ラッピング規則の逆写像**，(2) **抽象的なラッピング規則**，という 2 つの機能を実現するラッピング言語 iWraplet および Wraplet/A を提案する．これらの言語は，澤らの提案したラッピング言語 Wraplet[1] をベースとして設計した．

また，本研究では，新たな機能を実現した 2 つのラッピング言語について，機能を実現するための構文および処理アルゴリズムの検討を行い，実際の Web コンテンツを対象として評価を行った．

2. ラッピング規則の逆写像の実現

第一のラッピング言語は，ラッピング規則の逆写像を実現したラッピング言語 iWraplet である．

ラッパーによるデータ変換は，異なるデータ間の写像としてモデル化できるが，既存のラッピング言語では一方向の写像の実現だけに焦点が当てられており，その逆写像については実現されていなかった．

iWraplet は，HTML 形式のデータ h から XML 形式のデータ x への写像を与えるラッピング規則 $f : H \to X$ を記述すると，その逆写像 $f^{-1} : X \to H$(XML 形式のデータから HTML 形式のデータの復元) を自動的に計算できるように設計を行った．

ラッピング言語の逆写像の応用としては，HTML

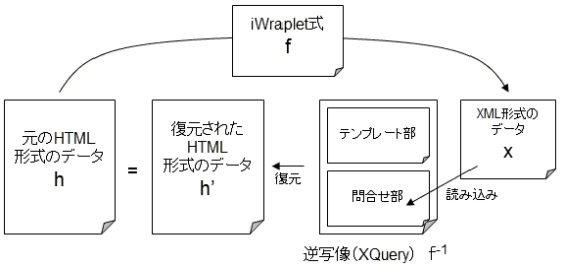


図 1　iWraplet を用いたラッピングおよび復元

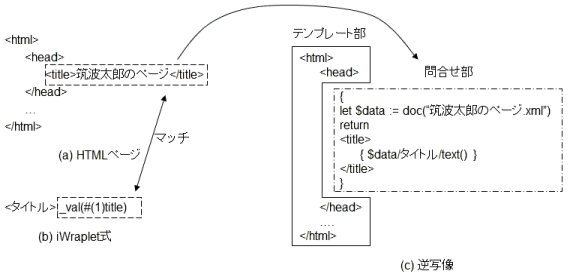


図 2　逆写像の生成

形式のデータだけからなる静的な Web サイトから，バックエンドの DB から Web コンテンツの生成を行う動的な Web サイトへの再構築などがある．

図 1 に，iWraplet を用いて記述したラッピング規則 (iWraplet 式) を用いたラッピングおよび逆写像 (本研究では XQuery を用いて逆写像を記述) によるデータの復元のイメージを示す．

逆写像は**テンプレート部**および**問合せ部**から構成される．テンプレート部は元の HTML 形式のデータから XML 形式のデータの値として抽出した内容を除いた，HTML 形式のデータである．問合せ部はラッピング結果の XML 形式のデータを読み込んで，元の HTML 形式のデータのタグを付与する．

逆写像を求める基本的なアイデアを図 2 に示す．逆写像の生成は，iWraplet 式の入れ子構造の外側から順に HTML 形式のデータとのパターンマッチを行い，マッチしたパターンに応じて問合せ部と置換する事を再帰的に繰り返すことによって行われる．

iWraplet では，逆写像の自動生成を実現するために，Wraplet に対して次の 2 つの制限を課した．(1) XML 形式のデータの要素の値は元の HTML 形式のデータのいずれかの要素の連続した部分文字列でなくてはならない，(2) 元の HTML 形式データに現れる同一の文字列が XML データに複数現れては

* “A Study on Design and Utilization of Wrapping Languages for Web Contents” by Yuuta ISHII

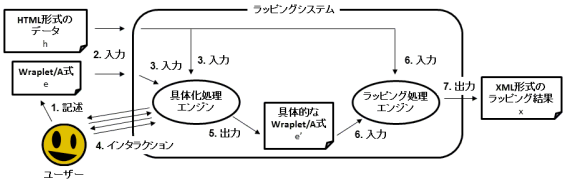


図 3　Wraplet/A を用いたラッピングの手順

ならない．これらの制限により，逆写像の自動計算が可能になり，iWraplet の表現力の範囲内で，低コストでの Web サイトの再構築などが可能となる．

評価．本研究室に所属する学生 3 名の Web ページを対象として，iWraplet を用いた Web サイトの再構築にかかるコストを評価した．評価の結果，DB に格納するデータの粒度に応じて 3 行から 11 行の iWraplet 式を記述することで Web サイトの再構築が可能であるという結果が得られた．したがって，提案言語を用いれば Web サイトの再構築コストを大幅に削減できる可能性があると推測できる．

3.　抽象的なラッピング規則の実現

第二のラッピング言語は，抽象的なラッピング規則を実現したラッピング言語 Wraplet/A である．

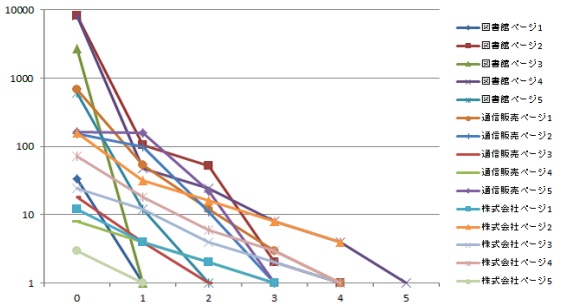
既存のラッピング言語では，完全に具体的なラッピング規則しか書けないため，次のような問題が生じていた．すなわち，(1) ユーザーは正規表現や対象データの構造の把握などに関する完全な知識が必要，(2) 記述したラッピング規則が対象データの詳細な構造 (タグの入れ子構造や区切り文字等) に依存しているため，記述したラッピング規則の再利用が困難．という問題である．

Wraplet/A は，具体的なラッピング規則だけでなく，詳細な構造を省略した抽象的なラッピング規則を記述することができるように設計を行った．Wraplet/A を用いて記述したラッピング規則を Wraplet/A 式と呼ぶ．また，具体的なラッピング規則のみからなる Wraplet/A 式を**具体的な Wraplet/A 式**と呼び，具体的なラッピング規則と抽象的なラッピング規則の混在した Wraplet/A 式を**抽象的な Wraplet/A 式**と呼ぶ．

Wraplet/A を用いることで，抽出対象のデータの**出現回数**，**出現順序**，**区切り文字列**の 3 点に関して省略した抽象的なラッピング規則が記述可能である．これらの抽象的な規則が記述できることにより，上記で挙げた，既存のラッピング言語の問題を緩和する事が期待できる．

Wraplet/A の処理系は具体的な規則と抽象的な規則が混在したラッピング規則を処理しなければならず，その実現方法は自明ではない．本研究では GNFA(各状態遷移が任意の正規表現に対応する非決定性有限オートマトン)[3] を用いた Wraplet/A の処理系を提案した．

表 1　実験結果



Wraplet/A を用いたラッピングの手順 (図 3) について説明する．(1) 利用者が，ラッピング対象となる HTML 形式のデータ h に対する Wraplet/A 式 e を記述し，e と h をシステムに入力する．(2) 入力された e が抽象的であった場合，システムは利用者にインタラクティブに問合せを行いながら，具体化した Wraplet/A 式 e' を作成する．ただし，最初の入力 e が既に具体的であった場合は，$e' = e$ となり何もしない．また，Wraplet/A 式に含まれる抽象的なラッピング規則が多いほど，インタラクションの回数やシステムの処理時間が増加する．(3) 具体化された Wraplet/A 式 e' と HTML 形式のデータ h がラッピングエンジンに渡され，ラッピング結果 x を出力する．

評価．実際のいくつかの種類の Web ページを対象として，ラッピング要求を仮定し，仮定した要求に基づいて抽象的な Wraplet/A 式を記述し，記述した式から具体的な Wraplet/A 式に変換可能か評価した．表 1 に実験結果を示す．今回実験した多くの Web ページに関しては，仮定した要求に基づいた抽象的な Wraplet/A 式を具体的な式 Wraplet/A 式に変換可能なことが確認できた．

4.　まとめ

本研究では，既存のラッピング言語ではサポートされていなかった機能を実現した新たな 2 つのラッピング言語を提案した．具体的には，2 つのラッピング言語について，機能を実現するための構文および処理アルゴリズムの検討を行い，実際の Web コンテンツを対象として評価を行った．

文献

[1] Natsumi Sawa, Atsuyuki Morishima, Shigeo Sugimoto, Hiroyuki Kitagawa. Wraplet: Wrapping Your Web Contents with a Lightweight Language. Proc. IEEE SITIS' 2007.
[2] Arnaud Sahuguet and Fabien Azavant. Building light-weight wrappers for legacy web data-source using W4F. In The VLDB Journal, 1998.
[3] Bakhadyr Khoussainov,Anil Nerode. Automata theory and its applications. Birkhäuser Boston, 2001.