

携帯端末画面への HTML 表データの表示方法

安富 大輔[†], 増田 英孝[†], 中川 裕志[‡]

[†] 東京電機大学工学部

〒 101-8457 東京都千代田区神田錦町 2-2

[‡] 東京大学 情報基盤センター

〒 113-0033 東京都文京区本郷 7-3-1

e-mail:[†]{yasutomi,masuda}@ecal.e.dendai.ac.jp, [‡]nakagawa@r.dl.itc.u-tokyo.ac.jp

概要

本来、表形式のデータ表現は、情報を簡潔に見やすくするために使われる。しかし、Web 上に多数存在する HTML の表を、携帯端末などの小さく解像度の低い画面に表示すると、表全体が一挙に表示できないので読みにくいことはもちろん、表内部のデータの折り返しが増えて可読性が低下し、改行位置によっては、データの読み誤りが発生する。また、頻繁なページの切り換えによって、データの行と列の関係を記憶しておくことが難しくなる。

そこで、本研究では、HTML の <TABLE> タグに主眼を置き、実際に Web 上に存在する表を調査し、分類を行った。その結果を元に、HTML の表を携帯端末に適した形に自動変換するシステムを提案する。

How To Display Tables in HTML on Personal Digital Assistants

Daisuke YASUTOMI, Hidetaka MASUDA

Department of Electrical Engineering,
Study of Engineering,
Tokyo Denki University,
2-2 Kandanishiki-cho, Chiyoda-ku,
Tokyo 101-8457, Japan

Hiroshi NAKAGAWA
Information Technology Center,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-0033, Japan

Abstract

Displaying information with table form are used to make information clearly. However in browsing HTML tables on a PDA, readability is almost dispaired by a PDA's small and low resolution display screen . In fact, its low readability is caused by the phenomena such as only a small portion of table is shown on the screen at once, original one line is usually broken up many lines, and so on.

In our research, we focus on <TABLE> tags in HTML. We investigated into existing tables in the Web and divided into three classes. On the basis of the result, we propose an automatic transformation system which can fit generic HTML tables to PDA's screen.

1 はじめに

近年では、iモードやPDAなどの小型携帯端末からWebページをブラウズしたいという要求が急激に増加している。

しかし現状では、解像度が大きいPCを対象としたページが大多数である。そのため、iモードやPDAなどの解像度の低い携帯端末では、画面の表示文字数が少ないことからの行の折り返しの増加、可読性の低下や読み誤りといった問題が発生する。また、画面をスクロールさせるための操作量が増加する問題がある。さらに、Webページ上のテーブルを表示する際に、ブラウザによって<TABLE>タグの取り扱い方や、対応するタグの種類が異なるため、問題が発生する[1]。

これらの問題を解決するために、情報を携帯端末の一画面に収めるための文章要約や、文章の体言止め、言い換えなどの手法が研究され始めている[2]。また、コンテンツを端末に向けて動的に生成する方式も考えられる[3]。この場合、XML(eXtensible Markup Language)などの内部データをサーバ側に用意し、携帯端末の情報を得てそれぞれに適したHTML,CHTMLなどの表示データを生成する。

しかしながら現状では、携帯端末向けに動的にコンテンツを生成するシステムは一部の企業でしか使われておらず、一般にはごく僅かしか利用できない。そのため、PC向けに作成されたページを新たに携帯端末向けに作成し直す方法が一般的である。また、PC専用で作られているWebページがほとんどであり、携帯端末で閲覧する時は、携帯端末向けに自動変換する必要がある。

そこで、本研究ではPC向けに作成されたWebページを閲覧するために<TABLE>タグに主眼を置き、テーブルを携帯端末に適した形に自動変換するシステムを提案する。

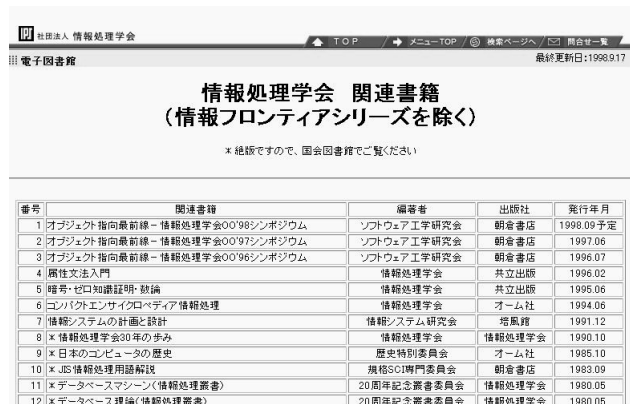
2 携帯端末表示の問題

携帯端末表示の問題として、携帯端末を用いたWebページのブラウズには表示領域の問題とテーブル表示の問題がある。

2.1 表示領域の問題

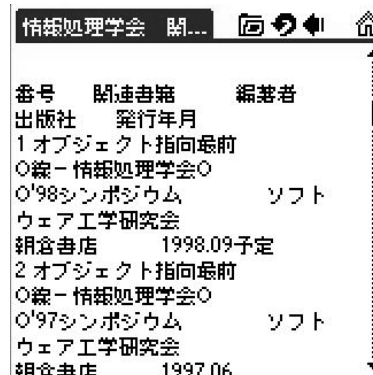
iモード,PDAなどの携帯端末においては、画面領域の問題は非常に重要な問題となる。これは現在のWebページの多くが、PCの表示領域に適した形で作られているからである。

PC上にテーブルを含むWebページを表示した結果を図1に示す。画面の解像度が高いので表全体が見渡せる。



番号	関連書籍	編著者	出版社	発行年月
1	オブジェクト指向最新録-情報処理学会00'98シンポジウム	ソフトウェア工学研究会	朝倉書店	1998.09予定
2	オブジェクト指向最新録-情報処理学会00'97シンポジウム	ソフトウェア工学研究会	朝倉書店	1997.06
3	オブジェクト指向最新録-情報処理学会00'96シンポジウム	ソフトウェア工学研究会	朝倉書店	1996.07
4	簡性文法入門	情報処理学会	共立出版	1996.02
5	暗号・ゼロ知識証明-跋論	情報処理学会	共立出版	1996.06
6	コンパクトエンサイクロペディア情報処理	情報処理学会	オーム社	1994.06
7	情報システムの計画と設計	情報システム研究会	培風館	1991.12
8	情報処理学会30年の歩み	情報処理学会	情報処理学会	1990.10
9	日本のコンピュータの歴史	歴史特別委員会	オーム社	1985.10
10	JS情報処理用語解説	規格SCT専門委員会	朝倉書店	1983.09
11	データベースマシーン(情報処理叢書)	20周年記念委員会	情報処理学会	1980.05
12	データベース理論(情報処理叢書)	20周年記念委員会	情報処理学会	1980.05

図1: 情報処理学会のページをPCで表示した例



番号	関連書籍	編著者
1	オブジェクト指向最新録 0線-情報処理学会0 0'98シンポジウム	ソフト ウェア工学研究会 朝倉書店
2	オブジェクト指向最新録 0線-情報処理学会0 0'97シンポジウム	ソフト ウェア工学研究会 朝倉書店

図2: 情報処理学会のページをAvantGoで表示した例

図2は、PDAブラウザのAvantGo[4]で同一のWebページを表示した例である。PCに比べて解像度が低いため、携帯端末からでは可読性が低下している。また、Blazer[5]でも同様な表示結果が得られる。

また、表示領域に起因してPC上では1ページ

として表示できるが、携帯端末からでは、数ページに及ぶことになり、操作量が増える。さらに、行の折り返しによって正しい情報がユーザに伝わらないという問題も発生する。

番号	関連書籍	編著者	出版社	発行年月
1	オブジェクト指向最新情報処理学会00'98シンポジウム	ソフトウェア工学研究会	朝倉書店	1998.09 予定
2	オブジェクト指向最新情報処理学会	ソフトウェア工学研究会	朝倉書店	1997.06

図 3: 情報処理学会のページを Palmscape で表示した例

図 3 は、PDA ブラウザの Palmscape[6] で同一の Web ページを表示した例である。発行年月のデータを見ると表示領域の横幅が狭いために“1998.09 予定”という文が多数の折り返しによって読みにくい上、読み誤りをする恐れがあることがわかる。

2.2 テーブル表示の問題

携帯端末では、ブラウザによって <TABLE> タグの取り扱い方が異なるため、テーブル表示に問題が発生してしまう。

図 2 の AvantGo のテーブル表示では、テーブルのデータを行毎に順に列挙して表示している。

しかしながら、テーブルのデータを順に表示するだけなので、元の表データの行と列の関係がわかりにくくなってしまふ。また表が長ければ長いほどユーザにとっては、テーブルが何を示しているのかを理解しにくくなる。

図 3 の Palmscape のテーブル表示では、罫線が表示されテーブルとしては、AvantGo より可読性がよいが、列が増えれば増えるほど、セル幅が狭くなってしまふため、可読性が低下してしまふ。また AvantGo と同様に、ページの切り換えによってデータの行と列の関係を記憶しておくことが難しくなる。

図 4(a) は、郵便料金のページを PC で表示して例である。また、図 4(b) は同一の Web ページを Palmscape で表示した例である。テーブルの <TD>、<TH> タグの colspan, rowspan オプションの値が増加すると 1 つの行データを一画面で表示することができなくなり、可読性が低下してしまふ。

		50gまで	1,150円
		40gまで	1,250円
第二種郵便物 (はがき)	郵便書簡(ミニレター)(注2)		80円
	通常はがき		50円
	往復はがき		100円
		小包はがき	50円
第三種郵便物 (認可を受けた定期刊行物・開封)	下記以外の第三種郵便物	50gまで	60円
		50gを超える1kgまで、50gまでごとに	8円増
	毎月3回以上発行する新聞紙1部又は1日分を内容とし、発行人又は売りさばき人から差し出されるもの	50gまで	40円
		50gを超える1kgまで、50gまでごとに	6円増
	毎月3回以上発行する新聞紙	50gまで	8円
		50gを超える1kgまで、50gまでごとに	3円増
心身障害者団体の発行する定期刊行物を内容とし、発行人から差し出されるもの	毎月3回以上発行する新聞紙	50gまで	15円
		50gを超える1kgまで、50gまでごとに	5円増
	その他のもの	50gまで	15円
		100gまで	15円
		100gを超え、1kg(91g)まで	10円増
		100gまでごとに	

(a)

行物・開封	重量	料
心身障害者団体の発行する定期刊行物を内容とし、発行人か	3回を超える以上発行する新聞紙	3円増
	50gまで	15円
	50g	

(b)

図 4: 行データが 1 画面に入らない問題

本来テーブルは、情報を簡潔に見やすくする目的で使用されるが、携帯端末ではテーブルの可読性が非常に悪くなってしまふ。また表示領域で起こる問題と同様に、操作の煩雑さや読み誤りといった問題も起こる。そのため、情報を維持したまま携帯端末用にテーブルを別の表現形式に自動変換する必要がある。

3 テーブルの調査と分類

現在 Web 上にどのようなテーブルが存在するのかを実際に調査した。今回の調査では、Web 上にある 100 のテーブルをランダムに抽出して調査対象とした。

本研究では、テーブルを使用目的に応じて以下の 3 つに分類する。

- レイアウト
- 本来のテーブル
- 特殊なテーブル

3.1 レイアウト

本来のテーブルとして使うのではなく、Web ページの体裁を整えるために使用する方法である。レイアウトとして利用する場合には、罫線を表示させないために、<TABLE> タグの border オプションを 0 にしている場合が多い。

3.2 本来のテーブル

本来のテーブルとしての利用をテーブルのタイプ別に以下の 3 つに分類する。

- 時間割型
- 縦一覧型
- 横一覧型

3.2.1 時間割型

行と列どちらにも属性 (AttributeX, AttributeY) を持っている場合で時間割などのテーブルがこれに相当する。このテーブルは、さらにメタデータとして列の属性にカテゴリー分けされた部分 (MetaX, MetaY) が存在することもある。本研究では、このようなテーブルを時間割型と定義し、この時間割型の基本型を図 5 に示す。

3.2.2 縦一覧型

1 行目のそれぞれの列データが属性となっており 2 行目以降のそれぞれの列データが属性の値となっているテーブルである。これを本研究では、縦一覧型とし基本型を図 6 に示す。

		MX		
		AX1	...	AXn
MY	AY1	value(X1,Y1)	...	value(Xn,Y1)
	AY2	value(X1,Y2)	...	value(Xn,Y2)

	AYm	value(X1,Ym)	...	value(Xn,Ym)

MetaY = MY, AX = AX, MetaX = MX, AttributeX = AX

図 5: 時間割型の基本型

AttributeX1	AttributeX2	...	AttributeXn
valueX11	valueX12	...	valueX1n
...
valueXm1	valueXm2	...	valueXmn

図 6: 縦一覧型の基本型

3.2.3 横一覧型

横一覧型は、縦一覧型を転置して、一列目のデータが属性となっており、2 列目以降の列データが属性の値となっているテーブルであり、2 列のテーブルであることが多い。本研究では、このようなテーブルを横一覧型と定義し、横一覧型の基本形を図 7 に示す。

AttributeX1	valueX11	valueX21	...	valueXm1
AttributeX2	valueX12	valueX22	...	valueXm2
...
AttributeXn	valueX1n	valueX2n	...	valueXmn

図 7: 横一覧型の基本型

3.3 特殊なテーブル

特殊なテーブルは、テーブルの目的であるデータの一貫性を考慮したものではなく、カレンダーやリンクの一覧などとして使用している場合である。

3.4 テーブルの使用頻度

実際に Web ページで使われているテーブルの調査を行い、テーブルタイプ毎に使用頻度を調査した結果を図 8 に示す。

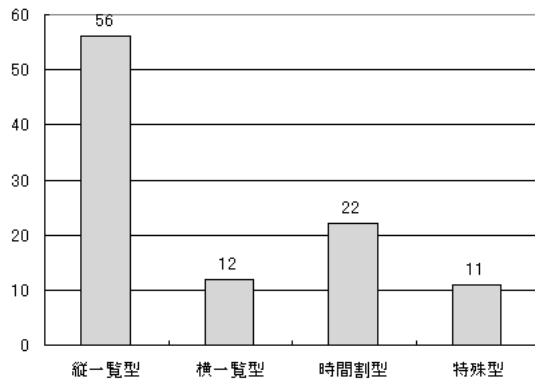


図 8: テーブルの使用頻度

縦一覧型が 5 割以上を占めており、使われる頻度が高いことがわかる。縦一覧型と横一覧型を 1 つの一覧型として考えると一覧型だけで 7 割近く使われていることが分かる。

本調査で、特殊型に分類されたものはカレンダーやリンクなどにテーブルを使用しているものである。また、レイアウトとして使われているテーブルは時間割型、縦一覧型、横一覧型、特殊型に比べ使われる頻度が高いこともわかった。

3.5 時間割型における使用頻度

時間割型には、Attribute, Meta データの組合せによってさまざまなテーブルタイプが存在するため、変換するテーブルタイプの判別が必要となる。そのため、さらに時間割型に限定して分類を行った。図 9 に時間割型の各部位の使用頻度を示す。

AttributeX, AttributeY の使用頻度に対して MetaX, MetaY の使用頻度は半分程度である。

また、各部位を利用したテーブルにはどのような組み合わせがあるのかを調査した。結果を図 10 に示す。

AttributeX, AttributeY のみを使用したテーブルが多いが、実際にはそれぞれのテーブルの使用

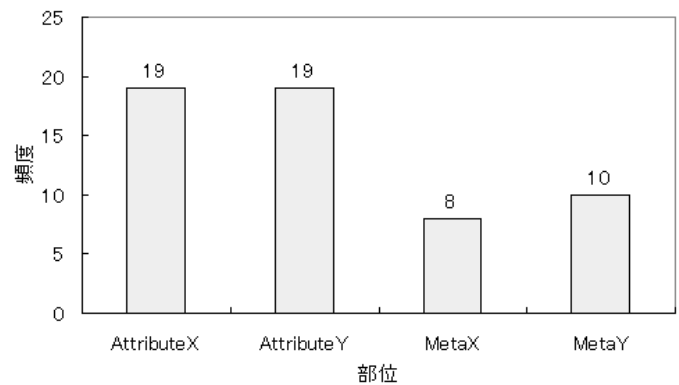


図 9: 時間割型における各部位の使用頻度

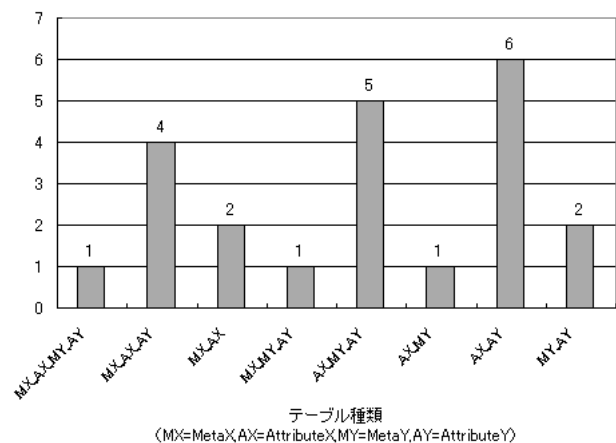


図 10: 時間割型におけるテーブル種別

頻度が分散している。

しかしながら、今回は 22 の時間割型のテーブルを分類しただけなので、今後はさらに多くのテーブルの調査を行う必要がある。

4 提案するシステム

表示領域の問題における操作の繁雑さや、行の折り返し問題を解決するためには自然言語処理による文章要約、言い換え、文章の体言止めを行うことが有効である [2]。

また、テーブル表示の問題を解消するためには、テーブルを別の表現形式に自動変換して表示する必要がある。

本研究では、表示領域の問題は自然言語処理で解決できるものとして、テーブル表示の問題である <TABLE> タグに主眼を置き、携帯端末に適した表現形式でテーブルを自動変換するシステムを提案する。

5 システムの構成

本研究では、Webサーバとして Apache と tomcat を、正規表現ライブラリとして Jakarta-oro を用いてテーブル自動変換システムを Java 言語で実装した。試作システムは、ユーザが変換したい Web ページの URL を入力すると、サーバ側でテーブルの型を自動判別し、それぞれの変換アルゴリズムでテーブルの自動変換を行い、クライアントに変換した Web ページを返すシステムである。システム構成を図 11 に示す。

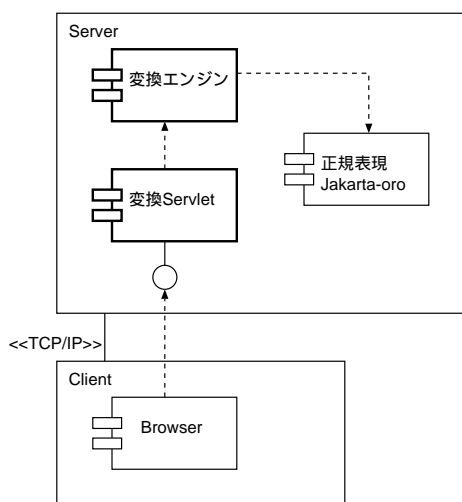


図 11: システム構成

5.1 変換アルゴリズム

5.1.1 レイアウト型

レイアウト型では、それぞれの行データを順に表示していく。

5.1.2 時間割型

時間割型は、AttributeX と AttributeY の 2 つの Attribute データがあるためメインの Attribute データをユーザが選択する必要がある。本システムでは、1 つの Web ページの全てのテーブルに対して AttributeX, AttributeY のどちらのデータを Attribute データにするか選択できる。図 12(a) の時間割型を AttributeY データを Attribute データとした変換結果は図 12(b) となる。

		MX		
		AX1	...	AXn
MY	AY1	value(X1,Y1)	...	value(Xn,Y1)
	AY2	value(X1,Y2)	...	value(Xn,Y2)

	AYm	value(X1,Ym)	...	value(Xn,Ym)

MetaY = MY, AX = AX, MetaX = MX, AttributeX = AX

(a)

[MetaY:AttributeY1]	
[MetaX:AttributeX1]	value(X1,Y1)
...	
[MetaX:AttributeXn]	value(Xn,Y1)
...	
[MetaY:AttributeY2]	
[MetaX:AttributeX1]	value(X1,Y2)
...	
[MetaX:AttributeX1]	value(Xn,Y2)

(b)

図 12: 時間割型の変換

5.1.3 縦一覧型

縦一覧型は、1 行目の Attribute データ、2 行目以降の value データの組を順に表示していく。図 13(a) の縦一覧型の変換結果は図 13(b) となる。

5.1.4 横一覧型

横一覧型は、縦一覧型とは逆に 1 列目の Attribute データと 2 列目以降の value データの組

AttributeX1	AttributeX2	...	AttributeXn
valueX11	valueX12	...	valueX1n
...
valueXm1	valueXm2	...	valueXmn

(a)

[AttributeX1]	value(X11)
[AttributeX2]	value(X21)
...	...
[AttributeXn]	value(X1n)
...	...
[AttributeX1]	value(Xm1)
[AttributeX2]	value(Xm2)
...	...
[AttributeXn]	value(Xmn)

(b)

図 13: 縦一覧型の変換

を順に表示していく。図 14(a) の横一覧の変換結果は図 14(b) となる。

AttributeX1	valueX11	valueX21	...	valueXm1
AttributeX2	valueX12	valueX22	...	valueXm2
...
AttributeXn	valueX1n	valueX2n	...	valueXmn

(a)

[AttributeX1]	valueX11,valueX21, ...,valueXm1
[AttributeX2]	valueX12,valueX22, ...,valueXm2
...	...
[AttributeXn]	valueX1n,valueX2n, ...,valueXmn

(b)

図 14: 横一覧型の変換

5.2 変換結果

図 1 の Web ページを変換した結果を図 15 に示す。

図 3 の Palmscape 使用時のテーブル表示と比

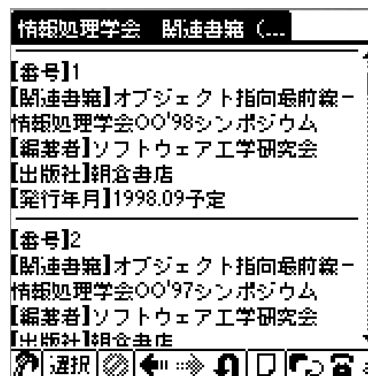


図 15: 図 1 の Web ページの変換結果

較して、表示領域の問題である行の折り返しや、テーブルのデータが増えることによる可読性の低下を Attribute データを必ず value データに付与して表示することによって解決している。

同様に、図 4 の Web ページを変換した結果を図 16 に示す。

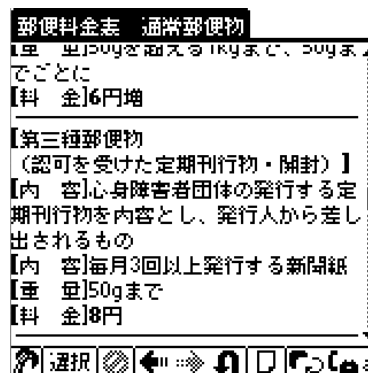


図 16: 図 4 の Web ページの変換結果

このページも同様に、Attribute データをそれぞれの value データに付与することによって可読性の低下を解決している。また、colspan, rowspan オプションの値の増加による、テーブルの行データを一画面で閲覧できない問題も解決している。

6 解決すべき問題

現在の試作システムでは、以下のような問題点がある。この章では、これらの問題点に対する解決策を述べる。

- Attribute のデータの自動判別
- データ量の増加
- 操作の煩雑さ

6.1 Attribute データの自動判別

現在のシステムでは、Attribute データの判別を <TD>, <TH> タグの colspan, rowspan オプションなどを利用しているため、時間割型の Attribute, Meta データの判別が必ずしも正しく判別ができない。また、縦一覧型と横一覧型の判別も同様である。

そこで、予め数字や曜日などの連続したデータをシステムに用意しておき、そのデータからテーブルの Meta, Attribute データを行、列をチェックして判別する機能が必要であると考えられる。また、ユーザの可読性を高めるために時間割型については、AttributeX と AttributeY のどちらをメインの Attribute とするかユーザ自身がテーブル毎に選択できる機能も必要である。

6.2 データ量の増加

本システムでは、Attribute データを value データに付与することによって可読性の低下を解決しているが、逆にどの value データにも Attribute データを付与しているため、データ量が増加してしまう。このため、i モードなどではデータ量に制限があるため情報が欠落してしまう恐れがある。

この問題の解決策としては、1 画面で行データを表示できるものについては Meta データは一度だけ表示する。また、それぞれの value データをリンクとして別のページに表示することも考えられる。

6.3 操作の煩雑さ

現在の変換システムは、解像度が小さいことによる操作の煩雑さは解決できていない。また、データ量の増加によって変換前のテーブルよりもさらに操作量が増えてしまう恐れがある。

操作の煩雑さの解決策としては、あらかじめ文章中の重要語を検索し、その重要語を使ったテーブルデータを最初にソートして表示したり、

ユーザ自身がソートする方策も考えられる。また、データ量の増加での問題と同様に value データをリンクとして別のページに表示することも考えられる。

7 まとめ

本研究では、携帯端末における表示領域の問題と、携帯端末のテーブル表示の問題に対する解決策としてテーブルを携帯端末に適した形に自動変換するシステムを提案し、試作システムを作成した。このシステムにより、携帯端末でのテーブル表示の問題である可読性の低下を解消することができた。

しかし、テーブル構造だけでテーブルを変換するには限界があり、現状の課題として Attribute データの自動判別、データ量の増加、操作の煩雑さの解消がある。これらを解決するためには、自然言語処理を用いてテーブルのデータをチェックする必要がある。また、現在のテーブル調査は 100 テーブルを対象とした結果である。そのため、さらなるテーブル調査・分類が必要になると考えられる。

参考文献

- [1] 北山 文彦, 広瀬 紳一:
“Dharma ささまざまなインターネット端末にコンテンツを適応させるソフトウェア技術”, Vol.42, No.6, 情報処理 (2001)
- [2] 渡部聡彦, 武井純孝, 杉本雅則, 中川裕志:
“携帯端末へのカタログの情報の表示のための言い換え方策”, 言語処理学会第 7 回年次大会ワークショップ (2001)
- [3] Vertex Link Corporation: C3GATE Server,
<http://www.vertexlink.co.jp/>
- [4] AvantGo, Inc: AvantGo3.3,
<http://avantgo.com/>
- [5] Handspring, Inc: Blazer1.0,
<http://www.handspring.com/>
- [6] 株式会社イリンクス: Palmscape3.1,
<http://www.ilinx.co.jp/>