

筑波大学大学院博士課程

システム情報工学研究科修士論文

Web 文書の構造編集に関する研究

小野田 紗智

(コンピュータサイエンス専攻)

指導教官 田中 二郎

2005年1月

概要

我々は、Web 文書と、そのリンク構造を編集する手法として「リンクの展開」と「畳み込み」という手法を提案し、それを用いたツールとして「Javara」を実装している。リンクの展開とは、リンク先の文書をリンク元のアンカー位置に挿入する操作である。畳み込みは、任意の範囲の Web 文書を新しいページとして切り出し、そのページへのリンクアンカーと置き換える操作である。Javara は、リンクの展開と畳み込みを利用して Web 文書の構造を編集するツールである。

従来の Javara では、Web 文書の編集機能の実装が展開・畳み込み・ズームのみであり、文書の移動・追加・削除といった編集を行うためには機能が不足していた。また、一度操作を行うと元に戻せない、展開・畳み込みを行う様子が動作を見て直感的にわからないといったインタフェース上の問題があった。

そこで本研究では、新しく JavaraII を設計した。編集機能についての議論を行い、従来の Javara において不足していたコピー・ペースト・UNDO・保存といった機能を追加した。また、Javara の編集特性を考慮したアノテーション機能を提案し、実装した。インタフェースにおける問題についても議論し、展開操作の確認や、展開・畳み込み動作のアニメーション表示といったインタフェースの改善を行った。

また、JavaraII における Web 構造上の問題点を考察した。まず、Web の構造について議論するためにリンクの分類を行った。次に JavaraII で起こる問題として、ページ同士のリンクの構造によって、リンクの展開を行うことで同じ内容の文書が重複する問題を取り上げ、考察を行った。その上で今後のシステム構築に対する方針を示した。

目次

第1章	序論	1
1.1	Webサイトの構築と再構築	1
1.1.1	Webサイトの構築	1
1.1.2	Webサイトの再構築	1
1.2	研究の目的と本論文の構成	2
第2章	Web文書の構造編集ツール Javara	3
2.1	inlineLink	3
2.2	Javara とは	4
2.3	リンクの展開と畳み込みによる編集	4
2.3.1	リンクの展開	4
2.3.2	畳み込み	5
2.4	Javara を用いた編集方法	7
2.4.1	ボトムアップ方式	7
2.4.2	トップダウン方式	7
2.4.3	その他の編集方法	8
2.5	Javara の利点	8
2.6	Javara の問題点	8
第3章	JavaraII の設計	10
3.1	JavaraII の設計方針	10
3.2	編集機能の強化	11
3.2.1	カット・コピー&ペースト機能	11
3.2.2	UNDO 機能	11
3.2.3	保存機能	12
3.2.4	アノテーション機能	12
3.3	インタフェースの改善	12
3.3.1	展開操作の確認	12
3.3.2	アニメーション表示	13
3.3.3	左右のマウスボタンの使い分け	13
3.3.4	メニュー表示	13

第 4 章	JavaraII の実装	15
4.1	インタフェース部	15
4.1.1	リンクの展開	15
4.1.2	畳み込み	17
4.1.3	カット・コピー機能	19
4.1.4	ペースト機能	19
4.1.5	UNDO 機能	20
4.1.6	REDO 機能	20
4.1.7	アノテーション機能	20
4.1.8	保存機能	21
4.1.9	インタフェースの改善	21
4.2	サーバ部	22
4.2.1	保存機能	22
4.3	JavaraII の利用法	23
4.3.1	長大な Web 文書の切り分け	23
4.3.2	アノテーション	23
第 5 章	Web 構造上の問題と解決手法	25
5.1	リンクの分類	25
5.1.1	リンク元に着目した分類	25
5.1.2	リンク先に着目した分類	26
5.2	内容の重複問題と解決手法	28
5.2.1	内容の重複が起こるリンク	28
5.2.2	解決手法	31
第 6 章	関連研究	32
6.1	ブラウザ上で編集を行うツール	32
6.2	構造編集ツール	32
第 7 章	結論	34
7.1	結論	34
7.2	今後の課題	34
	謝辞	35
	参考文献	36

目次

2.1	inlineLink	3
2.2	リンクの展開の例	5
2.3	畳み込みを行なう前	5
2.4	畳み込む範囲を選択	6
2.5	ウィンドウプロンプトでアンカー名を記入	6
2.6	選択したところが新しいアンカーと置き換わる	6
2.7	ボトムアップ方式で編集する例	7
2.8	トップダウン方式で編集する例	8
4.1	実装	15
4.2	original anchor	16
4.3	open anchor	16
4.4	リンクの展開をした状態のアンカー	16
4.5	remove-frame を実行	17
4.6	畳み込みのアンカー入力ウィンドウ表示	18
4.7	畳み込んだ状態のアンカー	18
4.8	カットする文字列を選択	19
4.9	カットを実行	19
4.10	ペースト位置を選択	20
4.11	ペーストを実行	20
4.12	アノテーションを入力するアンカー	21
4.13	メニュー表示	22
4.14	長大な Web 文書の切り分け	23
4.15	アノテーションの対象となる文字列を選択	24
4.16	アノテーションの入力	24
4.17	キーワードとアノテーションがリンク関係になる	24
5.1	同じサイト内のほかのページへのリンク	26
5.2	同じサイト内の他のページ的位置を示すアンカーへのリンク	27
5.3	同じページ的位置を示すアンカーへのリンク	27
5.4	他のサイト内のページへのリンク	28
5.5	複数の個所から一つのページにリンクがあるとき	29

5.6	相互にリンクしているとき	29
5.7	ページがループ状にリンクしているとき	30
5.8	ループ状のリンクの展開	30
6.1	Wiki の編集画面	32
6.2	DENIM の編集画面	33

第1章 序論

1.1 Webサイトの構築と再構築

1.1.1 Webサイトの構築

近年のネットワーク環境の著しい改善と計算機の普及にともない、World Wide Web（以下Webとする）が情報源としての役割を大きく担うようになってきた。World Wide Web サイト（以下Webサイトとする）が増えるにつれ、閲覧者に理解しやすいようにWebサイトを適切に構成することの重要性が増している。

Lynchは「Webサイトのデザインは、主にトップページとそれぞれのコンテンツの構造のバランスを取る問題である。デザインの目的は、ユーザにとって自然によく構成されており、Webの利用を阻害したり誤って導いたりしないような階層構造を構築することである」[1]と述べている。

Webサイトは一つのページのみで構成されるのではなく、いくつかのページに分割され、それぞれのページがリンクすることで構成されている。制作者は、Webサイトを構築する際、記述する内容とページのリンク構造をそれぞれ設計する。

特に文書をベースとするWebサイトは、文書の意味内容の関係を表す文書構造と、分割されたページのリンク構造をそれぞれもっている。この二つはサイトの閲覧者にとって密接に関係している。文書構造とリンク構造が整合していない場合、サイトの閲覧者は、サイトの内容の理解が難しくなる。そのため、サイトを構築する場合、文書構造と、リンク構造を十分考慮する必要がある。

1.1.2 Webサイトの再構築

たとえ綿密に文書構造とリンク構造を考えられて作ったWebサイトでも、その後の改変で全体構成が不適切になることがある。頻繁に更新を行うWebサイトでは、内容の追記や更新を繰り返す結果として、最初に設計した構成が、現在のWebサイトの内容に合致しないものになる可能性がある。その場合、内容の重複や分散が起これ、Webサイトの閲覧者にとって理解しづらいものになってしまう。

文章の間違いなど、細部の間違いであれば、各ページの修正を行えばよい。しかし、サイトの内容の理解が難しい原因が、各ページに記述された文書内容の分割方法やリンクの構成にある場合には、Webサイトの構造自体を変更して再構築する必要がある。

多くのWebサイト制作者は、いったん作成したWebサイトの再構築を行っている[2]。ま

た、閲覧者の経路や閲覧時間を、詳細に解析することによって制作者がサイトの適切な構成を発見するのを支援する研究もなされている [3, 4]。しかし、Webサイトを再構築する際の編集方法や編集ツールについてはあまり考慮されていないのが現状である。

1.2 研究の目的と本論文の構成

本研究では、青木らによって提案された Web 文書の構造編集ツール Javara [5, 6] を基に Web 文書の再構築を行うツールを作成する。Web 文書とは、ある内容について書かれた複数の Web ページであり、それらがリンクによって関連付けられているものを指す。従来の Javara における問題点を考察し、問題点を改善した新しいシステムとして Javara II を設計・実装する。

本研究の目的として、第一に、文書構造の管理を Web 上で行うことのできる、Web 文書の構造を編集するツールの作成が挙げられる。これは、従来の Javara の目的と共通する。第二に、Web 文書構造の適切な表示・編集方法を示すことが挙げられる。Javara では、他のオーサリングツールであまり取り扱われない Web 文書の構造を扱うため、文書構造の表示方法や編集方法について考察する必要がある。

本論文の構成は以下のとおりである。まず、第 2 章で Web 文書の構造編集ツール Javara について述べる。第 3 章で従来の Javara の問題点と、それを改善した Javara II の設計について、第 4 章で Javara II の実装について述べる。第 5 章では、Javara II における Web 構造上の問題点とその解決方法を提案する。第 6 章で関連研究について議論を行い、最後に、第 7 章で結論を述べる。

第2章 Web文書の構造編集ツールJavara

2.1 inlineLink

Javara の前身として、文書内挿方式による Web 閲覧システム inlineLink[7, 8] がある。

inlineLink は三浦らによって提案された Web ブラウザを利用した文書内挿による Web 文書の閲覧を行うシステムである。通常、Web ブラウジングを行う場合、1つのページは1つのウィンドウに表示される。この場合、閲覧者がリンク先文書を参照しようとしてアンカーをクリックしたときには元の文書が消えてしまい、戻るためにはいちいち“戻る”ボタンを押す必要がある。しかし、この“戻る”ボタンを押すときには読んでいる文章から目を離すので、効果的に閲覧できない。

そこで、リンク先の文書をリンク元文書に埋め込む閲覧手法 (inlineLink 手法) を提案している。inlineLink 手法では、図 2.1 のように、アンカー位置にリンク先文書を埋め込んで表示する。そのため、リンク元の文章とリンク先の文書をつながりのある状態で閲覧できる。また、リンク元文書に戻る“Back”ボタンが不要になる。評価実験の結果、マウスの移動操作量を軽減する効果があることを確認している。マウスの移動操作量が軽減されることにより、文章に集中しやすいと考えられる。inlineLink は、既存の Web ブラウザ上で文書内挿機能を実現するため、JavaScript による Dynamic HTML[9] と呼ばれる動的なページ変更の仕組みを利用している。

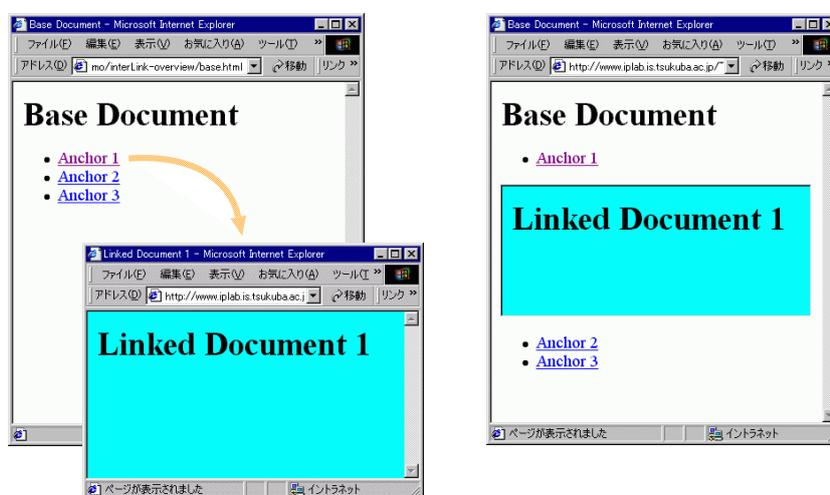


図 2.1: inlineLink

2.2 Javara とは

Javara は青木らによって提案された、リンクの展開と畳み込みによる編集手法を用いて実装された Web 文書の構造編集ツールである [5, 6]。

Javara では、特に一度構築された Web 文書を再構築することを目的としている。ここでの Web 文書の再構築というのは、文書の構造とリンク構造が整合しない、文書が長くて読みづらいといった理由で、ユーザが閲覧し難い不適切な構造となってしまったサイトを、構築しなおすことを指す。

Javara における Web サイトの再構築を行なう方針として、

1. 複数の文書を効率よく扱うために、リンク先とリンク元の文書を同時に把握できるようにする
2. 文書の内容に着目した編集を行なうために、HTML ファイルやタグを意識させない

ということが挙げられている。

Javara は、編集を Web ブラウザを通して行うために、JavaScript による Dynamic HTML 技術を用いて実装している。

2.3 リンクの展開と畳み込みによる編集

これらの方針を実現するために、Javara では、リンクの展開と畳み込みという編集手法を用いている。

2.3.1 リンクの展開

リンクの展開は、リンクアンカーをクリックすることで、アンカー位置にリンク先の文書を挿入する操作である。ここでは inlineLink における文書内挿手法を用いている。

図 2.2 は、リンクの展開を行う様子である。リンクアンカーの位置にアンカー先の文書が挿入されている。

従来の Web ブラウザ上でリンク先の Web ページを開く場合、アンカーをクリックすると、アンカー先の文書が同じウィンドウに表示される。このときリンク元のページは消えてしまい、もう一度元のページを閲覧するためにはブラウザの「戻る」ボタンを押す必要があった。あるいは、メニューから「Open Link in New Window」を選択し、新しいウィンドウにアンカー先の文書を表示する方法もあるが、しばしばウィンドウの表示が重なりウィンドウの移動やリサイズが必要になることがある。また、ウィンドウ間で視点を移動することがユーザの負担となる。

リンクの展開を用いることで、リンク元のページが残るのでユーザはリンク先とリンク元を同時に閲覧することができる。また、アンカーのあった位置にリンク先の文書が展開されるので、ページ同士の関連を把握することが容易になる。リンク先の文書は同じウィンドウ

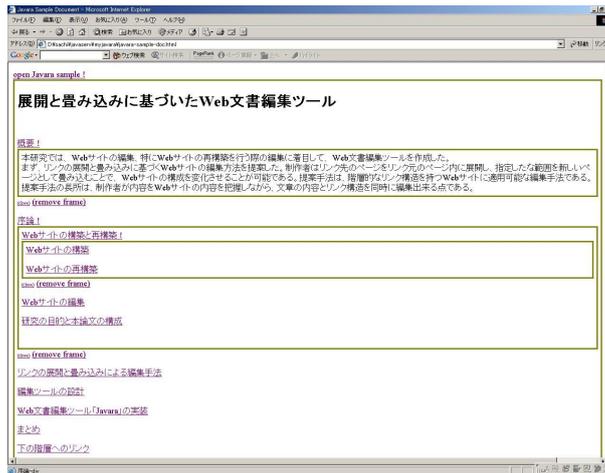


図 2.2: リンクの展開の例

内に表示されるためユーザが視点を移動する負担が軽くなる。

2.3.2 畳み込み

畳み込みは、Web ページの任意の範囲を、新しいページとして生成し、また新しいページへのリンクアンカーを元のページに生成する操作である。

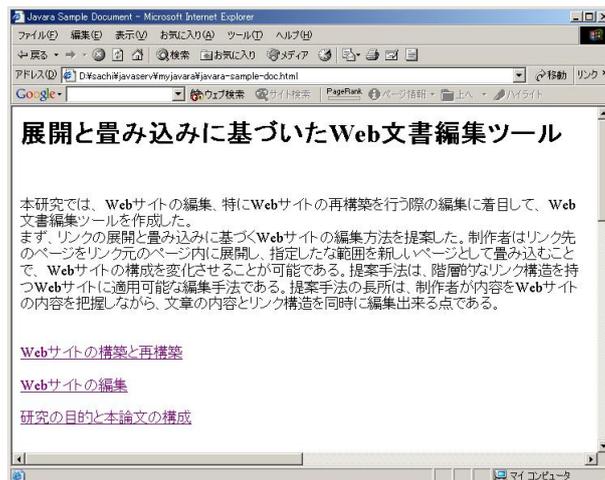


図 2.3: 畳み込みを行なう前

図 2.3 ~ 図 2.6 は畳み込みを行う様子である。図 2.3 は畳み込みを行う前の状態である。畳み込み範囲の文書を選択し(図 2.4)畳み込みの操作を行うと、コマンドプロンプトが表示される(図 2.5)。コマンドプロンプトに新しいアンカー名を記入し OK をクリックすると図 2.6 の

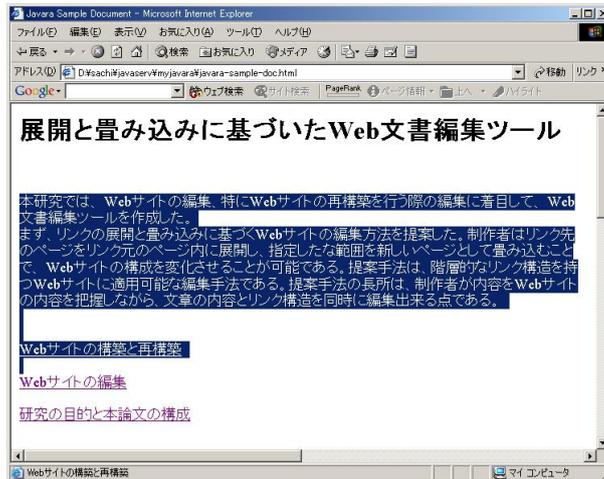


図 2.4: 畳み込む範囲を選択



図 2.5: ウィンドウプロンプトでアンカー名を記入

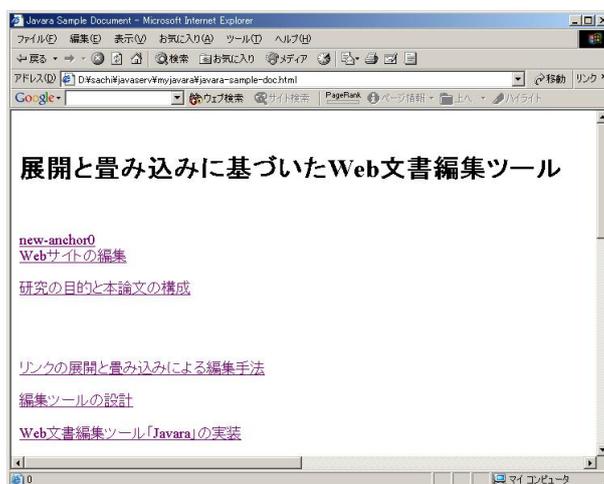


図 2.6: 選択したところが新しいアンカーと置き換わる

ように選択部分の文書が新しいアンカーと置き換わる。

畳み込みを用いることでHTML ファイルやタグを意識することなく Web 文書の再構築が可能となる。

2.4 Javara を用いた編集方法

Javara を用いた Web 文書の編集方法には、以下のような方法がある。

2.4.1 ボトムアップ方式

ボトムアップ方式の編集は、細かく畳み込みを行ない、畳み込んだアンカーに対して畳み込みを行う繰り返して編集する。

最初に Web サイト内のすべての Web ページを展開して一つの Web ページとする。ユーザは、全体を見て内容の重複や流れを確認したら、まず小さな単位で畳み込みを行い、畳み込んだアンカーに対して複数選択して畳み込む。

この方式は、Web サイトの構造を大きく編集しなおしたいときに有効である。細かい範囲からボトムアップで構成していくので、編集しながらサイトの構成を考えることができる。

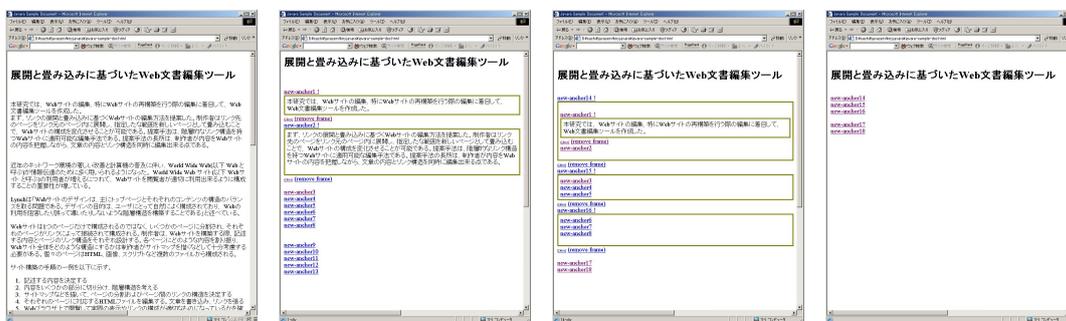


図 2.7: ボトムアップ方式で編集する例

図 2.7 はボトムアップ方式で編集する例である。左から右に編集が進んでいく様子を示している。

2.4.2 トップダウン方式

トップダウン方式は、全体を大きく分けて畳み込み、畳み込んだ内容に対して畳み込みを行う繰り返して編集する。

ボトムアップ方式と同様に、Web サイト内のすべての Web ページを展開して一つの Web ページとする。まずは全体を大まかに切り分けて畳み込みを行う。畳み込みを行った内容に対してさらに大まかに畳み込みを行うことを繰り返す。

この方式は、作り直すサイトの構成がある程度決まっている場合に有効である。大きく内容を分けてから細かく分類するため、文章の全体の流れを把握している必要がある。

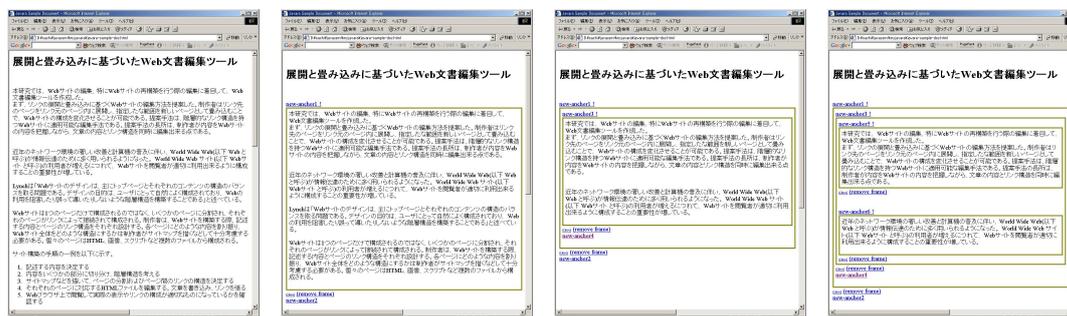


図 2.8: トップダウン方式で編集する例

図 2.8 はトップダウン方式で編集する例である。左から右に編集が進んでいく様子を示している。

2.4.3 その他の編集方法

ボトムアップ方式やトップダウン方式のように大きくサイトの構成を変えずに、サイトを部分的に直したい場合は、編集する部分のリンクのみ展開して編集しなおす方法が考えられる。ページ分割の微調整や、関連項目を別ページとして切り出す作業などの部分的な編集を行うことができる。

また、状況に応じてこれらの方法を組み合わせて編集することも可能である。

2.5 Javara の利点

リンクの展開と畳み込みの手法を用いることで、ユーザは、ファイル構造を意識することなくリンクと文書の階層構造を編集することができる。タグやファイルといった概念を隠蔽できるので、文書内容に注目し、より抽象度の高い編集が可能となる。

また、Web ブラウザを用いて編集を行うため、特別なアプリケーションのダウンロードの必要がない。

2.6 Javara の問題点

従来の Javara では、リンクの展開と畳み込みという2つの手法に着目し、その機能に特化していた。そのため、編集対象となる Web 文書の階層構造を変えることはできるが、文書の位置や順序を入れ替えるといった実質的な編集を行うことができなかった。また、Javara に

は、ブラウザ上で編集を行い、ファイル構造を意識しないでユーザが気軽に使えるという利点がある。しかし、その利点を活かすためのインタフェースに関して、考察の必要があった。そこで、3章において Web 文書の編集に必要とされる機能やインタフェースについて考察を行った [10]。

第3章 JavaraIIの設計

3.1 JavaraIIの設計方針

Javara の問題点を元に JavaraII の設計方針を以下のように定めた。

1. Web 文書編集においては、すでにある文章の位置を移動したりコピーしたりすることが必要となる。文書の位置や順序を入れ替えることによって、不適切な位置への文章の追記や、読み辛い文書を適切に編集しなおすことができるようになる。また、リンクアンカーを含めて文章の位置を移動することで、リンクの展開・畳み込み操作のみでは実現できない、Web の構造の並べ替えが実現できる。
2. 編集作業の一環として、編集経過を閲覧したり、編集内容を見比べる、また、間違えた操作を元に戻す機能が必要となる。編集経過を閲覧することで、間違った操作を行っていないか確認することができる。次回編集する際の参考にもなる。文書の位置や順序の入れ替えといった大きな編集作業を行った場合、その作業の前後を見比べて編集できるとよりわかりやすい編集が行える。間違えた操作を元に戻す機能があると初心者も安心して使いやすい。
3. Javara で編集した Web 文書は、Dynamic HTML によりブラウザ上で表示を変えることで編集した状態を表している。そこでは、表示方法によりリンク構造を表しているだけで、実際の HTML ファイルは存在しないため、後から編集した状態を再現することができない。そこで、Javara での編集状態を HTML ファイルへ変換する必要がある。
4. Javara では、inlineLink で実装されている文書内挿方式と同じ方法でリンクの展開を行う。文書内挿方式による閲覧に向けた Web ページはある特徴を持っている。その特徴を持つ文書を作成する編集機能が Javara の編集に有効性を発揮すると考えられる。文書内挿方式に向く Web ページの特徴は、3.2.4 節で詳しく述べる。
5. 編集作業の一環として、リンク先の Web 文書の内容を確認することが必要となる。リンクの展開を確認段階を経て行うことで編集作業を行う前に、Web ページの閲覧やリンク元とリンク先の文書の間接関係を確認するといった作業を行なうことができる。
6. Javara の利点である抽象度の高い編集を手軽に行える点を活かすために、より手軽な編集方法として、マウスのみでの編集を行えるインターフェースが必要となる。従来キーボード操作で行っていたものをマウスで行うことができるようにすることでより手軽な編集

操作が実現できる。また、リンクの展開・畳み込みの視覚的なフィードバックをユーザーに与えることで、操作のわかりやすさが向上すると考える。

これらの設計方針を元に必要な機能やインターフェースについて考察する。

3.2 編集機能の強化

3.2.1 カット・コピー&ペースト機能

編集作業の一つに文章の並べ替えがある。文章の並びが不適切な場合、文章を並べ替えることで適切な文書に編集する。そこで、文章の並べ替えを行う際に、必要な機能としてカット機能、コピー機能、ペースト機能がある。

カット・コピー&ペースト機能は次のように行われる。最初にカット・コピーを実行したい部分を選択し、カットまたはコピー操作を実行する。

選択されたデータは、システム内部のクリップボード領域に移動（またはコピー）される。その後、ペースト操作を実行するとクリップボードに保存されたデータが、任意の場所へ貼り付けられる。

文章の切り貼りを行うことができると、従来の Javara では実現できなかった文章の順番の並べ替えや削除、複製を実行することができるようになる。それにともない、編集可能な操作が増えるとともにツリー構造のリンクしか実現できなかった展開・畳み込み機能と比べ、同じ個所へ複数のリンクの作成を行えるようになり、ツリー構造のみでないリンク構造を実現できるようになる。また、カット機能を行うことで文章の削除が可能になる。

3.2.2 UNDO 機能

カット・コピー&ペーストのような編集機能が増えるとともに誤った操作を行った際に元の状態に戻す機能が求められる。

UNDO 機能とは、直前に実行した処理を無効にし、その処理を実行する以前の状態に戻す操作、またはそれを可能にする機能である。UNDO を利用できると、間違った操作を行ってしまった後であっても、元に戻すことができるので、特に初心者にとっては安心して作業を行うことができる。

また、間違っ UNDO を行ってしまった場合に元に戻す機能として、REDO 機能がある。REDO 機能があることにより、UNDO を元に戻すだけでなく、UNDO 機能と REDO 機能を併用として、UNDO を行い編集状態を元に戻し、そこからどのような編集を行ったのか REDO を行って編集工程を閲覧するといった利用ができる。

3.2.3 保存機能

Dynamic HTML 技術では、Web ブラウザ上に表示される状態を変更できる。従来の Javara は編集状態をブラウザ上に表示することを行っている。しかし、HTML ファイルに変換して、編集状態を保存することができないため、編集状態を更新してしまうと、編集前の状態に戻ってしまう。Javara を利用するためには、編集状態を HTML ファイルに変換し、保存する機能が必要となる。

Javara では、inlineLink で用いられている文書内挿方式によって、リンク関係にある文書を把握している。そこで、ユーザが編集状態と、HTML ファイルの閲覧時に違和感を感じないように inlineLink 手法による Web 閲覧ができるようにしている。

3.2.4 アノテーション機能

文書内挿方式で Web 閲覧を行う場合、Web 文書が次の特徴を同時に満たす時、文書内挿が有効性を発揮すると考えられる [8]。

- 内挿される文書が短い場合 (例えば、注釈や用語集等)。内挿後、リンク元の文書の多くが画面に残るため、読者はリンク先文書とリンク元文書を同時に閲覧できる。
- 複数の文書が密接な関連を持つ場合。一つの内容が複数の文書に分離している場合 (例として、メーリングリストのアーカイブや掲示板の記事等)。文書内挿は、リンク元文書とリンク先文書のつながりを自然に表現する。

JavaraII では、文書内挿方式による編集を行うため、内挿される文書が短く、且つ、リンク先の文書とリンク元の文書が密接に関係するアノテーションの機能が JavaraII の利点を伸ばすことができる。

アノテーション機能は、ユーザの注目するキーワードと、ユーザの入力した文字列をリンクの関係を持たせることで実現する。

3.3 インタフェースの改善

3.3.1 展開操作の確認

リンクの展開を行い、編集に入る前に、リンク先の文書内容を確認したい場合がある。従来の Javara では、リンク先の文書を確認するためには、展開しリンク元の文書に挿入する必要があり、また、元に戻す操作がなかったためリンク先の文書を確認することが困難だった。UNDO を実装することで元に戻す操作は実現できるが、展開の操作と同時にリンク先の文書をリンク元の文書に挿入してしまうと文書同士の関係を把握しにくい。

そこで、リンク元の文書へ挿入する前にリンク先の文書を枠で囲いアンカー位置に表示することでリンク元の文書とリンク先の文書の区別が付き、文書同士のリンク関係を確認することができるようにする。

枠を表示して確認を行うことが煩わしい場合は確認を行わないように切り替えることができるようにする。

3.3.2 アニメーション表示

従来の Javara のリンクの展開・畳み込みは、各操作を行うと同時に操作の結果が表示される。操作を行ったレスポンスが早いのはいいが、操作と操作の結果の間に何が行われているのかわかりにくい。

そこで、リンクの展開・畳み込みの操作を行う際にアニメーション表示をすることを提案する。リンクの展開を行う際は、リンクアンカーからリンク先の文章が開いていく様子が見えるように、リンク先の文章をアンカー部分から少しずつ拡大していくアニメーションを行う。畳み込みを行う際は、新しく作成したアンカーに畳み込みを行う文章が閉じていく様子が見えるように、畳み込みを行う文章をアンカー部分へ向けて少しずつ縮小していくアニメーションを行う。また、アニメーションを行うと同時に視点の移動を行う。リンクの展開の際は展開先の文章へ視点の移動を行い、畳み込みの際は新しいアンカーへ視点の移動を行うことで、ユーザの焦点を関心がある方向へ自然に向け、次の編集作業へスムーズに移れるようにする。

3.3.3 左右のマウスボタンの使い分け

従来の Javara では畳み込みを行う際の文章の選択を行う際のマウสดラッグと、ズームを行う際のマウสดラッグを、キーボードの操作でモードの切り替えをすることで両方ともマウスの左ボタンを用いて実現していた。

キーボードを用いたモードの切り替えは煩わしい。また、モードの切り替えを忘れることで、文章の選択とズームを誤操作する可能性がある。

そこで、マウスの左右のボタンを区別し、それぞれに機能を割り当てることを提案する。左ボタンのドラッグでは文章の選択を、右ボタンのドラッグでは画面のズームを行う。ズームを右ボタンで行うインタフェースは Java のズームングツールキット Jazz[11] でも用いられている。

マウスボタンの左右を使い分けることで、モード切り替えの煩わしさと、同じボタンを使うことで起こる誤操作を避けることができる。

3.3.4 メニュー表示

従来の Javara でキーボードで行っていた畳み込みの操作を、マウスの右ボタンクリックでメニュー表示することを提案する。メニューに表示する操作は畳み込み・カット・コピー・ペーस्टと、従来キーボード操作で行っていたページの全体表示、元の大きさで表示、ワイドモードへ切り替えの操作とする。

キーボードでの操作は、操作キーを覚えてしまえば簡単に行うことができるが、操作キー

を覚えていない場合、各操作を行うたびに操作キーを確認しなくてはならない。メニュー表示は、見た目にわかりやすく、操作キーを覚えなくても使えるので、キー操作に慣れていない初心者にもわかりやすい。また、新しく作るアンカーの入力操作を行わなければ、メニュー表示を使うことで、キーボードを使わずにマウス操作のみで編集を行うことができる。

第4章 JavaraIIの実装

JavaraIIはWebブラウザ上で動作するインタフェース部とCGIで呼び出されるサーバ部によって構成されている。インタフェース部ではリンクの展開、畳み込みといった編集機能を実装し、サーバ部では編集状態のHTMLファイルへの変換、保存を実装している。

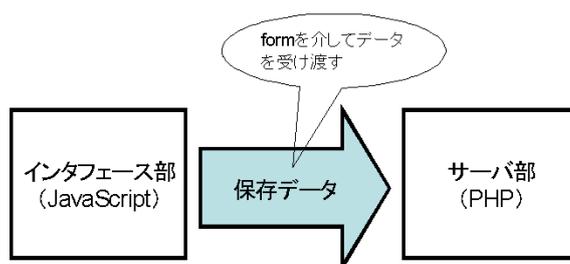


図 4.1: 実装

4.1 インタフェース部

インタフェース部はJavaScriptで実装した。JavaScriptは、既存のブラウザ上で動作するため、ユーザが日常的に使用しているブラウザを用いて編集を行うことができる。

JavaraIIは、JavaScriptプログラムを編集対象となるHTMLファイルへ読み込ませることで動作する。

4.1.1 リンクの展開

JavaraIIで編集するWebページのリンクアンカーは、まず、リンクの展開を行うJavaScriptプログラムを呼び出すアンカー(open anchor)へ書き換える必要がある。一般的なリンクアンカーは図4.2で表される。これを書き換えると図4.3のようになる。最初のWebページのリンクアンカーを書き換えておけば、そのページからリンクされている文書内のアンカーは、プログラムによって動的に書き換えられるので、書き換えておく必要はない。open anchorはspanタグで囲み、一意なID(ここではgoogle)を設定する。

```
<a href="http://www.google.com"> Google </a>
```

図 4.2: original anchor

```
<span id="_google">  
<a href="javascript:inline(' _google','http://www.google.com')">  
  Google </a>  
</span>
```

図 4.3: open anchor

open anchor

open anchor がクリックされると、リンクの展開が行われる。ここでは、open anchor を含む span 要素を書き換える。書き換える内容は、展開した文書を閉じるアンカー (close anchor)、リンク先の文書を表示する div 要素、展開した文書の枠を消去するアンカー (remove-frame anchor) である (図 4.4)。リンク先の文書はスタイルシートによる枠で囲んで表示する。枠の表示の詳細は 4.1.9 節に詳しく述べる。

```
<span id="_google">  
<a href="javascript:unfolding(' _google','http://www.google.com')">  
  Google !</a>  
<div class="bolder" id="_google-div">  
  -----リンク先の文書-----  
</div>  
<a href="javascript:unfolding(' _google','http://www.google.com')">  
  (close)</a>  
<a href="remove_frame(' _google')">(remove frame)</a>  
</span>
```

図 4.4: リンクの展開をした状態のアンカー

div 要素には、HTML ファイルを展開する場合、Inline Frame を用いてリンク先の文書を書き込む。Inline Frame は、通常のフレームと異なり、文書中にオブジェクトとして挿入可能なフレームである。Inline Frame は HTML4.0 Transitional DTD において iframe 要素として定義されている。HTML 文書中において、<iframe src=URI></iframe> と記述すると、任意の URI で指定された文書ソースが Inline Frame に読み込まれる。この Inline Frame は文書ソース取得が目的であるため、画面に表示されないよう非表示に設定する。

一度展開を行ったアンカーを再度展開する場合、アンカー先の文書は Javara 配列に書き込まれている。span タグで設定した ID を元に、Javara 配列から文書を読み出し、div 要素に書き込む。

close anchor

close anchor がクリックされると、設定された ID の span 要素を open anchor へ置き換える。これにより、展開したアンカーを閉じることができる。

remove-frame anchor

remove-frame anchor がクリックされると、枠で囲んでいたリンク先の文書の枠を取り除く。この枠は div 要素をスタイルシートを用いて枠で囲っていたので、div タグを取り除くことで枠が取り除かれる。また、ID 付けをする span タグや、close anchor、remove-frame anchor も同時に取り除く。これにより、リンク元の文書にリンク先の文書が余計なタグを取り除いた状態で書き込まれることになる (図 4.5)。



図 4.5: remove-frame を実行

確認操作を行わないモードの場合、スタイルシートによる枠を表示しないで展開操作を行い、その後、remove-frame 関数を実行することで、リンクの展開を確認する操作を行わずに、リンク元の文書にリンク先の文書が書き込まれる。

4.1.2 畳み込み

畳み込みは、次のように実装した。まず、ユーザは任意の文書を選択する。システム側は、選択した文書と全体の文書の文字列が一致する部分を調べる。その際、文字列以外の情報が影響を与えないように、<> 等の記号を変換 (エンコード) した後で文字列の適合を行う。

選択した文書と全体の文書が適合しなかった場合、エラーを出力して終了する。

選択した文書と全体の文書が適合した場合、ユーザにアンカー名を入力するためのウィンドウを提示する。アンカー名の入力欄には選択文書の先頭文字列半角 30 文字まで、あるいは、文書内の最初のタグに囲まれた文字列があらかじめ入力されている。ユーザはあらかじめ入

力された文字列を編集することも、そのままアンカー名に用いることもできる。

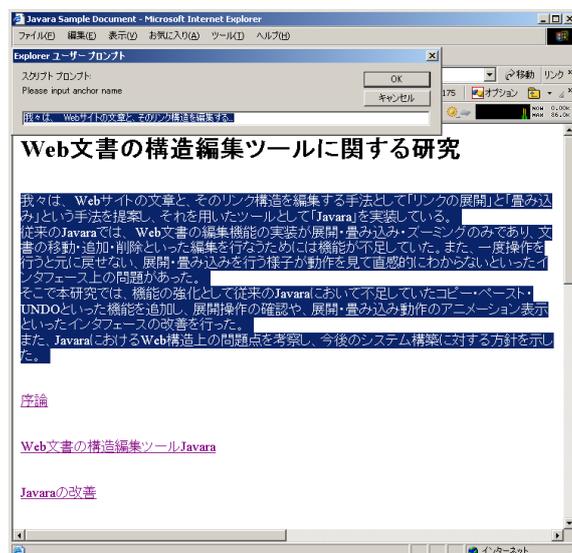


図 4.6: 畳み込みのアンカー入力ウィンドウ表示

ユーザがアンカー名を入力すると、全体の文書に選択した文書が適合した部分を、ID を設定した span 要素に置き換える。span 要素には、入力したアンカー名の open anchor、close anchor、remove-frame anchor と div タグで囲んだ選択した文書を書き込む(図 4.7)。これにより、選択した文書は枠で囲まれて表示される。open anchor、close anchor、remove-frame anchor の動作はリンクの展開で実装したものと同一である。

```
<span id="xx">
<a href="javascript:unfolding('xx',false)">入力アンカー</a>
<div class="bolder" id="xx-div">
-----
選択した文書
-----
</div>
<a href="javascript:unfolding('xx',false)">
(close)</a>
<a href="remove_frame('xx')">(remove frame)</a>
</span>
```

図 4.7: 畳み込んだ状態のアンカー

確認操作を行わないモードの場合、全体の文書に選択した文書が適合した部分を span 要素に置き換えた後に、close anchor をクリックするときに呼び出される関数を実行し、枠に囲まれている文書を open anchor へと置き換える。

リンクの展開、畳み込みで用いられる span 要素の ID は一意に決められており、ID とその ID の示す文書は Javara 配列で管理されている。

4.1.3 カット・コピー機能

カット機能では、まず、ユーザは任意の文字列を選択する(図 4.8)。畳み込み操作と同様に、選択した文書と全体の文書の文字列が適合する場所を調べる。文字列が適合しなかった場合、エラーを出力して終了する。文字列が適合した場合、全体の文書の適合した部分を空文字列と置き換える(図 4.9)。選択した文書は、エンコードして clipboard 変数へ代入する。

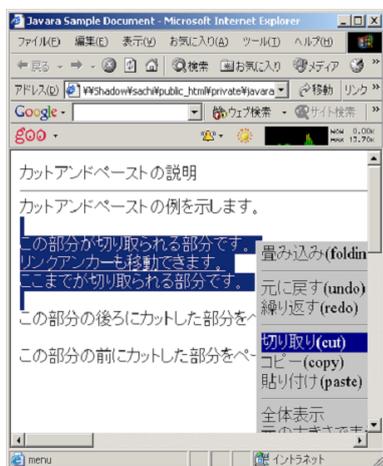


図 4.8: カットする文字列を選択

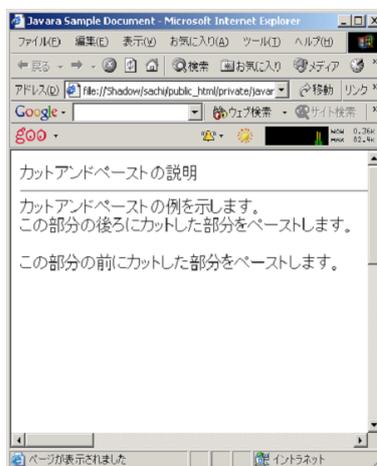


図 4.9: カットを実行

コピー機能では、カット機能の空文字列と置き換えを行わないで、選択した文書をエンコードして clipboard 変数へ代入する。

4.1.4 ペースト機能

ペースト機能を実行する際、ユーザは、文書をペーストしたい場所の前の文書を、マウスドラッグにより選択する(図 4.10)。全体の文書の先頭にペーストしたい場合は文書を選択しない。

ペースト機能が実行されると、まず、clipboard 変数の中身をチェックする。clipboard 変数が空であったら、そこで終了する。

選択した文書と全体の文書はエンコードを行い、文字列の適合を行う。

選択した文書が空文字列であった場合、全体の文書の前に clipboard 変数の文字列を付け加え、エンコードによって変換された記号を元の記号へ戻して(デコード)から全体文書を書き換える。

選択した文書が空文字列でなかった場合、全体文書に選択した文書が適合した部分の後ろに clipboard 変数の文字列を付け加え、全体文書を書き換える(図 4.11)。

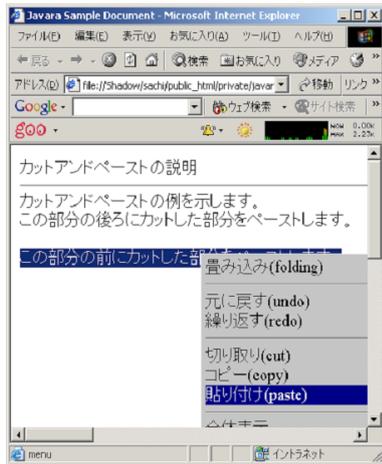


図 4.10: ペースト位置を選択

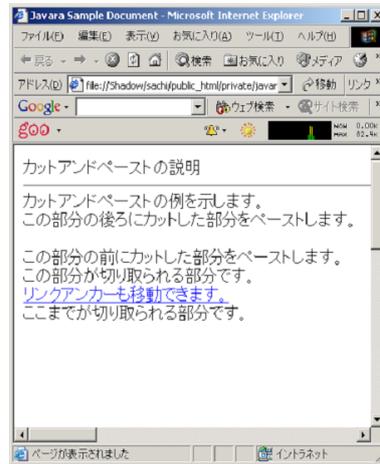


図 4.11: ペーストを実行

4.1.5 UNDO 機能

展開、畳み込み、枠の消去、カット、ペースト、アノテーションの機能を実行する際に、全体文書を UNDO 配列に格納する。UNDO 配列に格納すると、UNDO 配列の添え字 undoID をインクリメントする。

UNDO を実行した際に、undoID が 0 であった場合、JavaraII の編集作業は最初の状態となるので何も実行せずに終了する。それ以外の場合は、現在の文書を UNDO 配列に書き込み、undoID をデクリメントし、UNDO 配列内の文書を全体文書に置き換える。

4.1.6 REDO 機能

現在の undoID の次の配列要素が空文字列である場合、UNDO が行われていないため、REDO を行うことができないので何も実行せずに終了する。空文字列でなければ、その配列要素を全体文書に置き換え、undoID をインクリメントし終了する。

もし、UNDO を行った後に REDO 以外の機能を実行した場合、undoID 以降の配列要素を空にする。

4.1.7 アノテーション機能

ユーザはアノテーションを付けたいキーワードをマウสดラッグにより選択する。アノテーション機能を実行すると、選択文字列部分を span タグで囲い、そこに textarea を含んだ div 要素を書き込む (図 4.12)。

「書き込み」ボタンをクリックすると、関数 inputAnnote() が実行される。inputAnnote() では、対象となる ID(ここでは xx-div) の div 要素を、textarea に入力された文書と置き換える。

```

<span id="xx">
<a href="javascript:unfolding('xx',false)">キーワード</a>
<div class="bolder" id="xx-div">
  <form>
    <textarea name="source"></textarea>
    <input type="button" value="書き込み" onclick="inputAnnote(xx)">
    <input type="reset" value="リセット">
  </form>
</div>
<a href="javascript:unfolding('xx',false)">(close)</a>
<a href="remove_frame('xx')">(remove frame)</a>
</span>

```

図 4.12: アノテーションを入力するアンカー

4.1.8 保存機能

枠で囲って展開している部分を閉じるときに Javara 配列に書き込むため、保存機能の実行時に展開している部分を閉じる。文書が格納されている Javara 配列のデータを、ID 情報と文書を関連づけた文字列へ変換し、form の POST メソッドを用いてサーバ部へ受け渡す。

サーバ部での動作については 4.2 節に詳しく記述する。

4.1.9 インタフェースの改善

展開の確認

リンクの展開や畳み込みを行う際に、実行機能の対象となる文書を枠で囲い表示する。HTML のレイアウトに関するタグを定義し、スタイルシートとして組み込むための規格である CSS(cascading style sheets)[12] によって枠を表示する。対象となる文書は、div タグにより囲み、枠を表示するためのクラスを div に設定する。ここでは、JavaraII の表示に用いられている CSS を編集することで、表示枠の種類を変えることができる。

メニュー表示

マウスの右ボタンクリックをすると、メニュー表示を行う。右ボタンクリックによるメニュー表示を行う oncontextmenu を無効にし、右クリックを行った際にマウス位置にメニュー表示用の div を表示する(図 4.13)。畳み込み、カット、コピー、ペースト、UNDO、REDO、アノテーション、保存といった編集機能の他に、全体表示、元の大きさで表示、ワイドモード切替といった機能をメニューに表示する。メニューの内容は HTML で書かれており、各メニューの文字列をクリックすると各機能に対応する JavaScript 関数が実行される。



図 4.13: メニュー表示

マウスの左右ボタンの使い分け

event.button の数値によってマウスの左右ボタンを使い分ける。event.button が 1 の場合左ボタン、2 の場合右ボタンが使用されている。左ボタンは、文書の選択を行う。右ボタンがクリックされるとメニューを表示し、ドラッグされると文書のズームングを行う。

4.2 サーバ部

サーバ部はスクリプト言語である PHP で実装した。PHP は HTML 中にスクリプトを記述出来るという特性により Web 開発に適している。また、PHP はデータベースとの連携が優れているため、保存機能により HTML ファイル群をデータとして扱うことに有用であると考えた。

4.2.1 保存機能

インタフェース部から form の POST メソッドによって受け渡されたデータは、配列の要素ごとに分割し、ID を HTML ファイル名とする。文書にある open anchor 等の anchor は、対応する HTML ファイルへのリンクアンカーへと書き換える。HTML ファイルは保存日時ごとにログディレクトリを分割し、保存する。保存した HTML ファイルは、JavaScript を読み込んでいない文書と、文書内挿方式で閲覧できる JavaScript を読み込んだ文書と双方用意する。

4.3 JavaraII の利用法

どのように JavaraII を利用することができるか例を挙げる。

4.3.1 長大な Web 文書の切り分け

Web 文書が長くなってしまうと、ブラウザの画面上に収まりきらなくなってしまう、閲覧者は画面をスクロールして文書を読む必要がある。このような場合に、JavaraII の畳み込みによって、長くなった Web 文書を、ブラウザ画面に収まるように読みやすい長さに切り分けることができる。

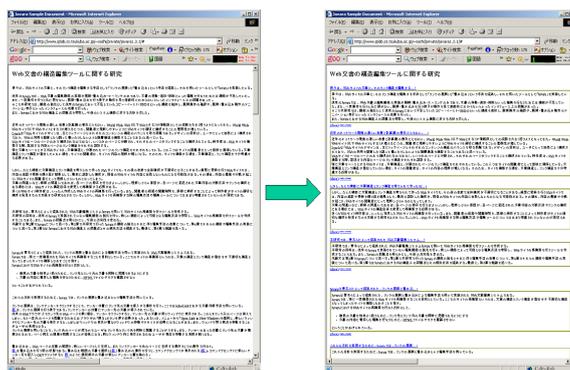


図 4.14: 長大な Web 文書の切り分け

また、Web 文書が長くなるにつれて、異なる内容の文書が同じ Web ページ内に現れてしまうことがある。その場合は、JavaraII の畳み込みを用いて、同じ内容の文書を一つのページにまとめて、カテゴリ分けをすることができる。

4.3.2 アノテーション

アノテーション機能を用いることで、JavaraII での編集に、任意のキーワードやフレーズと、コメントや注釈・用語説明などを自由にリンクさせることができる。まずユーザはアノテーションをつけるキーワード文字列を選択する (図 4.15)。選択した文字列に関して、文書を入力すると (図 4.16)、選択したキーワード文字列と、入力した文書がリンク関係になる (図 4.17)。このように、任意のキーワードやフレーズと、ユーザの入力文書をリンクにより結びつけることができる。

アノテーション機能は、個人的に閲覧している Web 文書に対してメモ代わりに用いることができる。また、ユーザが保有するサイトがあれば、そのサイト内のキーワードに対して説明書きや用語解説を行うといった使用ができる。

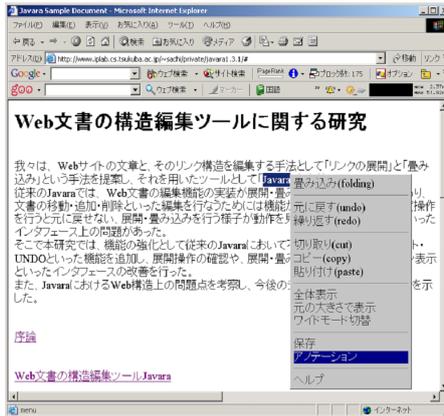


図 4.15: アンノテーションの対象となる文字列を選択

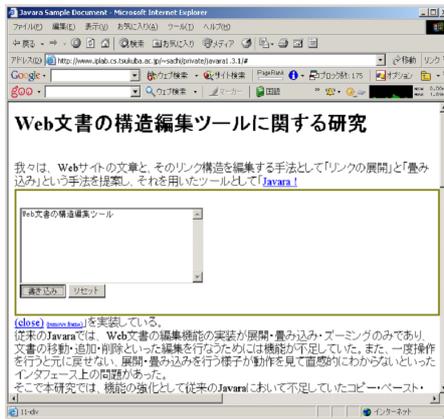


図 4.16: アンノテーションの入力

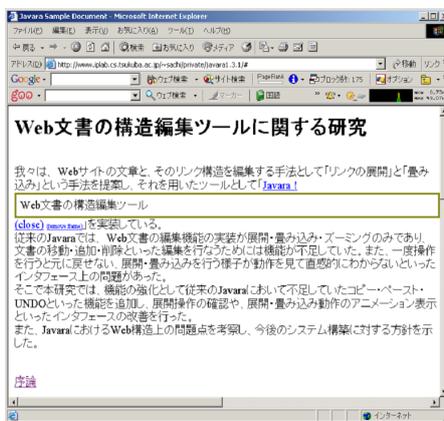


図 4.17: キーワードとアンノテーションがリンク関係になる

第5章 Web 構造上の問題と解決手法

リンクの展開を行っていくと、同じ Web ページを複数箇所に展開することで内容が重複してしまうことが起きる可能性がある。

一般に、Web 文書において同じ文書を参照したい場合、リンクをすることで参照できるため、同じ内容の文書が複数箇所に現れることは冗長であり、避けるべきである。また、再構築を行う際に文書を修正したい場合、同じ内容の文書が複数箇所にあると、同じ内容を何度も修正する必要があるため効率が悪い。

そこで、JavaRAII において編集を行う原則として、文書の内容が重複することを避けるようにする。

リンクを分類することで Web の構造について考察し、それにともなって起こる文書の内容が重複する問題と解決手法を述べる。

5.1 リンクの分類

Web のリンクを分類すると、リンク元に着目した分類と、リンク先に着目した分類ができる。また、その分類の中で、サイト内のリンク、他サイトのリンクといった分類も考えられる。ここでいうサイトというのは、ひとまとまりで公開されている Web ページ群であり、サイトの管理者であるユーザが、編集できる範囲のページ群を指す。

5.1.1 リンク元に着目した分類

リンク元に着目した場合、Web ページは以下のように分類できる。

- 一箇所からリンクされているページ
- 複数箇所からリンクされているページ
- 同じサイト内からリンクされているページ
- 他のサイトからリンクされているページ

一箇所からリンクされているページの場合、リンクの展開を行うことで文書の内容が重複することはない。しかし、複数箇所からリンクされているページを複数箇所にて展開した場合、同じ内容の文書が重複するため、これを避ける対処をする必要がある。

しかし、リンク元を調べるのはリンク先を調べることに比べて困難である。同じサイト内からリンクされている場合は調査が可能だが、他のサイトからのリンクを全て調べることは、全世界の Web ページを調査する必要があり事実上は不可能である。

そこで、文書の重複を避けるための対処法としてリンク先の文書に着目する。

5.1.2 リンク先に着目した分類

リンク先の URL に着目した場合、リンクは以下のように分類できる。

- 同じサイト内の他のページへのリンク (図 5.1)
- 同じサイト内の他のページの位置を示すアンカーへのリンク (図 5.2)
- 同じページ内の別の位置を示すアンカーへのリンク (図 5.3)
- 他のサイト内のページへのリンク (図 5.4)

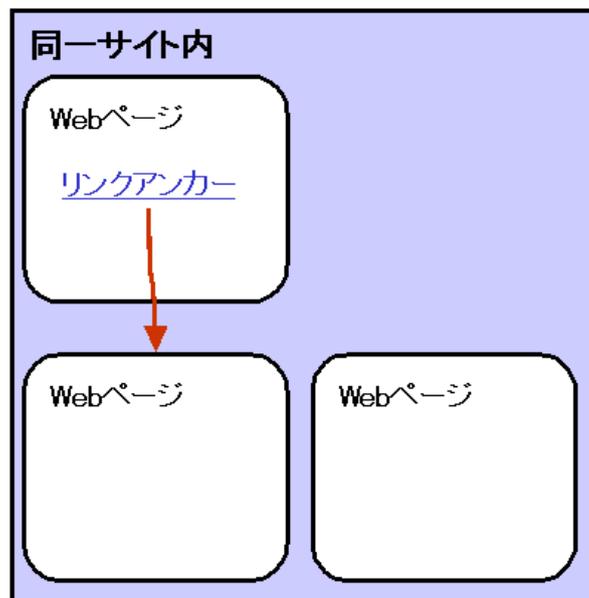


図 5.1: 同じサイト内のほかのページへのリンク

同じページの位置を示すアンカーへのリンクをは、展開すると自分自身を展開することになり、文書が重複するため展開を行わない。また、他のページの位置を示すアンカーへのリンクはそのページへのリンクと同等として扱うようにする。

他のサイト内のページへのリンクはユーザが管理するホームページ以外を無断で編集することで著作権の侵害につながる恐れがあるため、現段階では展開を行わない。

そこで同じサイト内の他のページへのリンクに着目して内容の重複が起こらない編集手法について考察する。

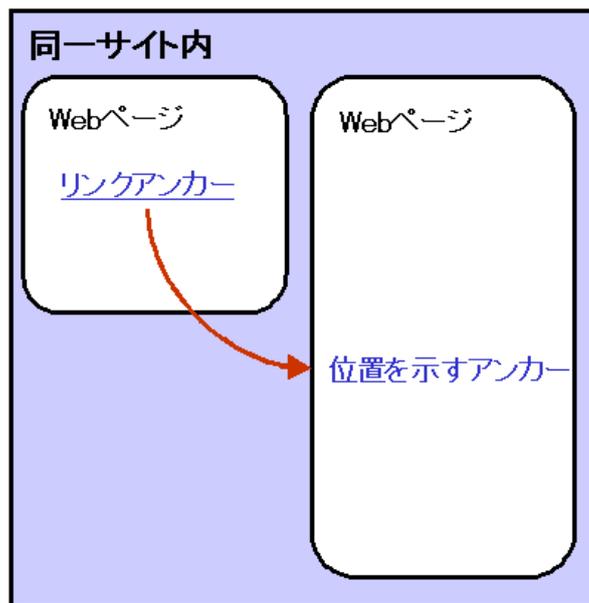


図 5.2: 同じサイト内の他のページの位置を示すアンカーへのリンク

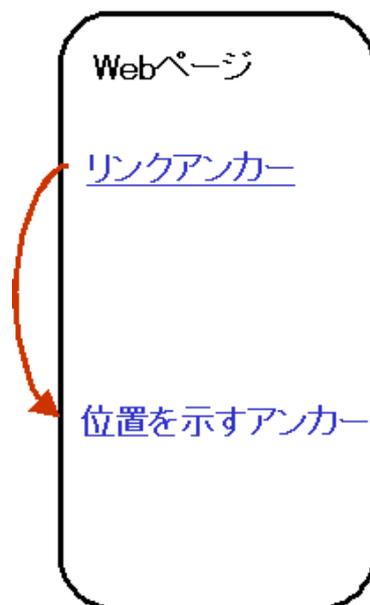


図 5.3: 同じページの位置を示すアンカーへのリンク

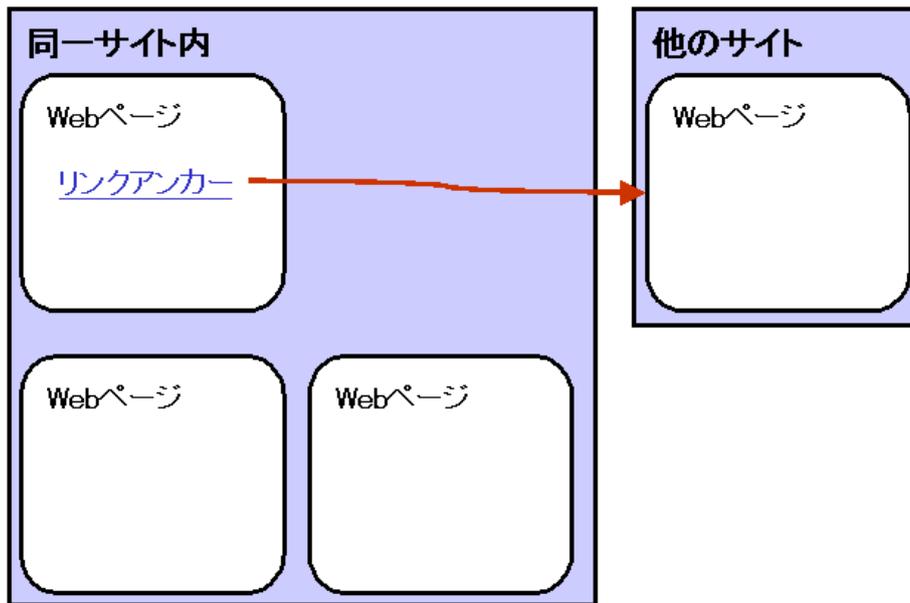


図 5.4: 他のサイト内のページへのリンク

5.2 内容の重複問題と解決手法

リンクの状態がどのようになっているときに内容の重複が起こるのか考察する。

5.2.1 内容の重複が起こるリンク

内容の重複が起こるのは次の状態のときである。

- 複数の個所から一つのページにリンクがあるとき (図 5.5)
- 相互にリンクしているとき (図 5.6)
- ページがループ状にリンクしているとき (図 5.7)

ただし、相互にリンクしているときは、ページがループ状にリンクしているときの一つの形として扱うことができる。

これらの場合はリンクの展開を行っていくと同じページを展開することになるので内容の重複が起こる。

例えば、図 5.7 のように、ページがループ状にリンクしている場合に、リンクの展開を続けると、図 5.8 のように同じ A というページを複数の個所で展開してしまう。

このような内容の重複を避けるために、次に解決手法を提案する。

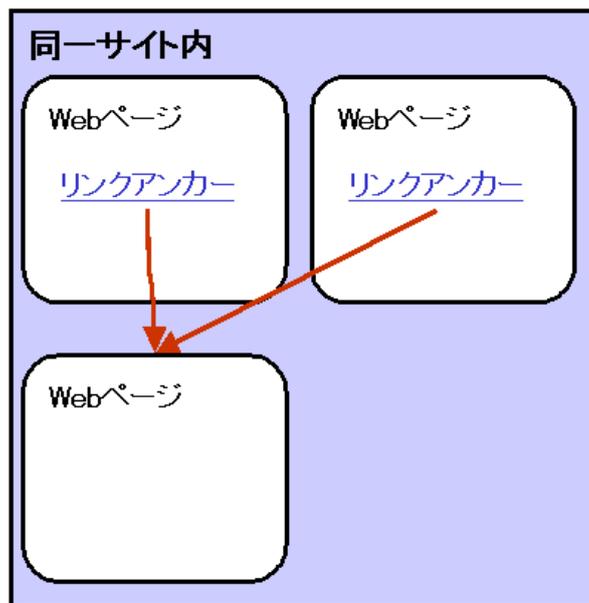


図 5.5: 複数の個所から一つのページにリンクがあるとき

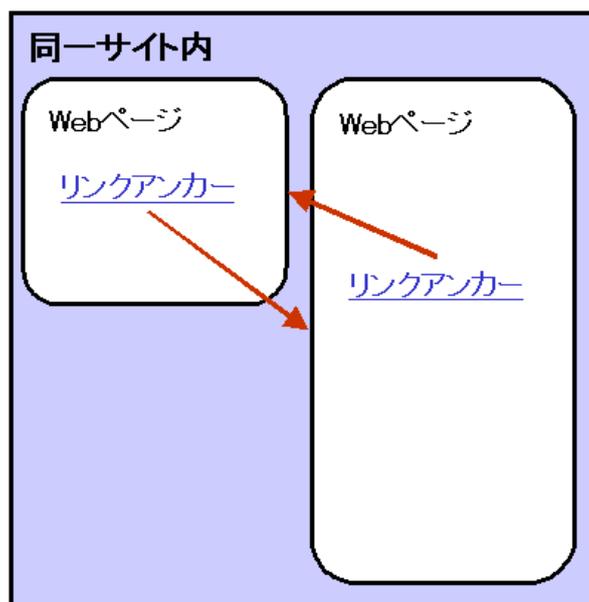


図 5.6: 相互にリンクしているとき

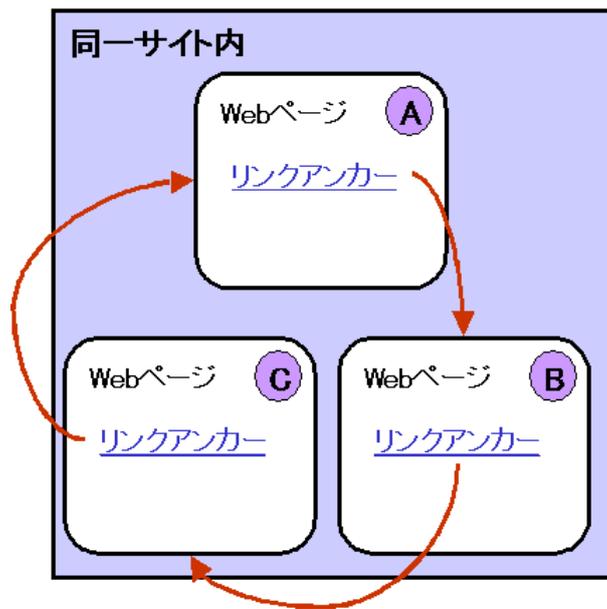


図 5.7: ページがループ状にリンクしているとき

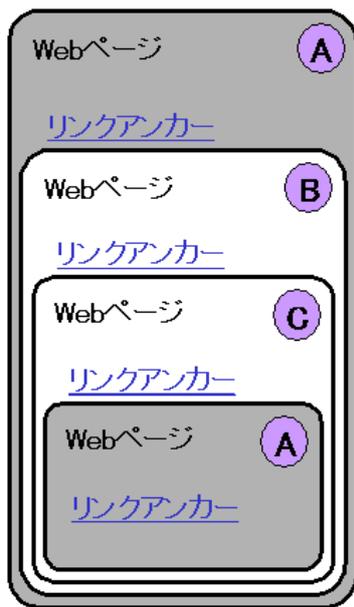


図 5.8: ループ状のリンクの展開

5.2.2 解決手法

内容の重複が起こる全ての場合を避けるために次のような手法を用いる。

1. 展開したページの情報はどの個所で展開したかという情報とともに記憶しておく
2. 一度展開したページは展開を行わない
3. 一度展開したページにリンクしているため展開を行えないアンカーは、展開する対象のページがどこで展開を行われたかという情報をマーキングしておく
4. 展開を行えなかったアンカーは、編集が終了し編集状態を保存する際に、マーキングされた情報を元にリンク先を新しく展開した個所へ書き換えるようにする

一度展開したページを開かないようにすることで複数の箇所からリンクしている場合、相互にリンクしている場合、ページがループ状にリンクしている場合のいずれでも文書の内容の重複を避けることができる。

また、同じ内容へリンクしているアンカーは、編集時に展開できなくても、最終的に同じ個所へリンクを書き換えるためリンク先が参照できないといった問題は起こらない。

この手法により、文書の重複による無駄がなくなり、また、リンクを改めて編集しなおす必要がないため、効率のよい Web サイトの再構築が可能となる。

第6章 関連研究

6.1 ブラウザ上で編集を行うツール

ブラウザ上で編集を行うことができるツールとして Wiki[13] が挙げられる。Wiki では、複数人による Web サイトの編集を目的とし、誰でも対象のサイトのある一定のルールに従い編集することができる (図 6.1)。Wiki では、Wiki ワードと呼ばれるキーワードによって Web ページ同士がリンクしあっている。キーワード同士が複雑にリンクしあっているため、文書構造の把握が容易でない。また、ユーザは Wiki に記述するためのルールを覚える必要がある。Wiki は編集を行うために、Wiki 専用の Web ページが必要だが、JavaII は一般的な HTML ファイルの編集が可能である。

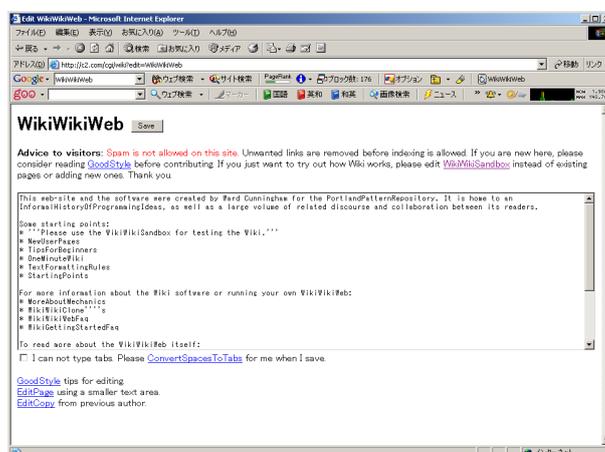


図 6.1: Wiki の編集画面

6.2 構造編集ツール

Web サイトの構造を編集するツールとして DENIM[14] が挙げられる。DENIM は Web サイトの設計の早い段階で、Web サイト全体の構造をラフスケッチとして手書きで作成する (図 6.2)。ユーザは Web ページ内の要素と、Web ページ同士のリンク関係を階層として記述する。DENIM は Web サイトのプロトタイプを作成することに長けているが、Web サイトを編集しなおしたり、Web ページの文書構造の把握はできない。また、JavaII は編集を行いながらブ

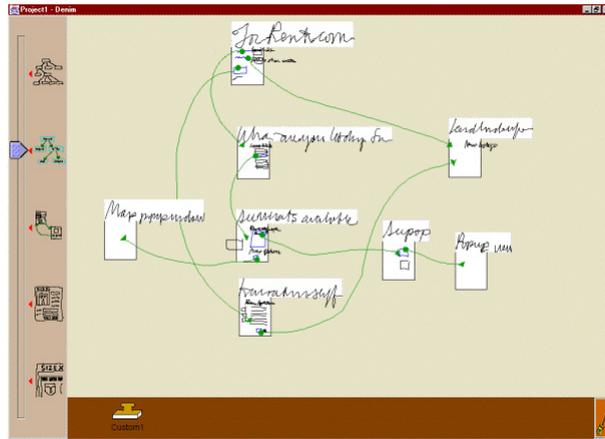


図 6.2: DENIM の編集画面

ブラウザ上でどのように表示されるか把握できるが、DENIM は編集が終わった後に DENIM と別にブラウザを起動して確認する必要がある。

また、広く知られている HTML オーサリングツールである Microsoft FrontPage[15] や Macromedia Dreamweaver[16] は、ページ同士のリンク構造を把握する機能を持つ。しかし、文書内容の構造の把握は、個々の HTML ファイルを個別に参照する必要がある。

第7章 結論

7.1 結論

本研究では、Web 文書の構造編集ツール Javara において、問題点を考察し、新しく JavaraII を設計・実装した。JavaraII では、従来の Javara で実装が不十分であったカット・コピー・ペースト機能、UNDO 機能といった編集機能について考察し、実装した。インタフェース部において行った編集状態を、HTML ファイルへ変換し保存する保存機能を実装した。Javara における編集特性を考慮したアノテーション機能を提案し、実装した。また、インタフェースの改善を行い、リンクの展開・畳み込み動作のアニメーション表示、左右のマウスボタンを使い分ける、従来キーボードで行っていた編集操作のメニュー表示などを実現した。

これらの機能の強化により、Web 文書の編集が行いやすくなり、Web 文書の編集において操作性が向上した。

また、Web 構造について考察し、JavaraII における Web 構造上で起こる問題とその解決方法を示した。

7.2 今後の課題

今後の課題として、Javara II の機能の評価が挙げられる。評価方法は、ユーザの使いやすさの主観的評価、Web 文書の構造を編集する速度等を評価対象として他のオーサリングツールとの比較することを考えている。

また、Web 構造上に起こる問題について提案した機能の実装と、その評価が挙げられる。提案した手法が優れているのか、他の手法との比較を行いながら、Web 構造上におこる問題の解決を行う。

謝辞

研究全般に渡り、ご指導くださった指導教官の田中二郎教授に心から感謝いたします。三末和男助教授、志築文太郎講師、高橋伸講師はじめ、田中研究室の皆さんには多くの貴重なご意見をいただきました。深く感謝いたします。また、Webグループのグループリーダーであった三浦元喜助手には、研究の詳細に渡りアドバイスをいただきました。ありがとうございます。友人や家族の他、私を支えてくださった多くの方々に感謝いたします。ありがとうございました。

参考文献

- [1] Patrick J. Lynch. *Web Style Guide: Basic Design Principles for Creating Web Sites*, March 1999.
- [2] Pawan R. Vora. Designing for the Web: A Survey. *ACM Interactions volume 5, Issue 3, May/June*, pp. 13–30, 1998.
- [3] Jason I. Hong and James A. Landay. WebQuilt: A Framework for Capturing and Visualizing the Web Experience. In *The tenth World Wide Web conference on World Wide Web*, pp. 712–724, October 2001.
- [4] Filippo Ricca and Paolo Tonella. Understanding and Restructuring Web Sites with ReWeb. In *IEEE MultiMedia, Vol. 8, No. 2, April-June 2001*, pp. 40–51, April 2001.
- [5] 青木裕伸. 展開と畳み込みに基づく Web 文章編集システム. Master's thesis, 筑波大学工学研究科, 2001.
- [6] Hironobu Aoki, Motoki Miura, and Jiro Tanaka. A Web Site Editing System based on Unfold/Fold Operations. In *Proceedings of the International Conference on Internet Computing (IC 2002)*, pp. 297–302, June 2002.
- [7] Motoki Miura, Buntarou Shizuki, and Jiro Tanaka. inlineLink: Inline Expansion Link Methods in Hypertext Browsing. In *Proceedings of International Conference on Internet Computing (IC2001)*, pp. 653–659, June 2001.
- [8] 三浦元喜, 志築文太郎, 田中二郎. Web ブラウザを利用した文書内挿機能の実現. *情報処理学会論文誌*, Vol.43, No.12, pp. 3706–3717, 2002.
- [9] W3C Working Draft. *Document Object Model(DOM) Level 2 HTML Specification Version 1.0* <http://www.w3.org/TR/DOM-Level-2-HTML/>, January 2003.
- [10] 小野田紗智, 三浦元喜, 田中二郎. 展開と畳み込みに基づく Web 文書の構造編集ツール「Javara」. *ソフトウェア科学会第 20 回記念大会*, pp. 391–395, 2003.
- [11] Benjamin B. Bederson, Jon Meyer, and Lance Good. Jazz: an extensible zoomable user interface graphics toolkit in java. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 171–180. ACM Press, 2000.

- [12] W3C Recommendation. *Cascading Style Sheets, level 2 CSS2 Specification*
<http://www.w3.org/TR/REC-CSS2/>, May 1998.
- [13] Bo Leuf and Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.
- [14] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: finding a tighter fit between tools and practice for web site design. In *CHI*, pp. 510–517, 2000.
- [15] Microsoft. Microsoft FrontPage, 1995. <http://www.microsoft.com/frontpage/>.
- [16] Macromedia. Macromedia Dreamweaver, 1997. <http://www.macromedia.com/>.