

オンライン Web ページ作成機構 HPS の実装と そのユーザビリティに関する評価

An Approach to Realizing a Writable Web for the On-line Web Page Creation Mechanism HPS
and Its Usability

西健太郎*1
Kentaro Nishi

松尾徳朗*1
Tokuro Matsuo

田代慎治*1
Noriharu Tashiro

伊藤孝行*1
Takayuki Ito

新谷虎松*1
Toramatsu Shintani

*1名古屋工業大学大学院 工学研究科 情報工学専攻
Graduate School of Engineering, Nagoya Institute of Technology

WWW has developed rapidly, and it is becoming easy to make personal web sites. In general, we create and edit web pages by using a HTML authoring software or writing HTML source codes in a text editor. Then, we need to upload them to a web server. When we make and build our own web pages by using the existing tools, it takes a lot of time and effort to complete the necessary tasks. In this paper, we propose a method for realizing a writable web page that is a home page authoring support system in which we directly edit and make web pages on a web browser. Current experimental results demonstrate that our system can effectively support novices to create their web pages. Also, we show real-world applications that effectively utilize our system.

1. はじめに

近年、インターネットの汎用化とともに、WWW が急激に発展している。それに伴い、Web サーバの数も増加し、個人用 Web ページをもつ人が増えている。現在、Web ページの作成および編集を行う手段として、HTML オーサリングソフトウェア [2] などの利用や、テキストエディタでの HTML ソースの記述などがあげられる。しかし、Web ページ作成の際、ユーザには様々な負担がかかる。具体的に、前者は、ソフトウェアのダウンロードやインストール、および利用方法の習得などが負担となり、後者は、HTML に関する初心者にとって、HTML ソースコードの記述などは、困難となる作業である。また、前者および後者の共通の問題として、サーバからのファイルのダウンロード、HTML ファイルの編集、アップロードといった一連の作業が、煩雑な作業であると考えられる。既存に、ブラウザ上から Web ページの作成が可能で、レンタルホームページサービスを行っているサイトの、Web ページ作成支援ツール [1] などもある。しかし、決められたテンプレートに沿った形式的な Web ページしか作成できないために、ユーザは Web ページのレイアウトが自由にできないという問題があげられる。以上のような既存の Web ページ作成における問題点を解決するために、我々は、文献 [4] において、既存の Web ブラウザ上でオンライン編集可能な Web ページ作成編集支援システム、HPS (Home Page Stickies) を試作した。本論文では、HPS の評価を行い、HPS を利用した実用的なアプリケーションを提案する。

HPS は、任意の Web ブラウザ上から Web ページの作成ができることが大きな特長である。HPS を利用できる Web ページの一部のフレームに、HTML ファイルを表示させると、ブラウザ上から直接テキストや画像などを貼り付けることが可能となる。また、貼り付けたテキストや画像は、自由に位置を変更することができ、レイアウトをすることが可能である。また、リンクの設定、表の書き込みなども可能である。また、

HPS は、サーバ側で動くシステムであるため、ユーザがシステムを利用するにあたって、専用のアプリケーションを用意する必要がない。Web ブラウザからテキストや画像を貼り付けると、即座に Web ページとしてサーバに保存することができる。以上により、HTML ファイルのアップロードやダウンロードといった煩雑な作業を省くことができる。

関連研究として、Wiki Wiki Web [3][6] があげられる。Wiki Wiki Web とは、すべてのユーザが編集可能であり、また、簡単に相互リンクを貼ることができるような Web ページの集合をサポートするシステムである。Wiki Wiki Web は、既存の Web ページを、マークアップ言語を用いて、Web ブラウザ上から誰でも編集可能であるという特徴を持つ。Web ブラウザ上から Web ページを作成および編集できるという点で、HPS と関連していると言える。しかし、テキストベースの Web ページを対象としている点で、Wiki Wiki Web は表や画像などを含んだ Web ページの作成および編集には適しているとは言えない。一方、HPS では、画像や表などを含んだ Web ページの作成も可能である。

また、本アプローチの特筆すべき点は、既存の Web ページへの書き込みも可能であることがあげられる。さらに、簡単なマウスアクションによる Web コンテンツのレイアウトを可能にしたことにより、Web ページの作成および編集にかかる時間的なコストを削減することができる。

本論文では、2 章では、HPS の特長と構成について述べる。次に、3 章では、HPS の評価実験、考察を行い、4 章で、HPS を利用したアプリケーションについて詳しく説明する。最後に 5 章で、本論文をまとめる。

2. Web ブラウザを利用したオンライン編集可能な Web ページ作成編集支援システム

2.1 システムの特長

HPS の最大の特長は、任意の Web ブラウザ上から直接 Web ページを作成することが可能となることである。具体的に HPS が提供する機能を以下に示す。

- オブジェクトの貼り付け
ブラウザ上から、テキストや画像、表などといったオブ

連絡先: 西健太郎, 名古屋工業大学大学院 工学研究科 情報工学専攻, 〒466-8555 名古屋市 昭和区 御器所町 名古屋工業大学, TEL:(052)733-6550, FAX:(052)735-5584, kentaro@ics.nitech.ac.jp

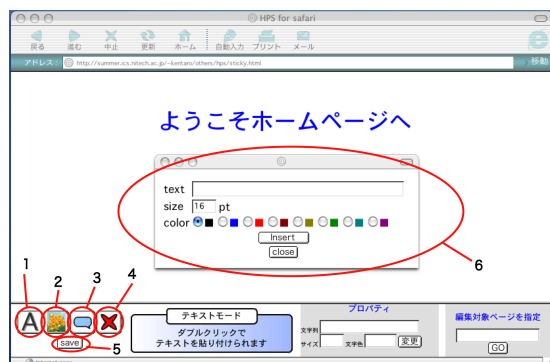


図 1: システムの実行例

ジェクトを、その属性を指定することにより、Web ページ上の任意の位置に貼り付けることができる。

- オブジェクトの移動
貼り付けた全てのオブジェクトは、マウスドラッグにより、簡単に位置を変更することができ、レイアウトが容易である。
- オブジェクトの編集
貼り付けたオブジェクトを選択し、属性を変更することで、容易に指定したオブジェクトを編集することができる。
- 別ページへのリンクの作成
リンク先を指定することで、別ページへのリンクを作成することができる。既存の Web ページへのリンク作成のほか、ユーザがリンク先のページを新しく作成することも可能である。
- 付箋形式のコメントの貼り付け
付箋形式のコメントを貼り付けることができる。テキストや画像のオブジェクトなどと違い、コメントは、Web ページ訪れる全ての閲覧者が貼り付けることができる。
- 既存の Web ページの編集
編集したい HTML ファイルに、書き込み権限がある場合に限られるが、HPS を利用可能となる Web ページの一部のフレームに、編集したいページを表示させることで、既存の Web ページの編集が行えるようになる。頻繁に書き換えられる Web コンテンツを、HPS の機能を用いて書き換えることが可能である。

HPS を利用する際に、ユーザが必要とするソフトウェアは Web ブラウザのみであり、専用のオーサリングソフトウェアは必要としない。また、Web ブラウザ上から即座に Web ページが完成するため、Web サーバとの HTML ファイルのダウンロードやアップロードの必要はない。さらに、FTP の知識や HTML の記述を一切必要としないため、ユーザにかかる負担を大幅に減らすことが可能である。

図 1 に HPS の実行例を示す。図 1 は、HPS を利用するユーザが、Web ページにテキストを書き込む例である。図 1 の 1 は、テキストを書き込むためのボタン、図 1 の 2 は画像を貼り付けるためのボタン、図 1 の 3 は付箋形式のコメントを貼り付けるボタン、図 1 の 4 はオブジェクトを消去するボタン、図 1 の 5 は Web ページを保存するためのボタンである。テキストを書き込む場合、ユーザは、まずテキストを書き込むためのボタン (図 1 の 1) を押し、次に、テキストを書き込みたい位置を、ダブルクリックすることで指定する。ダブルクリックに

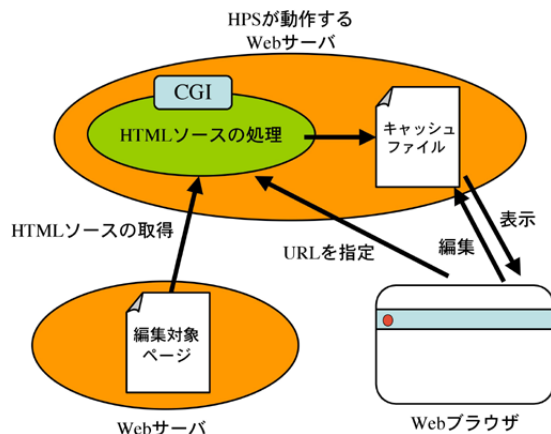


図 2: ファイルキャッシュによる既存のページの編集

より表示された、テキストを書き込むためのウインドウ (図 1 の 6) から、書き込む文字列、文字色、フォントサイズを指定し、Web ブラウザ上から書き込むことができる。画像や表を貼り付ける際も、テキストを書き込む際と同じように、画像のパスや、表の行数、列数などを指定することで、オブジェクトとして貼り付けることができる。ユーザは、オブジェクト貼り付け後は、マウスでオブジェクトを選択し、表示したい位置までオブジェクトを移動することができるので、Web ページのレイアウトを変更することも容易である。ユーザは、オブジェクトを貼り付けた後、「save」ボタン (図 1 の 5) を押すことによって Web ページを保存 (作成) することができる。

また、HPS は、ユーザに書き込み権限の与えられていない、既に WWW 上で公開されている Web ページについても、編集をしたい HTML ファイルをキャッシュすることにより、編集を行うことができる機能も提供している。この流れの詳細は、2.2 章で詳しく説明する。

2.2 ファイルキャッシュによる既存のページの編集

ここでは、既に公開されている Web ページの、ファイルキャッシュによる編集方法について詳しく説明する。流れを、図 2 に示す。HPS を利用可能になる Web ページから、ユーザが編集をしたいページの URL を指定すると、HPS の一部を構成している CGI スクリプトが、その URL にアクセスする。HTTP アクセスにより、与えられた URL の HTML ファイルのソースを読み込む。その際、HTML ソースに記述されている、リンク先の URL、および画像のパスなどを、相対パスで記述されている箇所を、絶対パスへと書き直す。以上のようにファイルをキャッシュし、ユーザにファイルへの書き込み権限を与えることで、Web ページの編集を実現している。システムは、キャッシュしたファイルの URL をユーザに示し、今後、ユーザが引き続き編集を継続したい場合は、その URL へアクセスすることで編集を行うことができる。

2.3 実装方法

オブジェクトの貼り付けは、JavaScript のイベントハンドラを利用して行う。オブジェクトは、すべて <div> タグにより構成される。ユーザがオブジェクトの貼り付けアクションを起こすと、HPS は JavaScript により、対応した <div> タグを生成する。オブジェクトの移動、編集の場合は、DOM (Document Object Model) を利用しタグにアクセスし、CSS (Cascading Style Sheets) を操作することによってオブジェクトを Web ブラウザ上に書き出す。テキストを書き込む場合を例に挙げる

自己紹介のページ

顔写真

氏名	○○○○
年齢	○歳
血液型	○型
生年月日	○年○月○日

50字程度の自己紹介文

図 3: 作成してもらった Web ページのテンプレート

と、ユーザがアクションを起こした場合、そのアクションの起きた座標、書き込むテキスト、テキストのフォントなどを JavaScript で実装された機構が受け取る。次に、受け取った情報をもとにして新たに<div>タグを作成し、<div>タグの DOM プロパティを操作し、ユーザが入力した文字列を<div>タグの内部に挿入する。その後、ユーザのフォントの指定や文字色の指定をもとに、テキストのスタイル属性の変更を行う。以上を利用することで、オブジェクトの貼り付けを実現している。

Web ページの保存は、CGI によりオブジェクトのデータを外部 JavaScript ファイルとして書き出すことにより行われる。HPS では、JavaScript によって HTML 上に書き出された隠しフォームにオブジェクトのデータが保管されている。オブジェクトのデータには、オブジェクトの個数、オブジェクトの貼り付けられている座標、オブジェクトの高さや幅、オブジェクトの内容などが含まれる。ユーザが「save」ボタンを押すと、オブジェクトのデータが CGI に送信され、CGI は、HTML ファイルに読み込むための外部 JavaScript ファイルを作成する。この外部 JavaScript ファイルは、オブジェクト用の<div>タグを HTML ファイルに書き出し、DOM や CSS を利用した、オブジェクトを書き出すためのファイルである。

以上のようにして作成された JavaScript ファイルを、HTML ファイルが読み込むことによって、オブジェクトを保存、つまり、Web ページの保存をしている。

3. 実験と考察

本章では、HTML および FTP に関して全く知識が無い、初心者を対象とした実験を行った。実験の内容、および方法を以下に詳しく述べる。

3.1 実験

小学生 29 人を被験者として実験を行った。小学生には、図 3 に示すような、自己紹介の Web ページを作成させた。作成させる Web ページは、タイトル、顔写真、表組にした個人の情報、および 50 字程度の自己紹介文からなるページである。顔写真や、自己紹介文の内容などは、あらかじめ用意しておいた。また、小学生には、システム的使用方法について説明を行い、実験中わからないことがあった場合は、指導者に質問できるものとした。実験の際、システムの利用方法、また使用した計算機の利用方法などいくつかの質問はあったものの、小学生は、平均約 18 分 16 秒で、Web ページ作成できることが確認できた。以上の評価より、Web ページの作成において、また、HTML や FTP に関する知識を持たないユーザであって

も、Web ページを作成可能であることが確認できる。

3.2 実験からの考察

評価実験より、HPS を利用することにより、Web ページ作成の際のユーザにかかる負担を削減できることが確認できた。従来の HTML オーサリングツールでは、ユーザが Web ページ作成時に描いていたイメージと、完成時のイメージが一致する、「WYSIWYG」という性質が利点としてあげられている。HPS では、Web ブラウザ上から直接 Web ページを作成するという新しい発想から、従来の HTML オーサリングツールよりも、高い性能の WYSIWYG を可能にしたことによって、ユーザのレイアウトを決めるという負担を削減できたと考える。また、HPS では、従来の Web ページ作成および編集の一連の作業である、HTML ファイルのダウンロード、編集、アップロード、Web ブラウザ上で Web ページの出来映えの確認、という、一連の作業の手間を省き、全てブラウザ上からの操作で Web ページを作成可能としたことで、煩雑な作業が減り、Web ページ作成および編集にかかる時間が削減できた。また、指導者の簡単なアドバイスにより、HTML や FTP に関する知識を持たない小学生であっても Web ページを作成できるということから、HPS が、高いユーザビリティを提供しているということが言える。

3.3 実用化に関する考察

HPS を応用することで、総合学習における小学生の調べ学習および体験学習を対象とした、Web アプリケーションの構築が可能である [5]。総合学習とは、小学生を対象とした教育プログラムの一環として、各学習者が個々に学習目標を定め、問題を解決することで目標を達成する学習プログラムである。ここで、提案したシステムを、総合学習を支援するために実際に使用した。具体的な活動内容は、小旅行のしおりを学習者自らが Web しおりとして作成し、その Web しおりにしたがって旅行を行い、見学や調査を実施することで、学習目標としてたてた問題を解決するという活動である。この総合学習での小旅行は、小学生 5 人 1 グループとなり、公共の交通機関を利用して、学習者が設定した目的地に向かい、自らが設定した計画通り、目的地での見学、調査をおこない、集合場所に戻るといったものである。学習者は事前に、目的地までの路線、乗車時刻、および乗り換え時刻などの旅程を調べ、Web しおりを作成する。従来の体験学習のしおり作りでは、学習者が用紙に旅程を記述し、指導者が確認を行っていたが、紙面上で、学習者が記述した複数の旅程のすべてを確認することは、指導者にとって負担であった。そこで、HPS を用いた実施例では、Web を用いて学習者が Web しおりを作成し、教師が Web ブラウザ上で確認を行う。HPS の付箋形式のコメント貼り付け機能により、学習者と指導者とのインタラクティブ総合学習が可能である。

4. HPS を利用したアプリケーション

HPS を応用した Web アプリケーションとして、レポート作成/採点支援システムを提案する。指導者が学習者に課すレポートは、紙面やメールなどによる提出形式がある。レポートの種類には様々なものがあげられるが、ここでいうレポートは、その採点結果を学習者に通知する必要があるものとする。また、レポートを課す者を指導者とし、課せられる者を学習者とする。

4.1 問題提示型レポート

本章では、問題提示型レポートの作成/採点支援の方法について、詳しく説明を行う。問題提示型レポートとは、学習者



図 4: 採点結果ページの一例

が列挙された問題について解答を行い、それぞれの問題に対する解答の、正解／不正解により算出された合計点が採点結果となるレポートのことである。ここで、提出するレポートの形式が、紙面である場合は、指導者が紙面上に、解答の正解／不正解（以降、丸付けとする）の明示、および不正解の理由などのコメントの記述などを行い、レポートを提出した学習者へ返す。しかし、紙面上で採点を行う場合、レポートの量が膨大になると、指導者にかかる負担は大きい。一方、レポートの形式がメールである場合は、学習者の提出したレポートの解答に丸付けをすることは難しい。よって、問題提示型のレポートの提出形式がメールであることは少ない。ここで、HPS を利用し、Web 上でレポートを作成することで、指導者にかかる負担を軽減する。具体的には、指導者が問題提示型のレポートを Web (HTML ファイル) 形式で作成し、URL を学習者に通知しておく。学習者は HPS を利用し通知された URL にアクセスし、レポートのページに解答を書き込む。解答後、学習者は、指導者に対し、編集後のページとして新しく作成された HTML ファイルの URL を提出する。指導者は、提出された URL から、学習者の解答を閲覧することができる。また、指導者は、「○」や「×」の画像を用意しておき、学習者が作成したページに直接貼り付けることで、レポートの丸付けが可能である。レポートの採点結果のページの一例を、図 4 に示す。

4.2 記述型レポート

本章では、記述型レポートの作成／採点支援の方法について、詳しく説明を行う。記述型レポートとは、レポートの問いに対する解答が一つに決まっておらず、学習者が自由に解答を記述できるレポートのことである。具体的には、あるプログラムを作成し、そのソースコードを示せ、などというレポートである。記述型レポートは、問題提示型とは違い、指導者の、レポートを提出した学習者に対する採点は、主として丸付けよりも、レポートの内容に関するコメントや、学習者へのアドバイスが多い。例として、上述した、プログラムのソースコードをレポートとして作成／採点する場合を考える。指導者によるレポートの作成、および学習者によるレポートの解答までの手順は、4.1 章で述べた通りである。その後、指導者による採点が行われる。プログラムのソースに誤り、または改善すべき点があった場合に、指導者が、その修正箇所直接文字列やコメントを挿入することにより、場所を特定して、学習者に示すことが可能である。メール形式のレポートでは、指導者がそのメールに対して返信することで、レポートに対するコメントは可能である。しかし、メールでは、ソースコードの間に空行を挿入

し、そこにコメントや、見本となるソースコードを挿入することになり、指導者、学習者の両者にとって見づらいものとなる。

HPS を利用することの利点として、Web 上での議論が可能であることがあげられる。レポートは HTML ファイルであることから、Web 上で公開できる形式となっており、そのページ上で、指導者、および複数の学習者が文字列、およびコメントを貼り付けることにより、議論を行うことが可能である。Web 上で、指導者、および学習者が議論を進め、レポートに手を加えていけば、議論が進むたびに成長していくようなレポートを作成することができる。また、紙面やメール形式でのレポートでは、複数の指導者による採点は難しいものである。具体的には、レポートを複製するという負担がかかることや、レポートに対するコメントやアドバイスが、重複する可能性があるという問題点がある。一方、HPS では Web 上で動的にレポートが採点されるため、複数指導者による採点において、以上のような問題を解決することが可能である。

5. おわりに

本論文では、従来の Web ページ作成および編集の際、ユーザに必要な一連の作業の負担であるという問題を解決するための、オンライン Web ページ作成機構について説明した。HPS を利用し、Web サーバからの HTML ファイルのダウンロード、Web サーバへの HTML ファイルのアップロード、および、Web ページの出来映えの確認などの負担が軽減されることを述べた。次に、小学生を対象とした評価実験を行い、HTML や FTP に関して知識を持たないユーザでも、HPS を利用することにより、Web ページを作成できることを確認し、HPS の有効性を示した。また、HPS を利用した実用的なアプリケーションとして、レポート作成／採点システムを提案した。このアプリケーションでは、紙面でのレポートの形式に比べ、作業にかかる時間的コストが少なく、また、メール形式の場合に比べ、採点結果が見やすいことなどを挙げた。さらに、Web ページ上での議論により、レポートを成長させることができることを利点として述べた。

参考文献

- [1] goo 簡単ホームページ
<http://members.goo.ne.jp/>
- [2] Home Page Builder
<http://www-6.ibm.com/jp/software/internet/hpb/>
- [3] wiki wiki web
<http://c2.com/cgi/wiki/>
- [4] 西健太郎, 田代慎治, 伊藤孝行, 新谷虎松, "ブラウザをインターフェースとした簡易 Web ページ作成・編集支援システム," 第 66 回情報処理学会全国大会論文集, Vol.4, pp.529-530, 3W-3, 情報処理学会, 2004
- [5] Masaaki Tanaka, Tokuro Matsuo, Noriharu Tashiro, Kentaro Nishi, Takayuki Ito, Toramatsu Shintani, "An Implementation of Web-Based Interactive Integrative Learning Supporting System," International Conference on Artificial Intelligence(IC-AI), Las Vegas, Nevada, USA, 2004
- [6] Bo Leuf, Ward Cunningham: 『Wiki Way』, SOFT BANK Publishing, 2002