

日本語文書における表記述のための表モデルについて

三浦好弘 古川善吾 牛島和夫
九州大学工学部情報工学科

JT_PX、J-Star など日本語文書作成支援システムが存在する。これらは英語版を日本語化したものであるために、日本語文書に馴染まないところがある。特に、表生成機能については、英語文書に出てくる形の表は問題なく書けるが、履歴書のような日本語文書特有の表を書くには問題が多い。そこで、日本語文書にあった表形式の生成方法を開発する。日本語文書にあった表を生成するためには、日本語文書特有の表を分析し、その構造を明らかにする必要がある。表を分析するために表モデルという概念を導入する。この表モデルを基にして、表生成システムを完成させる。本稿では、表モデルについて主に述べる。

Table model for table descriptions in Japanese documents

Yoshihiro MIURA, Zengo FURUKAWA, and Kazuo USHIJIMA

Department of Computer Science and Communication Engineering

Kyushu University

6-10-1 Hakozaki, Fukuoka 812, Japan

There are some systems printing out Japanese documents such as JT_PX and J-Star. They are not appropriate for printing Japanese documents of some kinds because they were transferred from those for printing English documents. In particular, it is difficult to print out Japanese documents with such tables as a personal history table. Most tables in English documents are lists of n-tuples of some items, however, tables in Japanese documents have composite structures. We need to develop a new system for printing tables in Japanese documents. Firstly we analyze tables in Japanese documents and construct a Table model for them. The table model consists of physical structure, layout information and entity.

1. はじめに

JTeX,J-Starなど日本語文書作成支援システムが存在する。しかしながら、これらは英語版を日本語化したものであるために、日本語文書に馴染まないところがある。とくに、表生成機能については、英語文書に出てくる単純な表は問題なく書けるが、履歴書のような日本語文書特有の表を書くには問題が多い。例えば、表のある枠の中にさらに表を割り付けたり、部分的に縦書きを含んだり、均等割り付けを行ったりすることが困難である。これらの問題点を解決するためには、日本語文書に合った表形式の生成方法を開発する必要がある。

また、現在の日本語文書作成支援システムでは、表の枠と中にはいるデータの記述が混在していて再利用が不便である。表枠やデータを再利用するには、それらを別々に取り扱う必要がある。

表は、単に枠と枠で囲まれた箱の集合ではなく、箱同士が関係を持ったり、箱の集合がより大きな箱を構成したり、というように構造を持っている。表形式の生成方法の開発には、このような、表の構造を明らかにする表モデルが必要である^[3]。これまで、文書出力のために文書モデルが検討されている^[7]けれども、表そのもののモデル化が充分になされているとは言い難い。

関係データベースは、関係を組で表現し、その関係が集まったものである。関係の間は、組を構成する要素によって関連づけられている。一方、表は、表の構成要素間に存在する関係だけでなく、関係の間の包含関係を階層的に表現することができる。このように、表モデルは、単に表出力のためだけでなく、情報の関連性についての1つのモデルとしても考えることが可能である。

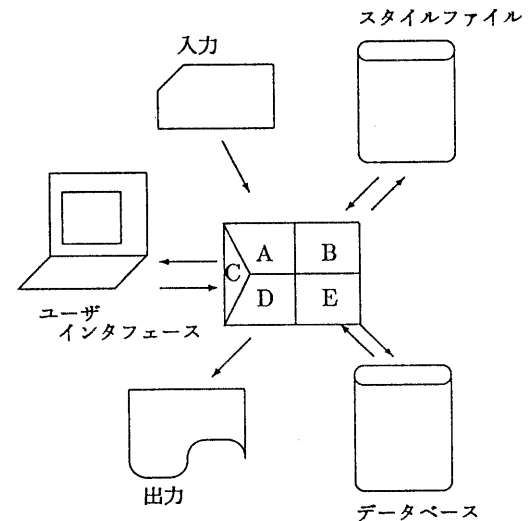
実際に、日本語文書にあった表を出力するためには、データの入力や出力、表の操作、表の格納を行なうための表生成システムが必要になる。表モデルは、利用者が表を把握するためのモデルであるので、表生成システムの操作性に影響を与えると同時に、計算機内部での表の格納方法にも影響を与える。

第2章では、表生成システムの概要を述べ、表モデルの役割について簡単に触れる。第3章では、表モデルを説明する。最初に、表モデルで用いる様々な用語を説明し、表モデルの体系を示す。第4章では、表モデルを具体例を用いて説明する。第5章で、結論として、まとめ、問題点、

今後の課題を述べる。

2. 表生成システム概要

表生成システムの目的は、入力を与えて表の出力を得るだけでなく、表の枠組や見出しをスタイルとして保存しておき繰り返し利用したり、表の内容をデータベース化して蓄えることができ、各種のスタイルで出力させることである。また、入力を対話型で行ない、操作性の向上を目指す必要がある。



- A: 入力処理部
- B: 書式管理部
- C: ユーザインタフェース部
- D: 出力処理部
- E: データ管理部

図1: システム構成

システムは、以下の5つの部分からなる。

- i) 入力処理部
表記述言語で書かれた入力を受けとり、一括して処理する部分である。
- ii) 書式管理部
入力から表のスタイルを抽出して保存したり、スタイルファイルに登録されている表のスタイルを取り出す部分である。
- iii) ユーザインタフェース部
対話の入力やグラフィカルな表示を行ない

利用者の入力負担を軽減する。WYSIWYG(What You See Is What You Get)方式とコマンド埋め込み方式との良いところを組み合わせた方式を採用する。これは、WYSIWYG画面とソース画面を同時に表示し、どちらからでも編集できる方式である[4]。

iv) 出力処理部

スタイルファイルとデータベースから生成した表を出力する。

v) データ管理部

表の中に入れるデータを取り出し、別に蓄えておいたデータを入力の一部にしたりする部分である。また、各データの関係演算を行なう。

我々は、このシステムの基本となる表の特性をモデル化した「表モデル」を構築しなければならない。表モデルの特徴を、以下に示す。

i) 表の構造を一意に表現できる

ii) 枠情報とその内容を別々に扱う

特徴のiは、単純な表と単純な表の組合せとなっている表とを同じ規則で構造が表せるということである。また、単純な表の構造を複雑な構造の表の一部にすることができるということを実現する。すなわち、単純な表を他の表の一部として割り付けることができるということである。特徴のiiに従って、書式管理部とデータ管理部を設計することができる。

3. 表モデル

3.1 用語の説明

表モデルを説明するために、表モデルで扱う用語についてまず説明する。

i) 「セル」

表は線によって紙面が一定の領域に区切られている。この線によって区切られた領域を「セル」と呼ぶ。セルは線によって区切られているので重なりがない。

ii) 「枠」

セルを囲む線の集合を「枠」と呼ぶ。枠は一般に4本の線からなっている。線の種類には、直線、点線、陰線がある。

セルとそれを囲む枠を合わせたものを箱と呼ぶ。表の中のセルは、それぞれ独立したものではなく、「見出し」と「内容」を構成している。見出しは、内容に対する表題を与えるものであり、内

容に対して縦についている場合と横についている場合がある。縦についている見出しを「縦見出し」、横についている見出しを「横見出し」と呼ぶ。(図2に見出しと内容の関係を示す。)つまり、見出しに対して縦、または横の属性を付けるということである。また、内容は各々を識別するための「識別子」と文字列そのものを表す「データ」とからなる。

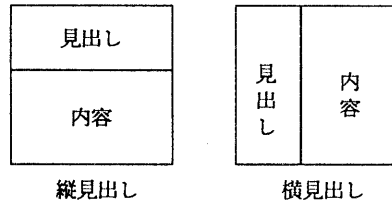


図 2: 見出しと内容

3.2 体系

表モデルの体系は、物理構造、レイアウト、エンティティの3つから構成されている。

3.2.1 物理構造

「物理構造」とは、表の中のそれぞれの見出し、内容がどのように配置されているかを表すものである。その配置の関係を表すために閉路のない有向グラフ(Directed Acyclic Graph: 以下、dagと呼ぶ。)を用いる。そのdagを説明するために、まず要素レコードを説明する。

「要素レコード」とは、縦見出し、横見出し、内容の識別子フィールドを持つレコードである。左から順に、縦見出し、横見出し、内容の識別子とする。見出しのないものは、“なし”と書くことにする。全ての箱は要素レコードとすることができる。図3において(a)の表には箱が5つある。その箱を全て要素レコードとすると(b)のようになる。ただし、実際は文字列自身ではなくその識別子が入るけれども、分かりやすくするためにここではデータ自身を入れている。

木の作り方を示す。

- i) 内容の見出し(縦見出しでも横見出しでも良い)が同じ要素レコードを選び出す。
- ii) 内容から見出しへ枝を引く。
- iii) 枝を引き終わったら、枝の先が内容のものを見出しのもの一段上の階層に置く。

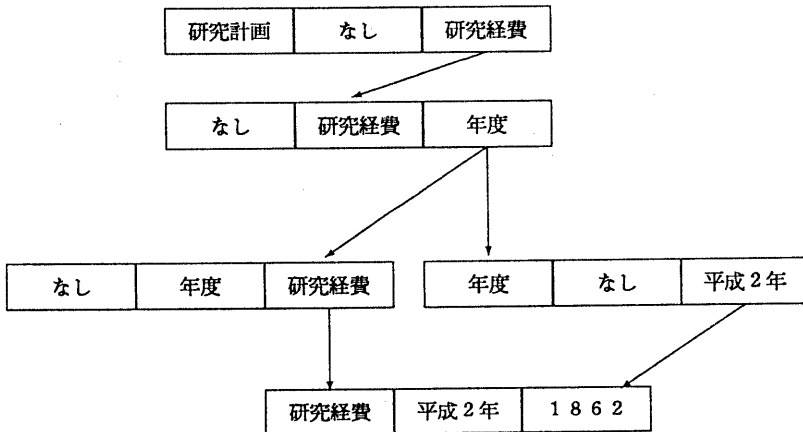


図 4: 要素レコードから dag へ

研究計画		
研究 経 費	年度	研究経費
	平成2年	1862
	⋮	⋮

(a) 表例題

縦見出し	横見出し	内容
研究経費	平成2年	1862
年度	なし	平成2年
なし	年度	研究経費
なし	研究経費	年度
研究計画	なし	研究経費

(b) 要素レコード例題

図 3: 要素レコード

iv) 縦見出しでつながっている場合は、枝に縦の属性を付ける。

図 3 の例で、iii) の要素レコードの処理が終了した時の有向グラフを図 4 に示す。iv) が終わった時の

図を図 5 に示す。この作業が終ると、内容が dag ができる。この dag の内容を項目と呼ぶ。一番上の項目を L1 項目 (Level1 項目) と呼ぶ。以下 L2, L3... と続く。一番 Level の大きい項目をデータ項目と呼ぶ (図 5 参照)。

3.2.2 レイアウト

レイアウトには、束縛を決める Bound- レイアウト (以下、B- レイアウトと呼ぶ) と数値を自由に決められる Free- レイアウト (以下、F- レイ

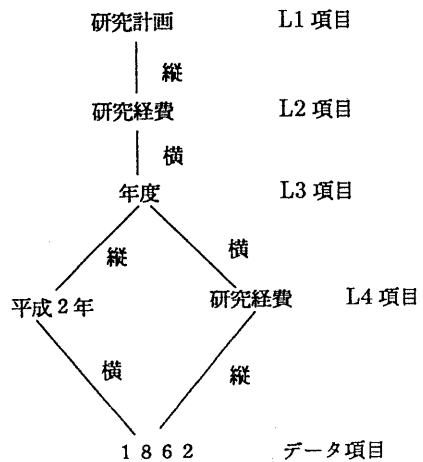


図 5: dag

ウトと呼ぶ)の2種類がある。まず、束縛と数値の決定について説明する。

- i) 束縛決定 — 2つ以上の対象についての関係を決めること
- ii) 数値決定 — 1つの対象についての数値を決めること

対象とは、セル、枠、箱、文字である。束縛の例としては、縦見出しの箱はその内容の箱と同じ幅でなければならない、枠の中に内容が収まらなければならないなどがある。数値の例としては、この箱の縦は5mm、横は20mm、などがある。すなわち、束縛とは、守られなければならない規則であり、数値とは、具体的数値のことである。

表を決定するには、まずB-レイアウトで各々の対象の関係を全て決定する。この時点では、まだ表は決定されておらず各々の対象の数値はある関係を規定しているだけである。その後、F-レイアウトによって対象の数値を具体的に決定することにより表を決定する。Fレイアウトで、決定できない場合や、矛盾が発生する場合などについては後に述べる。

i) B-レイアウト

B-レイアウトには、「物理構造による束縛」と「それ以外の束縛」に分けられる。

(1) 物理構造による束縛

「物理構造による束縛」は、前述した物理構造による束縛である。具体的には、以下の2つがあげられる。

- 内容に対して縦見出しは、見出し、内容の箱は同じ幅でなければならない(図6参照)。
- 内容に対して横見出しは、見出し、内容の箱は同じ高さでなければならない(図6参照)。

これを前述のdagの用語を用いて説明すると、縦という節を経由してつながっている項目は親の項目の幅を継承し、横という節を経由してつながっている項目は親の項目の高さを継承することを意味する。図3の例における継承を付記すると図7のようになる。

(2) その他の束縛

その他の束縛には、物理的束縛とユーザが決定できる束縛の2つがある。

a) 物理的束縛

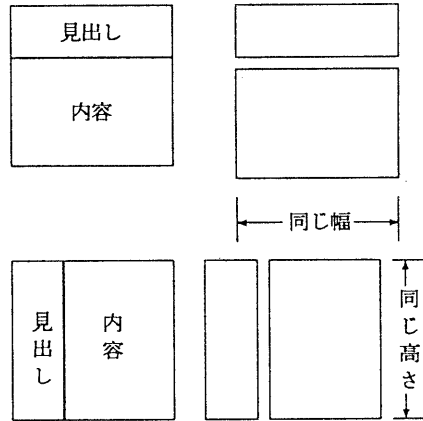


図6: 物理構造による束縛

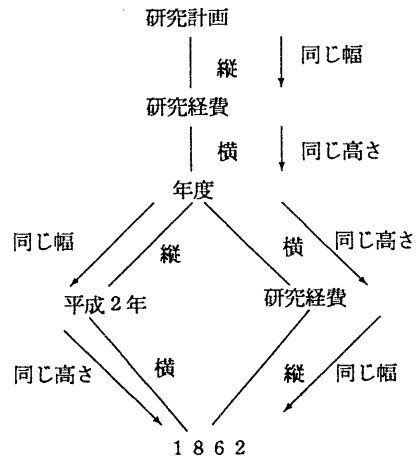


図7: 継承

物理的束縛は、具体的に以下の2つがある。

- 内容は枠内に収まらねばならない。
- ページの中に収まらねばならない。

b) ユーザが決定できる束縛

ユーザが決定できる束縛には、いろいろなものが考えられる。以下に例を3つを示す。

- 同じレベルは均等に分割する。
- 同じレベルは内容の文字数に比例して分割する。
- 内容の割り付け(左、右よせ、中央、均等割付け)。

ii) F-レイアウト

F-レイアウトは、ユーザが自由に決められるレイアウトである。具体例を以下に3つ示す。

- 文字のフォント
- 各々の箱の幅や高さ
- 完成した表の大きさ

ただし、F-レイアウトはB-レイアウトで規定された束縛を満足しなければならない。

3.2.3 エンティティ

エンティティとは、1つの単位として参照できる文字の集まりである。前述の物理構造、レイアウトのエンティティと内容のエンティティの3種類がある。

i) 物理エンティティ

物理構造で示した dag の一部、および全部を物理エンティティという。dag 全体の場合は、特に「スタイル」と呼ぶ。

ii) レイアウトエンティティ

レイアウトエンティティには、B-レイアウトエンティティとF-レイアウトエンティティの2種類がある。

(1) B-レイアウトエンティティ

B-レイアウトでよく使用する束縛を1つのエンティティとして定義しておき、他の表を書く時にそのエンティティを使用すれば記述量を大幅に削減することができる。

(2) F-レイアウトエンティティ

F-レイアウトでよく用いるレイアウトを1つのエンティティにしておけば、他の表を書く時にそのエンティティを利用することができる。

iii) 内容エンティティ

内容を表すために、識別子とデータの結び付きをエンティティにしておけば、枠を固定して幾つものデータを入れた表が簡単に作れる。

4. 表の例

この章では、具体例を用いて表モデルを説明する。例として図8の表を出力したいとする。まず、その入力を示し、エンティティとして考えられるものを示し、最後に実際に用いて明らかにした問題点について述べる。

4.1 入力

入力として必要なものは以下の通りである。

i) 物理構造 — 図9に示す識別子の dag

ii) B-レイアウト

- A 4のサイズに収める
- 同じレベルは内容の文字数に比例して分割する
- 内容は全て中央に割り付ける

iii) F-レイアウト

- 文字は 10pt
- 表の横幅は 60mm

iv) 内容

- L1 項目 — 出張申請
- L2 項目 1 — 出張者名
- L2 項目 2 — 出張月日
- L2 項目 3 — 出張場所
- L2 項目 4 — 用務内容
- L2 項目 5 — 旅費種類
- L2 項目 6 — 金額
- L2 項目 7 — blank(空白)
- L3 項目 1 — 三浦好弘
- L3 項目 2 — 自 1991年5月23日
至 1991年5月25日
- L3 項目 3 — 神奈川県川崎市
- L3 項目 4 — コンピュータネットワーク研究会に出席
- L3 項目 5 — 計算機ソフトウェア基金
- L3 項目 6 — blank(空白)

B-レイアウトとF-レイアウトに関しては、利用者がどこまでこだわりを持つかによって変わる。とにかく紙に収まって、見て分かれればよいという程度であれば、B-レイアウトは紙の大きさだけを指定すればよい。1つ1つの箱の大きさまでこだわるのであればF-レイアウトで箱の大き

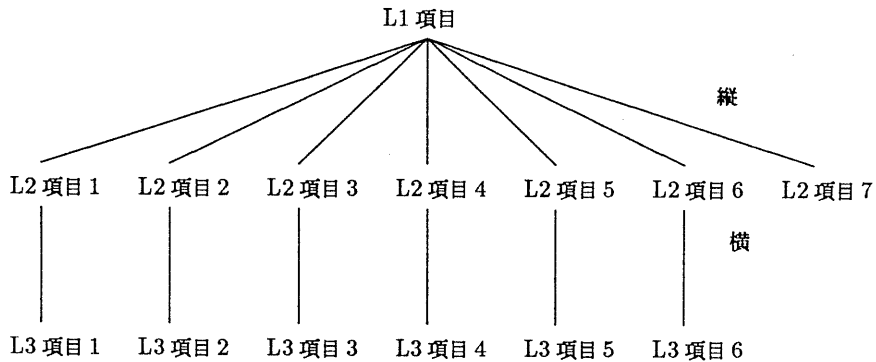


図 9: 物理構造

さまで指定する必要がある。また、内容は識別子とデータの結び付きを指定する。訂正をする場合に従来は文字列毎に書き直す必要があったが、今回の表モデルでは、結び付きを変更することで訂正が可能となる。識別子は、レベル分けされているのでレベル単位での訂正が可能である。

4.2 エンティティ

エンティティに関しては、図9の dag 全体を物理エンティティとして定義すれば、出張申請表式というスタイルになる。また、L1項目、L2項目1～L2項目4、L3項目1～L3項目4までの dag の一部を物理エンティティとして定義しておき、別の dag の一部とすることにより別の表の物理構造の一部とすることができる。すなわち、出張報告の一部を別の表の一部として割り付けることができる。その場合、各内容の識別子のレベルは新しい dag によって書き換えられる。B-レイアウト F-レイアウトの全てをレイアウトエンティティとして定義すれば、同じ条件で表を書く際にそれを使用することができる。内容、つまり識別子と文字列のつながりをエンティティとして定義しておけば、他の表式で出力する際にそのエンティティを使用できる。

4.3 問題点

実際に例を記述することによって、いくつかの問題点が明らかになった。以下にその問題点を表モデルの問題点とシステムの問題点とに分けて示す。

i) 表モデルの問題点

出張申請

出張者名	三浦好弘
出張月日	自1991年5月23日 至1991年5月25日
出張場所	神奈川県川崎市
用務内容	コンピュータネットワーク研究会に出席
旅費種類	計算機ソフトウェア基金
金額	

図 8: 出張報告

- F-レイアウトで、数値指定ではなく範囲を指定したい場合がある。

ii) システムの問題点

- レイアウトの際に指定されていない数値をどう決めるか
- レイアウト間(B-レイアウト間、あるいはF-レイアウト間、B-レイアウトとF-レイアウトの間)が矛盾する場合どう対応するのか
- エンティティの具体的な実現方法
- 識別子の dag の入力方法

表モデルの問題点は、範囲を指定する場合、それは束縛になるのでB-レイアウトに入れることによって解決できる。システムの問題点は、表生成システムを完成させるために解決しなければならない問題である。最初の2つについては、利用者の要求によるところが大きいので、利用者に関わり合せて処理をするという方法を考えている。

5. おわりに

日本語文書における表の特性を分析して表モデルを構築した。表モデルは、表を出力するための表生成システムの中核をなす概念であり、システムの操作性に影響するものである。表モデルは、要素の配置関係を表す物理構造と、要素間の束縛関係を表すレイアウトと、要素や物理構造やレイアウトを表現するためのエンティティから構成されている。

現在、表モデルにしたがって表を記述するための言語の設計と出力方式の検討を行なっている。

表を例にとった場合に明らかになった問題点として以下のようなものがある。

- 数値が指定されていない場合がある。
- レイアウトが矛盾することがある。
- エンティティの具体的な実現方法をどうするか。
- 識別子の dag の入力方法をどうするか。

表システムの問題点としては以下のものがある。

- 表生成システムはこの5つの部分だけで良いか。
- 入力部分は言語での入力だけで良いか。
- データを蓄えるのにデータベース部分が本当に必要か。

今後は、前に述べた問題点の解決、表生成システムの完成を行なう。

参考文献

- [1] Donald E.Knuth.The TeXbook, Addison-Wesley Publishing Co.,1984.
- [2] Leslie Lamport.L^AT_EX—A Document Preparation System, Addison-Wesley Publishing Co., 1986.
- [3] Bryan, M:SGML An Author's Guide, Addison-Wesley Publishing Co., 1988.
- [4] Kenneth P.Brooks:Lilac:A Two-View Document Editor, COMPUTER, Vol24, No.6, pp.7-19, Jun 1991.
- [5] 山川, 川端, 田村:OA 業界からみた DTP, 情報処理, Vol31, No.11, pp1508-1517, 1990.
- [6] 高橋延匡:大学の研究室における DTP の研究開発, 情報処理, Vol31, No.11, pp1518-1528, 1990.
- [7] 上林憲行:文書エディタの現状と将来展望, 情報処理, Vol31, No.11, pp1535-1542, 1990.
- [8] 伊藤和人:L^AT_EX トータルガイド, SHUWA SYSTEM TRADING CO., LTD., 1991.
- [9] A.V.Aho, R.Sethi, Compilers(「コンパイラ」 J.D.Ullman 原田賢一訳), サイエンス社, 1990.
- [10] 三浦好弘, 古川善吾, 牛島和夫:日本語文書出力方式について, 情報処理学会第 43 回全国大会講演予稿, Vol3, pp305-306, 1991.
- [11] 三浦好弘:日本語文書における表生成方式の開発, 九州大学工学部情報工学科卒業論文, 1991.