

ポータルサイト運営者のための 軽量マッシュアップ開発ツールの提案と評価

新野 朝丈¹ 平山 雅樹¹ 松澤 芳昭^{2,a)} 児玉 公信^{3,b)} 太田 剛^{2,c)}

受付日 2012年4月23日, 採録日 2012年10月10日

概要: 本論文では, エンドユーザである運営者自身が利用できる軽量ポータルサイト開発システムを提案する. 本システムは, 運営者が業務時間合間の限られた時間でポータルサイトの構築・運用ができることを目標とする. 本システムの設計面での特徴は, 1) スクレイピングからクロールリング, サイトの運用まで一貫して開発が可能な「抽出規則作成分離型アーキテクチャ」を採用する, 2) エンドユーザが使用できる PBD (Programming By Demonstration) を基本インタフェースとして要素技術を組み合わせる, の2点である. エンドユーザ 6 名 (25~63 歳の事務職) を被験者とする使用実験では, スクレイピング規則の定義からサイト運用までの一貫した課題の成功率が 100%であり, それにかかわる時間も既存ツール (Karma) と同等以下であり, 目標とした 10 分以内に収まることが確認された. TAM (Technology Acceptance Model) に基づくユーザのシステム受容度調査では, 「システムを使うことの楽しさ」, 「出力の品質」で良い評価を得て, 「有用性」, 「使用性」は中立の評価を得た. 以上をもって, 提案するシステムの有用性を示した.

キーワード: マッシュアップ, PBD (Programming By Demonstration), 事務員, エンドユーザコンピューティング

Proposal and Evaluation of Lightweight Mashup Development Tool for End-user Administrators of Portal Site

TOMOTAKE NIINO¹ MASAKI HIRAYAMA¹ YOSHIAKI MATSUZAWA^{2,a)} KIMINOBU KODAMA^{3,b)}
TSUYOSHI OHTA^{2,c)}

Received: April 23, 2012, Accepted: October 10, 2012

Abstract: In this paper, we propose a light-weight mashup development tool, designed for clerical workers who are given the responsibility of being end-user administrators of portal sites who do not have any programming skills. The tool allows end-users to develop and maintain their sites within limited time apart from their main responsibilities. The two characteristics of the system design are 1) “The Architecture of Definition-Application Separation for Extraction Rule” that enables end-users to manage entire procedure of a portal site, such as scraping, crawling, site creation, and uploading to server, and, 2) the PBD (Programming By Demonstration) method which is applied as the main user interface of the tool. Our experimental study shows that 100% of six clerical workers (ages 25 to 63) with no programming experience succeeded in completing the development and maintenance of a portal site within ten minutes. Their task completion time was faster than that attained with the most successful tool to date (Karma). The results of the questionnaire based on the TAM (Technology Acceptance Model) showed that the system was accepted by the subjects with positive responses including “Enjoyment” and “Output Quality”. Their evaluation of the tool was neutral for “Usefulness” and “Ease of Use”. We conclude the effectiveness of the proposed system as being verified by these results.

Keywords: Mashup, PBD (Programming By Demonstration), Clerical Worker, End-User Computing

1. はじめに

WWW (World Wide Web) の入口にあたるものとして様々な「ポータルサイト」が構築されてきた。「ポータル」という用語は、異なる複数の情報システムのプレゼンテーションレイヤをソースとして、パーソナライゼーション、認証、コンテンツ集約を提供する Web アプリケーションを指す [1].

これと類似する概念として「マッシュアップ」がある。Albinola らは、ウェブマッシュアップについて「複数の Web ページから収集したい情報のみを取り出し、組み合わせることによって新たな情報を創出する技術」と定義している [2]. それぞれの目的・技術については同等と考えられるが、本論文ではそれらの用語を整理して、ポータルを作成するための技術をマッシュアップと呼ぶことにする。

我々はマッシュアップ技術を利用して、複数の学協会の Web ページからイベント情報を収集するポータルサイトを試作した [3]. このシステムのように、継続的な更新が行われるマッシュアップについては、規則の生成管理はエンドユーザ（ここでは、サイトの運営組織の事務員を想定する。以下「運営者」と呼ぶ）が実施できることが望ましい。しかしながら、特定の言語による規則の作成や、サーバへのアップロードなどの作業は、主として技能不足の問題で運営者による作業をすることは難しい。

本研究のテーマは、上記問題を改善し、ポータルサイトを運営者自らが構築・運営できるツールを開発することである。これにより、ポータルサイトの開発コスト・運営コストを削減できる可能性がある。実際の開発プロジェクトを分析して必要な要件を定義し、その要件を満たす範囲で既存の技術を組み合わせ、または既存技術の改良を実施する。

本論文は全 9 章からなる。続く 2 章で、我々が実施したポータルサイト開発の経験を説明し、運営者のためのツールの要求をまとめる。次に基礎概念として 3 つの用語の定義を 3 章で行う。4 章で先行研究のレビューを行い、5 章では提案するシステムの概要と設計について、既存技術の組合せ方法に焦点をあて述べる。6 章では本システムの評価実験の方法、7 章で結果を述べる。8 章で結果を吟味して、本研究の有効範囲について議論する。9 章はまとめて

ある。

2. 要求分析

本章では、提案するシステムの要件を定義する。実際に発注されたプロジェクトである「学会 세미나情報収集システム開発プロジェクト」から要求を分析し、要件としてまとめる。

2.1 学会 세미나情報収集システムの概要

要件抽出対象とする「学会 세미나情報収集システム」は、日本工学会が実際に、あるソフトウェア企業に外注したシステムである。我々は、正規の受注企業とは独立したプロジェクトとして、学生が並行して開発を行うプロジェクトを実施した [3].

このプロジェクトのテーマは、日本工学会 CPD (Continuous Professional Development) 協議会に所属する学協会が主催する講演や講習会、セミナープログラムの情報（以下、セミナー情報）を自動的に集約し、学協会横断的にセミナー情報を閲覧可能にする Web システムの開発である。セミナー情報とは、各学協会が主催する講演や講習会、セミナーのタイトルと日付、開催場所である。

2.2 学会 세미나情報収集システムの機能と問題点

我々が開発した「学会 세미나情報収集システム」の機能をまとめると、以下のようになる。

機能 1 抽出規則登録機能

システムが各学協会のサイトからセミナー情報を抽出（スクレイピング）するために、その規則を記述し、登録することができる。

機能 2 セミナー情報収集機能

システムは学協会のイベントカレンダーページを巡回（クローリング）し、登録された抽出規則を適用して、情報を収集することができる。システムは収集した情報をデータベースに登録する。

機能 3 セミナー情報提示機能

システムはデータベースに登録されたセミナー情報を Web ページとして閲覧者に提示することができる。この Web ページではセミナー情報の検索、ソートが可能である。

このシステムは、発注者の受入れテストに合格し、実際に利用できるシステムであることが認められた。しかしながら、実際の運営にあたっては次の問題点が考えられた。

第 1 の問題は、抽出規則の管理である。抽出対象となる Web ページはときとして構造が変化し、以前に作成した抽出規則では抽出が行えなくなることがある。その場合、Web ページに対する抽出規則を再度作成し、登録しなおす必要がある。抽出規則ファイルをサーバに再登録、またはプログラムを変更しサーバに再登録する必要がある。いず

¹ 静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University,
Hamamatsu, Shizuoka 432-8011, Japan

² 静岡大学情報学部
Faculty of Informatics, Shizuoka University, Hamamatsu,
Shizuoka 432-8011, Japan

³ 株式会社情報システム総研
Information Systems Institute, Ltd., Minato, Tokyo 107-
0662, Japan

a) matsuzawa@inf.shizuoka.ac.jp

b) kodamak@isken.co.jp

c) ohta@inf.shizuoka.ac.jp

れにせよ，専門的な知識を必要とするため，開発者が作業を実施することになる．これと同等の問題として，抽出対象 Web ページの追加，削除などの際にも開発者へ抽出規則の作成を依頼しなければならない．

第2の問題は，サーバの管理である．専用サーバでの運用については，セキュリティの問題や負荷，ハードウェアトラブルなどの問題を考慮する必要がある，サーバを管理できる技術者が必要になる．抽出規則の管理についても，抽出規則の変更ファイルの更新のたびに技術者がサーバにアクセスし，その処理を実施する変更がある．

このように，1 回限りの開発で，一定期間は正常に動作することが期待できるシステムであるが，中長期の運営に際しては，相応のメンテナンスコストがかかることが予想された．

2.3 ポータル作成システムの要件

上記の問題について検討し，サイトの「運営者」に注目することによる問題の解決を考えた．運営者とは，たとえば，学会セミナー情報収集システムでは，発注者の組織である日本工学会のオフィスに常駐している事務員を指す．こうした運営者が低コストで利用できる軽量化されたポータル開発・運営システムを開発することによって，先に述べた問題を解決し，運営コストだけでなく，開発コストも含めて削減できる可能性がある．

本研究で提案するポータル開発・運営システムの要件を次のようにまとめる．

要件1：運営者が本システムを使用することができること
運営者は，以下の要件を満たすエンドユーザである．

- ポータル提供に関連する組織で働いており，ドメイン知識がある．
- Web ページを巡回し，収集したい情報を指示できる．
- ブラウザ使用可能．
- プログラミング能力の有無を問わない．
- 使用できる時間 10 分/週．

エンドユーザの定義は 3.1 節で行う．利用できる技能に制限があるため，必要な技能のレベルを下げる必要があり，ユーザの本務ではないため，低コスト（短時間）で実施できる必要がある．要求技能レベルが低く，短時間でできることを本論文では「軽量」と呼ぶ．

要件2：抽出規則の作成からサイトの運用まで一貫して実施できること

エンドユーザが使用するシステムであるため，2.2 節で列挙した，スクレイピング（機能1），クローリング（機能2），およびサイト構築（機能3）が1つのパッケージになっている必要がある．さらに，2.2 節の後半で議論したように，1 回限り，一定期間正常に動作するだけでなく，中長期の運営まで一貫して実施できる必要がある．具体的には，要件1とあわせ，運営者が抽出規則を作成・改変でき，サ

イトに反映できる必要がある．

要件3：運営者がサーバの管理を実施する必要がないこと

システムを開発・運用するためには，当該システムのソフトウェアだけでなく，OS（Operating System）やハードウェアの管理も必要になる．しかしながら，エンドユーザを想定するとき，技能の問題によりサーバ管理を期待することはできない．したがって，アプリケーション層以下のレイヤについては，エンドユーザによる運営の対象外とする．

3. 用語の定義

3.1 エンドユーザ

マッシュアップを支援するためのツールについて，Albinola らは対象とするユーザの技能を3レベルによって分類する方法を提案している [2]．その3レベルとして，プログラミングの技能を保持している者を developer，プログラミングはできないが専門的ツール^{*1}を使いこなすことができる者を poweruser，Web ブラウザのみ利用できる者を casualuser と定義している．

本論文では，そのうちの poweruser と casualuser，つまりプログラミングの技能を持たないユーザ向けのツール提案をテーマとし，それらをまとめて「エンドユーザ」と呼ぶ．

3.2 マッシュアップ工程

Tuchinda らはマッシュアップ作成システムにおいて支援が必要な工程として，下記の5工程を定義している [4]^{*2}．本論文では，このマッシュアップ工程の定義を基礎として，システム設計，評価の議論を行う．

Data Retrieval

データ収集対象 Web ページに対して，抽出規則を定義する工程である．Web ページからテキストを抽出する規則の記述手段として一般的なものに，XPath [5] がある．XPath によって，HTML ドキュメント上の任意のタグの位置を記述することができる．

Source Modeling

抽出したデータを保存するデータベーススキーマを定義する工程である．たとえば，学会セミナー情報では，「イベント名」，「開催日」など，抽出したデータを意味づけする作業がこれにあたる．

Data Cleaning

抽出したデータを標準化する規則を定義する工程である．たとえば，日付データであれば，「10 月 1 日」と

^{*1} 例として，スプレッドシートやブロック線図構築などがあげられている．

^{*2} 原著ではこの5工程をマッシュアップ工程として定義し，Data Display を除いた4つの工程についてツール支援の結果を述べている．本論文では Data Display 工程も支援対象に含むため，5工程すべてを扱う．

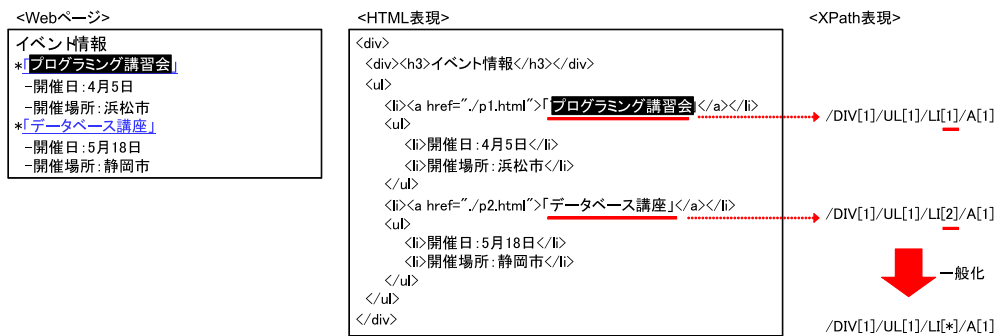


図 1 PBD による抽出規則の定義とその一般化

Fig. 1 Creation and generalization of the extraction rule by PBD.

いうデータを「10-1」のような形式に統一する作業がこれにあたる。

Data Integration

複数のデータベーステーブルの結合規則を定義する工程である。このとき、結合するデータベーススキーマが同様の場合は Union といい、行を追加する形で自動的に結合される。結合するデータベーススキーマが異なる場合は Join といい、どの列とどの列を結合するか、その規則を指示する必要がある。

Data Display

収集されたデータを提示する工程である。たとえば、学会セミナー情報収集システムでは、収集されたデータを一覧にまとめ、Web ページとして提示している。原著では触れられていないが、一覧上での検索やソーティングなどの機能提供もこの工程に含まれるものと定義する。

3.3 Programming By Demonstration (PBD)

PBD (Programming By Demonstration) とは、ユーザがプログラムを記述せずに、オブジェクトの操作によってアルゴリズムを設計するための手法である。これによってユーザはプログラムを直接記述せずにアルゴリズムの設計が可能になり、プログラムを記述するよりも早くアルゴリズムを作成できるとされている [6]。

PBD のマッシュアップにおける適用例として、Data Retrieval 工程における例を用いて説明する。抽出規則の定義とその抽象化の様子を示した図を図 1 に示す。まず、ユーザが Web ページを選択し、どのテキストを抽出したいのかを提示する（図 1 では、<Web ページ>の「プログラミング講習会」が選択された例を示している）。システムは HTML テキストを解析し、当該テキストの位置を示す XPath を自動生成する（図 1 では、<XPath 表現>に “/DIV[1]/UL[1]/LI[1]/A[1]” が生成されている）。この XPath が抽出規則となる。

2 つ以上の例を選択した場合、抽出規則の一般化が行われる。図 1 の右部は、抽出規則抽象化の様子を示してい

る。ユーザが「プログラミング講習会」と「データベース講座」を選択すると、それぞれの XPath が生成される。システムはこの 2 つの XPath の一般化を行う。2 つの XPath の異なる場所を検索し（赤下線に示した箇所）その箇所を「*（ワイルドカード）」に変更する。ワイルドカードは、抽出の際に任意の文字として解釈することを表す。したがって、タグの構造が繰返しのパターンによって構成されている場合には、出現するすべての繰返しパターンに対応するデータの抽出が期待できる。

このように、PBD 方式では、ユーザが直接プログラムを書くのではなく、いくつかの例を提示して、プログラムが生成される。そのため、プログラミング言語の知識がなくても操作可能である。

4. 先行研究

マッシュアップ技術によるデータ収集支援ツールは多数提案されている。Beemer らによるマッシュアップ支援ツールのレビュー論文では、ツールを 7 種類のカテゴリに分類している [15]。我々はその中でも “End User Programming” のカテゴリに属するツール、およびその派生研究に注目した。このカテゴリのツールの中には 2 章で述べた要件を部分的に満たすものが散在する。

以下の節では、End User Programming のカテゴリに属するマッシュアップツールについて、1) Widget 型、2) PBD 型、3) どちらにも属さないもの、の順に紹介し、その特徴をまとめる。

4.1 Widget 型

Marmite [7], Yahoo’s Pipes [8], Microsoft Popfly [9] はいずれも Widget 型のマッシュアップツールと呼ばれている。ユーザは Widget と呼ばれる、ブロック型のモジュールをブロック図のようにつなぎ合わせることでプログラミングを実施することができる。

Marmite が最初に提案された Widget 型のツールであり、Yahoo’s Pipes は Widget 型のツールを Web サービスとして提供している。Microsoft Popfly は他の 2 つのツールに

表 1 マッシュアップツールの比較
Table 1 Comparison between different mashup tools.

ツール名	Data Retrieval	Source Modeling	Data Cleaning	Data Integration	Data Display
Marmite [7]	Programming	Manual	Manual	Manual	WebPage, RDF
Yahoo Pipes [8]	Programming	Manual	Manual	Manual	WebPage, RDF
Microsoft Popfly [9]	Programming	Manual	Manual	Manual	WebPage, RDF
Piggybank [10]	RDF	Manual	N/A	Manual	WebPage
Potluck [11]	RDF	Manual	PBD	Manual	WebPage
Dapper [12]	PBD	Manual	Manual	Manual	Table, RDF
Karma [4]	PBD	Database	PBD	PBD	Table
Vegemite [13]	PBD	Manual	N/A	Manual	Table
iGoogle [14]	Programming	Manual	Manual	Manual	WebPage

比べて Widget の種類が最も多く、また使用可能な Widget を推測し、ユーザに提示するという特徴がある。

Tuchinda らは Widget を組み合わせることとプログラミングを書くことは基本的に同じであると指摘している [4]. そのため、エンドユーザが Widget 型のマッシュアップツールを使用することは難しい。

4.2 PBD 型

Dapper [12], Karma [4], Vegemite [13] はいずれも、抽出規則の生成に PBD 方式を採用している。

Karma は 2008 年に最初のバージョンが提案され [16], 2011 年に最新のバージョンが発表されている [4]. 現段階では Data Display を除く 4 つの工程について PBD 方式が実装されている。Dapper は無料の Web サービスとして提供されており、出力形式を RDF (Resource Description Framework) 形式、表形式、Web ページ形式で選択できるという特徴がある。Karma では抽出規則の作成だけでなく、データベースの設計も PBD 方式を採用しているという特徴がある。Vegemite は PBD によって生成された抽出規則をスクリプトで出力し、ユーザが自由に編集できるという特徴がある。

Potluck [11], Piggy Bank [10] はともに、RDF 形式の Web ページのみを対象とした PBD 型マッシュアップツールである。RDF では Web ページの中にテキストと属性名がタグに示されており、システムはそのタグの位置情報を利用してデータを抽出する。Potluck では Potter's Wheel 方式 [17] を用い、RDF ファイルから取り出したテキストの編集に PBD 方式を採用している。しかしながら、これらのシステムでは RDF 形式ではないページからテキストを抽出することができない。

4.3 その他

本節では Widget 型にも PBD 型にも該当しないエンドユーザ向けと謳われているマッシュアップ作成ツールを紹介する。

表 2 ツール比較に使用した用語の説明

Table 2 Descriptions of terminologies used in the comparison of the tools.

工程	用語の説明
Data Retrieval	PBD は PBD 方式によるもの、RDF は RDF ファイルのタグ指定して抽出規則として用いるもの、Programming はプログラムを直接記述するもの ^{*3} 。
Source Modeling	Manual は、ユーザが属性名を 1 つずつ入力してスキーマを定義するもの、Database は Manual 方式に加えてシステムが過去に作成したテーブルを自動的に探索し、スキーマを推測することができるもの。
Data Cleaning	Manual はユーザがテキスト標準化する規則を直接記述するもの、PBD は PBD 方式によるもの、N/A は Data Cleaning を実施しないもの。
Data Integration	Manual はユーザが結合方法を提示するもの、PBD は PBD 方式によるもの。
Data Display	WebPage は収集したテキストが Web ページとして提示されるもの、RDF は収集したテキストが RDF ファイルとして出力されるもの、Table は収集したテキストが表としてローカルに保存されるもの。

iGoogle [14] は、ユーザが個人的な Web ページを作成するための Web サービスである。iGoogle では Gadget と呼ばれるマッシュアップ要素を複数提供しており、ユーザは自分の目的に適った Gadget を選択し、Web ページに配置する。Gadget を作成するためにはプログラミングを行う必要がある。

エンドユーザが直接プログラミングを記述することの補助を行うためのツールも提案されている [18], [19], [20]. これらのツールはユーザがプログラミングを行う必要がある。

^{*3} 本論文では Widget を含め、ユーザが直接命令を記述する工程をすべて Programming とする。Tuchinda らは Programming について詳細に定義しているが、本システムでは Programming に該当するものはすべてエンドユーザが使用できないため、Programming 以下の詳細な分類を重視しない。

4.4 ツールの分類

本章で紹介したツールを分類し、表 1 にまとめる。表 1 における縦列は、3.2 節で述べたマッシュアップ工程を利用している。Tuchinda らも同様の表でツールを分類しているが、その観点が PBD の適用であるのに対して、我々は End User Programming のツールを網羅的に比較する目的から特徴を再定義しているところが異なる。再定義された分類定義を表 2 に示す。

5. システム設計

本章では、開発したポータルサイト作成システムについて、まずシステム概要について、ユーザインタフェースを提示しながら説明し、次に内部設計としてシステムアーキテクチャ、要素技術の組合せの考え方を順に述べる。

5.1 システム概要およびユーザインタフェース

5.1.1 システム概要

本システムは、大まかに以下の 3 部から構成される。

- (1) 抽出規則定義部
- (2) ポータルサイト生成部
- (3) データ抽出・提示部

上記のうち、「抽出規則定義部」「ポータルサイト生成部」はブラウザ埋込み型 (Firefox アドオン型) の GUI (Graphical User Interface) 方式のクライアントアプリケーションである。Web ブラウジング作業と並行して PBD 方式による例の指定ができるようになっている。

「データ抽出・提示部」は、ポータルサイトコード生成方式により生成された Web アプリケーションである。Web アプリケーションフレームワークとして CakePHP を採用し、生成された PHP のコードを編集することで抽出結果一覧ページのレイアウトの変更や機能の追加を行うことができる。

以下、それぞれの要素の詳細機能について、ユーザインタフェースを紹介しながら述べる。

5.1.2 抽出規則定義部

「抽出規則定義部」「ポータルサイト生成部」は Firefox プラグインとして動作する。Firefox から起動された「抽出規則管理ウィンドウ」を図 2 に示す。抽出規則管理ウィ



図 2 抽出規則管理ウィンドウ

Fig. 2 The window to manage extraction rule.

ンドウは、「Web ページ」、「項目」、「例」の 3 つの区画からなる。

「Web ページ」区画は抽出対象ページを管理することが目的である。「Web ページ追加」ボタンを押下することで、現在ブラウザ上に開いている Web ページが抽出対象ページとして追加される。

「項目」区画は、Web ページから抽出するデータスキーマを定義することが目的である。「項目追加」ボタンを押下することで、項目の追加することができる。項目の登録時に、項目の型の選択も行う。

「例」区画は、PBD 方式によりデータを選択するための例を管理することが目的である。「例追加」ボタンを押下することで、例を追加することができる。例の指定は、「例追加ウィンドウ」で行う。「例追加ウィンドウ」を図 3 に示す。例として追加したいテキストを選択し (図 3 では中央のハイライト箇所)、「OK」ボタンを押下することで、例の追加を行うことができる。

例を登録した後は、URL、項目、例のそれぞれのリストにおいて、右クリックメニューから、その段階での抽出規則で抽出することができるテキストのプレビューを表示することができる。

5.1.3 ポータルサイト生成部

ポータルサイト生成は、先に述べた「抽出規則管理ウィンドウ」 (図 2) で行う。必要な Web ページ、項目、例の登録が完了した後、ウィンドウ上部にある「ポータルサイト名」にポータルサイトの名前を入力し、「アップロードボタン」を押下すると、システムは Web アプリケーションコードを生成し、サーバにアップロードする。

5.1.4 データ抽出・提示部

ポータルサイト生成部で生成され、アップロードされた Web アプリケーションは、そのままサーバ上でアプリケーションとして動作する。作成されたサイトの一例を図 4 に示す。この例では、情報処理学会 IPSJ カレンドと電気学会の学会イベントカレンダーから、イベント名称と日付、開催場所を抽出している。

生成された Web アプリケーションはデフォルトで、抽

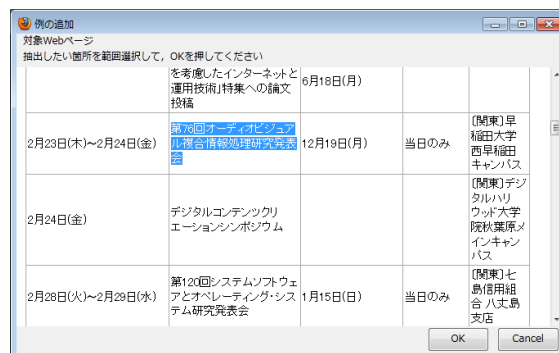


図 3 例追加ウィンドウ

Fig. 3 The window to add example of the data.

出規則に従ってデータを抽出する定期巡回（クローリング）機能、および、抽出結果の表形式による提示、整列機能が提供される。

5.2 システムアーキテクチャ

ポータルサイト作成システムの構築に際し、3つのシステムアーキテクチャを考案した。それらを図5に示す。以下、これらのアーキテクチャについて説明し、トレードオフ分析を行う。

5.2.1 ローカル型アーキテクチャ

図5の(a)はローカル型アーキテクチャである。このアーキテクチャは、抽出規則生成部とデータ抽出・提示部から構成されている。抽出規則生成部では、ユーザの指示に基づき、抽出規則を生成する機能を持っている。データ抽出・提示部では、生成された抽出規則を基に、Webページからテキストの抽出を行う。抽出したテキストは、テーブ

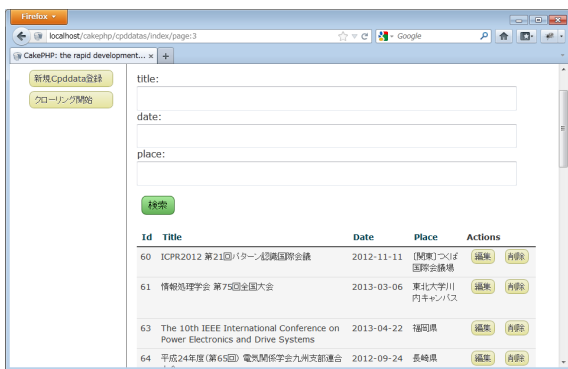


図4 生成されたポータルサイト

Fig. 4 The generated portal site.

ルの形にまとめられ、ユーザへと提示される。このアーキテクチャを持つツールとしては、Karma [4] があげられる。

5.2.2 サーバ型アーキテクチャ

図5の(b)はサーバ型アーキテクチャである。このアーキテクチャも、抽出規則生成部とデータ抽出・提示部から構成されている。抽出規則生成部では、ユーザの指示に基づき、抽出規則を生成する機能を持っている。データ抽出・提示部では、Webページごとに生成された抽出規則を基に、Webページからテキストの抽出を行い、抽出したテキストをデータベースへと保存する。抽出したテキストは、RSS (RDF Site Summary) フィードとして配信する、あるいはWebページとして表示することで、ユーザへと提示される。このアーキテクチャを部分的に持つツールとしては、Dapper [12] があげられる。

5.2.3 規則作成分離型アーキテクチャ

図5の(c)は規則作成分離型アーキテクチャである。このアーキテクチャは、抽出規則・コード生成部とデータ抽出・提示部から構成されている。抽出規則・コード生成部は、クライアントアプリケーションとなっており、ユーザの指示に基づき、抽出規則を生成するとともに、データ抽出・提示部において、クローリングおよび抽出結果の提示に使用されるソースコードを生成し、ポータルサイト用サーバへとアップロードする。データ抽出・提示部では、アップロードされたソースコードを使用して、Webページごとに生成された抽出規則を基に、Webページからテキストの抽出を行い、抽出したテキストをデータベースへと保存する。抽出したテキストは、アップロードされたソースコードに記述されているレイアウトの形でWebページとして

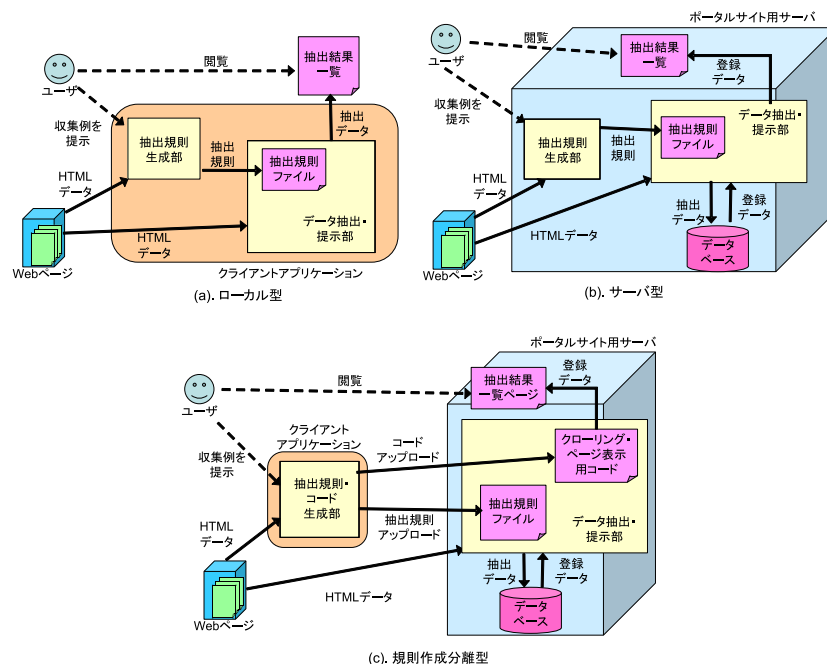


図5 ポータルサイト作成システムアーキテクチャの比較

Fig. 5 Comparison of three architecture for portal site development system.

表 3 要件とアーキテクチャの対応

Table 3 Table of selectable architecture and requirements.

	要件 1	要件 2	要件 3
ローカル型	○	×	×
サーバ型	△	△	○
規則作成分離型	○	○	○

ユーザへと提示される。また、アップロードされたソースコードを編集することで、抽出結果を提示する際のレイアウトをユーザの手により改変することが可能である。

5.2.4 アーキテクチャの選択

3つのアーキテクチャについて検討した結果を表3に示す。表3において、“○”は要件を満たすことを、“×”は要件を満たさないことを、“△”は要件を部分的に満たすことを示す。

要件1（エンドユーザが使用できる）については、抽出規則生成部にPBD方式を利用するために、ユーザがブラウザの機能を利用する操作が必要なため、ローカル型か抽出規則生成型である必要がある（ただし、サーバサイドで専用ブラウザプログラムを用意することは高コストではあるが技術的には可能であるため、△としている）。

要件2（サイトの運用まで一貫して実施できる）については、まず、ローカル型ではサイトの公開をすることが不可能である。サーバ型、規則作成分離型の双方でサイトの公開が可能であるが、規則作成分離型は1回限りの動作ではなく、中長期的な運営をするための利点がある。規則作成分離型ではコードが生成される仕組みになっているため、開発者によるアプリケーションの改変が容易であることである。サーバ型ではシステムを提供するサーバ上にプログラムやテンプレートが存在するため、使用者ごとのシステムの改良や、レイアウトテンプレートの変更などのページの調整が難しく、セキュリティレベルも落ちる。開発者と協同がしやすいアーキテクチャは運営者の利点である。これが、サーバ型の評価を△とした理由である。

要件3（運営者がサーバの管理をする必要がない）については、サーバ型および規則作成分離型においては、ポータルサイトの運用はあらかじめ用意されたサーバに行われるため、運営者はサーバの管理を行う必要がない。一方ローカル型は、収集した結果はファイルとして出力されるため、Web上で公開し共有するためには、運営者自身がサーバを用意し、Web上に公開する方法を構築する必要がある。

このように、本システムの設計においては、既存ツールの方式およびそれらの組合せを検討し、システム要求に合致し、費用対効果の高いアーキテクチャデザインを考案し、アプリケーションとして適用した。

5.3 要素技術の組合せ

本システムの設計においては、作業工程ごとに利用でき

表 4 既存技術の要件充足性

Table 4 Requirements meeting table for existing technologies.

作業工程	特徴	要件 1	要件 2	要件 3
Data Retrieval	Manual	×	○	-
	PBD	○	○	-
	改良 PBD	○	○	-
	RDF	△	×	-
Source Modeling	Manual	○	○	-
	Database	△	○	-
Data Cleaning	Manual	×	○	-
	PBD	○	×	-
	N/A	△	×	-
	Auto	○	○	-
Data Integration	Manual	×	○	-
	PBD	△	○	-
	Auto	○	○	-
Data Display	規則作成分離型	アーキテクチャ選択を参照		

る要素技術を吟味し、要件と照らし合わせて費用対効果の高い組合せを考えるとともに、必要に応じて既存の技術を改良して採用した。

表4に各工程における要素技術とシステム要件の充足性の関係を示す。なお、表中において、“○”は要件を満たすことを、“△”は要件を部分的に満たすことを、“×”は要件を満たさないことを、“-”は要件と関連性がない特徴であることを示している。方式欄の太字は本システムで選択する技術であることを示している。以下、それぞれの作業工程ごとの技術採用理由を説明する。

5.3.1 Data Retrieval

本システムでは、Data Retrievalの実装方針として改良型PBDを選択する。Manualではプログラミングの専門知識がなければ実施できず、RDFではRDF形式以外のWebページからテキストを収集できないためである。

PBDは一見すべての要件を満たしているが、PBDによる抽出規則の作成にも問題がある。PBDでは、抽出規則としてXPathを使用し、XPathに基づいてテキストを抽出する。しかし、XPathによって抽出されるテキストは、HTMLタグの中に存在するテキストすべてであり、テキストの一部だけを取り出したい場合に、対応することができない。タグの中のテキストの一部を切り出して抽出することもあるため、タグの中のテキストの一部を切り出すアルゴリズムが必要となる。

これを実現するための方法として、XPathにテキストマッチングを加えた改良PBD型アルゴリズムを開発した。抽出規則を生成する際に、抽出対象のテキストの前後にあるテキストを切り出しパターンとして記録しておき、XPathによりタグの中のテキストを抽出した後に、切り出しパターンとのマッチングを行う。切り出しパターンがマッチした場合には、その位置でテキストを切り出すこと

で、タグの中のテキストの一部を取得することができる。

5.3.2 Source Modeling

本システムでは、Source Modeling の方式として Manual (ユーザが収集するデータスキーマを定義) を採用する。Source Modeling の方式として、Manual 方式と Database 方式が提案されており、いずれも要件 1 を満たす。Database 方式は、1 度定義したスキーマと同様のスキーマを定義しようとする際にデータベースから適切な候補をユーザに提示する方式であるが、初回定義時には Manual よりも作業時間がかかることが分かっている [4]。本システムにおいては、ポータルサイト 1 つにつきデータスキーマは 1 つであり、スキーマの再利用は考慮する必要がない。したがって、本システムでは初回定義時の作業時間が短く、仕組みが単純な Manual 方式を選択する。

5.3.3 Data Cleaning

本システムでは、Data Cleaning の実装方針として Auto を選択する。既存の技術ではすべての要件を同時に満たすことができないためである。PBD ではすべての要件を満たすことができそうに見える。しかし、Data Cleaning の PBD 手法である Potter's Wheel は英語を前提とした技術であり、日本語に対応していない。そのため要件 1 を満たすには不十分である。

そこで、本システムでは、ユーザ自身には編集規則の提示を求めない「Auto」アルゴリズムを開発した。ここでいう Auto とは、データタイプごとに変換ルールをあらかじめ定義しておくことにより運営者が編集する必要がない環境である。なお、データタイプとしては標準的によく利用する数値、文字列、日付^{*4}、時間^{*5}の 4 つの型に限定したが実用上問題はない（その程度については 8.2 節で議論する）。

5.3.4 Data Integration

本システムでは、Data Integration の実装方針として Auto を選択する。Auto とは、Union のみに限定し、自動でデータを結合する方式である。Manual はプログラミングの専門知識がなければ実施できず、PBD は Auto と比較して作業時間が見込まれる。Auto では、結合が Union に限定されることになるが、Join については、Modeling の際に決められた一意のスキーマにあわせてデータを抽出することで同等のポータルサイトが開発可能である。

5.3.5 Data Display

Data Display に関しては、5.2 節で述べたアーキテクチャ方式がその検討事項とおおむね対応する。本システムでは

規則作成分離型アーキテクチャを採用することにより、クローリングやソーティングも含めた総合的なポータルサイトを生成する。

6. 評価方法

本システムを評価するために、「エンドユーザによるポータルサイト作成実験」を実施した。本章では、その方法を述べる。

6.1 被験者と実験概要

被験者は著者らの所属大学に勤務している事務員 6 名のボランティア（無報酬）である。被験者の年齢は 25 歳～63 歳（平均 43.8 歳）、性別は男性 3 名、女性 3 名であった。全員仕事でコンピュータを利用しており、ブラウザは使用でき、プログラミング経験はない。

実験では最初にプログラミング経験を調査した後、12 分の動画によって、本システムの使用方法について学習する機会を与え、続いて課題を実施し、最後にアンケート調査への回答を実施した。

6.2 設問

被験者が実施した課題は 3 問あり、それぞれ難易度と目的が異なる。表 5 に課題、難易度をまとめて示す。対象ページ数は、抽出対象となる Web ページが単一であるか複数であるかを、対象項目は Web ページから抽出する項目が単一であるか複数であるかを、Data Display 工程は実験課題として Data Display 工程を実施するか否かを示している。

課題 1 は操作練習が目的である。課題 2 は Karma の評価実験 [4] 同一の課題であり、本システムと既存システムの比較したインタフェース評価が目的である。要件 1 の評価に該当する。課題 3 は Data Retrieval から Data Display までの全工程を実施し、さらに自ら作成したポータルサイトを用いて必要なデータを探す課題も含まれている。要件 2 および 3 の評価に該当する。

6.3 アンケート調査方法

アンケート調査では、Technology Acceptance Model3 (TAM3) [21] を用いた。TAM3 は、ユーザがシステムを受容できるかどうかを調査するモデルであり、16 個の評価項目と、それらを評価するための 52 個の質問が提供される。

表 5 ポータルサイト作成実験課題

Table 5 Features of tasks done in this study.

	対象ページ数	対象項目	Data Display 工程
課題 1	単一	単一	含まない
課題 2	複数	複数	含まない
課題 3	複数	複数	含む

^{*4} 日付については、平成 23 年 3 月 14 日 (月)、2012/03/14、2012.03.14、2012-03-14、の形式に対応しており、年が指定されていなければ解析した年と見なす。月と日が逆転しているものは対応しておらず、日本語サイトのみで構成するマッシュアップでは問題ないが、英語を併用する場合には改良が必要である。

^{*5} 時間については、午前 3 時 10 分 40 秒、3:10:40am の 2 形式に対応しており、それぞれ午前/午後、時、分、秒の単位は省略できる。

このモデルを使用する理由は、秘書を対象としたマッシュアップツール Mixer [22] の論文でエンドユーザが知覚した有用性と使用性を測定する方法として提案されているためである。Mixer と同様に、下記の 5 つの項目についてユーザによるシステムの受容度を評価した。

Perceived Enjoyment (ENJ) システムを使うことが楽しいと感じたか。

Output Quality (OUT) システムの出力は高品質と感じたか。

Perceived Usefulness (PU) システムが有用だと感じたか。

Perceived Ease of Use (PEOU) システムを使うことが簡単と感じたか。

Behavioral Intention (BI) システムを使いたいと思うか。

アンケート調査では、まず評価項目ごとに関連づけられたいくつかの質問項目それぞれについて Likert 尺度による 7 段階評価 (1=まったく同意しない, 4=中立, 7=非常に同意する) を行う。次に、評価項目ごとのスコアについて、質問項目の平均値をとって算出する。

7. 結果

7.1 課題の成否と作業時間

課題ごとの課題完了までの作業実施時間の平均を表 6 に示す。課題を終えることができなかった被験者は失敗とし、成功者数を記録した。平均時間、SD (標準偏差) については、成功者についてのデータである。この表を含めて、以後、時間については「分:秒」表記で示す。

6 名の被験者がそれぞれ 3 つの課題に対して失敗した例はなく、成功率は 100% であった。実施時間についても、すべての課題について平均時間 10 分以内を達成し、かつ全員が 10 分以内に課題を終了している。標準偏差も小さく、被験者に多様性があることを考慮すると、エンドユーザが本システムを用いて 10 分以内でポータルサイトを作成可能、という目標は達成していると考えられる。

7.2 Karma との作業時間比較

課題 2 について、Karma での実験 [4] との比較結果を図 6 に示す。縦軸の単位は時間である。なお Karma での実験の被験者は、プログラマ 20 名 (情報系研究室の修士、博士) によるものである。比較する本システムの工程別時

間は、Karma での実験の測定方法に合わせ、システムを直接操作していた時間のみを収集している。そのため、図 6 の時間を合計しても表 6 と同一になっていない。

本実験とは母集団が異なるため、平均値を単純比較して速度の優劣を確定づけることはできないことを前提として、以下 3 点について考察する。1) 本システムでは Data Cleaning の時間がなく、その分 Karma より作業時間を短縮できている。2) Data Retrieval 工程について、本システムが約 2 分 30 秒、Karma が約 1 分 30 秒であった。母集団におけるプログラミング経験、年齢構成の条件が本研究の方が不利であることを考慮し、さらに両システムとも同様の方式 (PBD 方式) を採用していることを考慮すると、同等の作業効率であることが期待できる。3) Data Retrieval と Source Modeling の比率が、本システムが約 20%, Karma は約 40% であることから、初回定義時の Source Modeling における Manual 方式の作業効率の良さが示唆される。

7.3 アンケート結果

本システムのアンケート結果 (TAM3 のスコア) を箱ひげ図にまとめて図 7 に示す。縦軸の数値は 7 段階評価 (1 から 7) の素点である。それぞれの平均値は、ENJ (楽しめた)=5.1, OUT (品質)=5.3, PU (有用性)=4.3, PEOU (使用性)=4.3, BI (利用するか)=3.8 であった。

TAM3 [21] は、ENJ は PEOU 要因の 1 つとなり、OUT は PU の要因の 1 つとなり、それらの総合評価が BI となるモデルである。したがって、ENJ, OUT は中立より良い評価がされているものの、それらが被験者の PU, PEOU の評価に結び付いておらず、最終的な BI も中立程度 (平均は 1 名が押し下げている。中央値は 4.3) にとどまっている、と解釈できる。OUT が PU に結び付かなかった理由として、今回の被験者にとって、本システムを利用した課題の問題領域は仕事と直接関係ないことが考えられる。ENJ が PEOU に結び付かなかった理由として、実験や新しいシステムへの習熟に不慣れなことから実験への不安を口にする被験者が数名観察されたことから、作業の不慣

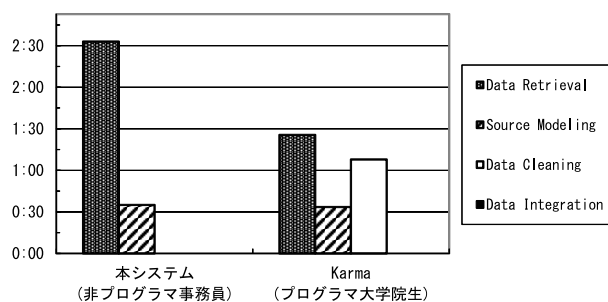


図 6 Karma との作業時間比較結果。Karma の結果を文献 [4] より引用

Fig. 6 Comparison of working time between the proposed system and Karma. Results of Karma was reprinted from [4].

表 6 課題の実施時間

Table 6 Time to complete the tasks.

	被験者数	成功者数	平均	SD
課題 1	6	6	5:00	0:40
課題 2	6	6	6:42	1:36
課題 3	6	6	6:12	0:40

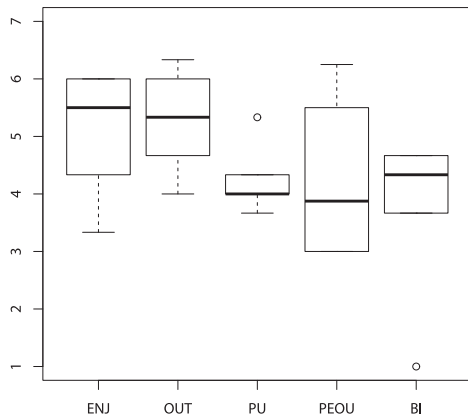


図 7 本システムの TAM3 スコア

Fig. 7 TAM3 score for our system.

れが理由で「利用が簡単である」という評価ができなかったことが考えられる。実験中に基本的な操作方法に疑問を持ったり、操作性の不満を持ったりする態度の被験者がいなかったことも、これを支持する。

これらの結果を総合すると、「1)「有用性」,「使用性」は、本システムが想定した運営者である事務員に対して、積極的に受容されるものではないが、拒否されることないレベルと評価された, 2) しかしながら、その中でも「システムを使うことの楽しさ」,「出力の品質」は受容できると評価されている」となる。この結果は、本システムが要件 1 である「エンドユーザが利用可能」という条件を満たすことを支持するものである。

8. 考察

本章では、実験結果を考察し、本システムの有効性および適用範囲を主張する。

8.1 要求目標の達成度

エンドユーザ使用実験の結果は、1) すべての被験者が本システムを用いて 10 分以内でポータルサイトを作成可能であること, 2) Karma と比較して同等以上の作業効率が期待できること, および, 3) TAM3 スコアは「有用性」,「使用性」に際して、本システムが想定した運営者である事務員に対して、中立と評価された, の 3 点にまとめられる。

1) の結果に際して我々が最も主張する点は、本実験の最後の課題（課題 3）は単なるマッシュアップ作成ではなく、ポータルサイトを作成し、公開し、そのポータルサイトを使用する工程までが一貫して課題に含まれている点である。我々の調査の範囲では、このようなポータルサイトの作成から使用まで一貫して処理するシステム、および実験を実施している例はない。本システムでは、規則作成分離型アーキテクチャによる設計によって抽出規則のアップロードが容易にできたことが、エンドユーザによる課題の成功率向上に寄与したものと考えられる。

2) の結果に際して、各被比較論文は研究の目的がまったくの同一ではないため、実験の母集団が異なるという問題はああるものの、一定の客観性をもってシステムのユーザビリティ、および受容可能性を示せたと考えている。同一ユーザによるシステム比較使用実験によって客観性の向上が見込まれるが、被比較論文ではシステムを公開していないため、そのような実験は不可能であったことを付記する。我々の実験では年齢構成・性別におい本研究が想定した被験者の多様性をカバーしていることから、実際の事務員の大部分が本システムを利用可能であることに確信が持てる結果である。

2) の結果は、我々が 5.3 節で述べた要素技術の組合せの妥当性を支持するものと考えられる。特に、Data Cleaning 工程で Auto 技術を開発したことは Karma と比較して工程そのものの工数を削減できており、軽量化の効果が大きいことが分かった。

3) の結果によって、本システムのユーザが認識する「有用性」,「使用性」は拒否されることのないレベルであるという結論になる。「システムを使うことの楽しさ」,「出力の品質」は評価されていることから、業務と関連する問題領域において、業務命令で行う場合には問題なく本システムは現場で適用可能であると我々は評価している。

これらを総合して、2 章で提示した本システムの要件、すなわち、「運営者が本システムを使用でき」,「抽出規則の作成からサイト運用まで一貫して実施でき」,「ユーザがサーバの管理を実施する必要がない」, システムの開発目標は達成されたと考えている。

8.2 システムの軽量化と適用範囲

本システムはポータルサイトの作成に特化してシステムの軽量化を行った。この軽量化によって下記の 2 つの制限が生じる。

- 複数抽出可能なデータはタグ繰返しパターン（表、リストなど）に限定される。
- Data Cleaning はあらかじめ用意された 4 パターンのみに限定される。

この制限事項が実際のプロジェクト構築においてどの程度の制約となるか。その詳細については本論文の対象範囲外とするが、程度を把握するために 2 種類の簡易実験を行った。1 つ目は学会 세미나 情報収集システムの再現で、14 件中 12 件を収集できた。もう 1 つは 6 章に示した実験とは別に、大学生 8 名に対して利用してみたい例を自由に選択させてサイトを作成させる実験で、17 件中 12 件を収集できた。合計して収集可能率は 75% ほどであった。

収集不可能サイトについて、結論からいえば、本システムの限界ではなく、PBD による Data Retrieval 方式そのものの限界であった。Adobe Flash などのサイトは専用のシステムが必要になることは自明で、そのほかの例では、

PBD による Data Retrieval が、単一のタグに単一のデータが前提となっていることによって生じるものであった。

これらのことから、システムの軽量化によって他の PBD 型ツール以上に対象サイトが制限されることはないといえる。我々の提案する改良 PBD 型により、単一のタグの部分テキストは抽出可能である。この機能を拡張し、2 カ所以上の部分抽出の機能追加によって改善が見込まれる。

9. まとめ

本論文では、実際のポータルサイト開発経験に基づいて、運営者（オフィスに常駐している事務員）が低コストで利用できる軽量ポータルサイト開発・運営システムを提案した。提案するシステムは、以下の3つの要件を満たすように設計された。

- 要件1：運営者が本システムを使用することができること
- 要件2：抽出規則の作成からサイトの運用まで一貫して実施できること
- 要件3：運営者がサーバの管理を実施する必要があること

我々は3つの要件を満たすツールを作成するために既存のマッシュアップツールの要素技術を組み合わせ、全体のソフトウェアアーキテクチャとして「規則作成分離型アーキテクチャ」を提案し採用した。我々の所属大学に勤務している事務員6名を被験者とする使用実験では、2項目を2ページから合成しマッシュアップを作り、サーバに公開し使用する総合的な課題の成功率が100%であり、かかる時間は平均約6分程度であった。かかる時間は目標である10分以内を達成し、Data Cleaning 工程の削減効果で既存ツールの同等以下になると評価した。ユーザのシステム受容度調査では、「システムを使うことの楽しさ」、「出力の品質」で7段階中5の評価を得て、「有用性」、「使用性」は4（中立）の評価を得た。以上をもって、提案するシステムが当初の要件を満たす有用なシステムであることを評価した。提案するシステムは、運営者、開発者とデザイナーの協業によるポータルサイト運用の普及に寄与し、ひいてはエンドユーザコンピューティングの普及に寄与することが期待できる。

謝辞 成瀬弘臣さんにシステム評価再実験の補助をしていただきました。静岡大学情報学部事務職員の皆様に無報酬の被験者として協力をいただきました。感謝いたします。

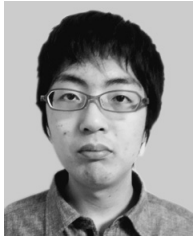
参考文献

- [1] Hepper, S.: Java Portlet Specification, available from <http://download.oracle.com/otndocs/jcp/portlet-2.0-fr-oth-JSpec/> (accessed 2012-03-21).
- [2] Albinola, M., Baresi, L., Carcano, M. and Guinea, S.: Mashlight: A Lightweight Mashup Framework for Everyone, *18th International World Wide Web Conference*, pp.10–49 (2009).
- [3] 平山雅樹, 新野朝丈, 児玉公信, 松澤芳昭, 太田 剛: 学生プロジェクトが直面した問題事例とアジャイルによる対処可能性の考察, 情報処理学会研究報告, IS116(3), pp.1–7 (2011).
- [4] Tuchinda, R., Knoblock, C.A. and Szekely, P.: Building Mashups by Demonstration, *ACM Trans. on the Web (TWEB)*, pp.16:1–16:45 (2011).
- [5] Clark, J. and DeRose, S.: XML Path Language (XPath) version 1.0 w3c recommendation, Technical Report REC-xpath-19991116, World Wide Web Consortium (1999).
- [6] Nevill-Manning, C.: Programming by demonstration, *New Zealand Journal of Computing*, Vol.4, No.2, pp.15–24 (1993).
- [7] Wong, J. and Hong, J.: Making mashups with Marmite: Towards end-user programming for the web, *Proc. CHI 2007*, pp.1435–1444, ACM Press (2007).
- [8] Yahoo!: Yahoo pipes, Yahoo! (online), available from <http://pipes.yahoo.com/pipes/> (accessed 2011-10-27).
- [9] Griffin, E.: *Foundations of Popfly: Rapid Mashup Development*, Apress (2008).
- [10] Huynh, D., Mazzocchi, S. and Karger, D.: Piggy Bank: Experience the Semantic Web Inside Your Web Browser, *4th International Semantic Web Conference*, pp.413–430 (2005).
- [11] Huynh, D.F., Miller, R.C. and Karger, D.R.: Potluck: Data Mash-up Tool for Casual Users, *16th International Conference on World Wide Web*, pp.737–746 (2007).
- [12] Yahoo!: Dapper: The Data Mapper, Yahoo! (online), available from <http://open.dapper.net/> (accessed 2011-10-24).
- [13] Lin, J., Wong, J., Nichols, J., Cypher, A. and Lau, T.A.: WEnd-User Programming of Mashups with Vegemite, *Proc. IUI*, pp.97–106 (2009).
- [14] Google: iGoogle, Google (online), available from <http://www.google.co.jp/ig> (accessed 2011-10-27).
- [15] Beemer, B. and Gregg, D.: Mashups: A Literature Review and Classification Framework, *Future Internet*, pp.59–87 (2009).
- [16] Tuchinda, R., Szekely, P. and Knoblock, C.A.: Building Mashups By Example, *13th International Conference on Intelligent User Interfaces*, pp.139–148 (2008).
- [17] Raman, V. and Hellerstein, J.M.: Potter's wheel: An interactive data cleaning system, *VLDB Journal*, pp.381–390 (2001).
- [18] Tatemura, J., Sawires, A., Po, O., Chen, S., Candan, K.S., Agrawal, D. and Goveas, M.: Mashup Feeds: Continuous Queries over Web Services, *ACM SIGMOD*, pp.1128–1120 (2007).
- [19] Brandt, J. and Klemmer, S.R.: Lash-Ups: A Toolkit for Location-Aware Mash-Ups, *Proc. 19th Annual ACM Symposium on User Interface Software and Technology*, pp.79–80 (2006).
- [20] Ennals, R. and Gay, D.: User-Friendly Functional Programming for Web Mashups, *12th ACM SIGPLAN International Conference on Functional Programming*, pp.223–234 (2007).
- [21] Venkatesh, V. and Bala, H.: Technology Acceptance Model 3 and a Research Agenda on Interventions, *Decision Sciences*, Vol.39, No.2, pp.273–315 (2008).
- [22] Gardiner, S., Tomasic, A., Aziz, J.Z.R. and Rivard, K.: Mixer: Mixed-initiative Data Retrieval and Integration by Example, *INTERACT 1*, Vol.6946, pp.426–443 (2011).



新野 朝丈

2010 年静岡大学情報学部情報科学科卒業。2012 年静岡大学大学院情報学研究科修士課程修了。現在、東芝テック株式会社所属。在学中は、情報システム開発に興味を持ち、研究に従事。



平山 雅樹

2010 年静岡大学情報学部卒業。2012 年静岡大学大学院情報学研究科修士課程修了。現在、三栄ハイテックス株式会社所属。



松澤 芳昭 (正会員)

2000 年慶應義塾大学環境情報学部卒業。2002 年慶應義塾大学大学院政策・メディア研究科修士課程修了。2007 年同後期博士課程単位取得退学。2008 年より静岡大学情報学部特任助教。現在、同学部助教。博士（政策・メディア）。オブジェクト指向技術を応用したソフトウェアの設計と開発、情報教育、情報システム技術者育成の研究に従事。日本教育工学会、情報システム学会各会員。



児玉 公信 (正会員)

1978 年東京都立大学人文学部（心理学専攻）卒業。2008 年静岡大学創造科学技術大学院博士後期課程中退。日本石油（株）、北海道大学工学研究科受託研究員、（株）エクサを経て、現在、（株）情報システム総研取締役副社長。技術士（情報工学部門）。博士（情報学）。モデリングに基づく企業情報システムの再構築のコンサル等に従事。著書に『実践ファンクションポイント法』（JMAM）、『UML モデリングの本質』、『UML モデリング入門』（日経 BP）。日本心理学会、経営情報学会、日本人間工学会、ACM 各会員。



太田 剛 (正会員)

1989 年静岡大学大学院電子科学研究科電子応用工学専攻中退。同年静岡大学工学部情報知識工学科助手。1998 年同大学情報学部情報科学科講師。2000 年同大学情報学部助教授。2007 年同大学情報学部准教授。博士（情報科学）。ソフトウェア開発環境の研究に従事。電子情報通信学会、教育システム情報学会各会員。