Vol. 30 No. 9 情報処理学会論文誌 Sep. 1989

文書の論理構造を備えた日本語清書システム「浄書」の設計と実現†

里山元章†† 中川正樹††† 高橋延匡†††

本論文は，高品質な文書出力と高度な文書清書機能を持つインテリジェント清書マシン「浄書」(JOSHO: Japanese Output Server with HOspitality) の設計と実現について述べたものである．浄書は，レーザビームプリンタをインテリジェント化し，清書機能や文書交換機能を持たせている．このため，パソコンやワードプロセッサの清書マシンとして利用できる．浄書には，日本語フォーマッタや英文フォーマッタ，リスティングツールなどのソフトウェアが用意されているが，本論文ではそれらのうち，特に日本語フォーマッタについて述べている．浄書の持つ日本語フォーマッタは，章や節，見出し，図といった文章の内部構造を書式化に反映しており，それらの文書構造に対して書式を設定するようになっている．また，和欧混合組版規則に基づく行頭，行末，分離禁則処理やスペーシング処理を実現している．浄書はすでに研究室において実用され，論文や研究報告書，プログラムとそのドキュメント，議事録などの印刷に利用されている．

1. まえがき

計算機を文書作成に利用し始めたのは，プログラム作成やその仕様書作成のためであった．計算機科学に従事する研究者は，マニュアルや仕様書を計算機で管理する目的や自らの論文や報告書を秘書にタイプしてもらうコストを軽減するためあるいは，印刷屋に頼めないような文書を作成することを目的として，さまざまな文書作成システムを作り上げた．troff/nroff[1]，TEX[2]，Scribe[3] などは有名な例である．それらのいくつかは，日本語化されている[4]．

我々の研究の長期的な目的は，日本の文化に根ざした計算機による文書作成環境の実現にある．その環境では，論文もプログラムも仕様書も同一の環境下で扱われるべきであると考えている．このような文書作成環境の実現のためにオペレーティングシステムのレベルから研究を行っている．

その研究の一環として我々は，『文書を美しく仕上げることができるという喜びは我々の創作意欲をそそるものである』というモットーのもと，高度な日本語文書清書機能を持つインテリジェントプリンタの開発を1979年より始めた．これを「浄書」(JOSHO: Japanese Output Server with HOspitality)[5] と呼んでいる．浄書は，単なる印字機能だけでなく，高度な組版処理を行うフォーマッタをプリンタ上に置くといった点で他のプリンタシステムや清書システムと異なる．

本研究の具体的な目的は，次のようなプリンタを開発することである．

（1） 文書の論理的な構造を反映した書式設定機能や和欧混合組版規則に基づく禁則処理機能を持つ日本語フォーマッタを備えたプリンタ

（2） プログラムやドキュメントを扱うソフトウェア工学的な研究など，文書出力を必要とする研究の出力システムとして利用できる柔軟性と，ソフトウェアによる高い付加価値を持つプリンタ

最近，電子出版システムが注目され，米国で開発された幾つかのシステムが日本語化されている．我々は，ソフトウェアの多くが，使われる国の文化に大きく依存すると考えている．浄書は，わが国の伝統的な組版処理を重視して設計したオリジナルなシステムである．

浄書の開発の経緯について次に述べる．

我々は，1983年10月にフォーマッティングマシンの基本構成を示し[6]，1986年7月に，本論文で提示したシステムの基本仕様の実現を報じ[10]，同年11月，和欧混合組版規則をインテリジェントソフトウェアとして組み込んだ[11]．

1983年における我々のフォーマッティングマシンの発表の方が，PostScript (1984年8月発表) や LaserWriter (1985年1月発表) よりも早い[12]．また，和欧混合組版規則を文書整形に組み込んだのも我々が最初

† Japanese Formatting System JOSHO and Its Formatting Strategy Based on the Logical Structure of a Document by MOTOAKI SATOYAMA (Systems Development Laboratory, Hitachi, Ltd.), MASAKI NAKAGAWA and NOBUMASA TAKAHASHI (Department of Computer Science, Faculty of Technology, Tokyo University of Agriculture and Technology).

†† (株)日立製作所システム開発研究所
††† 東京農工大学工学部数理情報工学科

である.

2. フォーマッティングマシンのアーキテクチャ

高印字品質を実現するため，高解像度で柔軟性の高いプリンタ上で高度な文書出力をユーザが行うには，多くのプログラム開発をする必要がある.

現在，幾つかの電子出版システムやワークステーションによって高品質な文書が得られるようになった.しかし，こういった高価なシステムを使わず，安価なワードプロセッサで作成した文書をユーザ自身の手で手軽に，なおかつ，写植のようにきれいに出力したいといった要求もある.我々はフォーマッタをプリンタに持たせた.この機能分散により入力編集を行うシステムに依存しない清書機能の実現が可能になる.

また，プリンタの解像度を上げれば上げるほどデータの転送量の増大などからシステムの負荷が大きくなるといった問題もこの負荷分散により軽減される.

我々はこのようにインテリジェント化したプリンタに清書機能を持たせたものをフォーマッティングマシンと呼んでいる.次にフォーマッティングマシンの利点をあげる.

（1） 高機能プリントサーバとして種々の計算機から利用可能であること

（2） ホスト側に負担をかけずに高解像度プリンタの性能を生かせること

（3） プリンタ上で資源を共有できること

（4） データのコード化による転送量の減少と高い機種独立性を保てること

（5） ソフトウェアによる高い付加価値をプリンタに持たせることができること

さらに，フォーマッティングマシン用の言語で表現された文書ファイルを中間ファイルとして会話的に生成できるソフトウェアの実現が望まれるが，本稿では先に実現したプリンタ側のソフトウェアについて述べる.なぜなら，日本において普及している日本語ワードプロセッサや，パーソナルコンピュータ上のワープロソフトなどから利用できるフォーマッティングマシンを実現することの方が現実的であり，まず手掛けるべきことであると考えるからである.特に我々のプログラム開発環境をモデルに考えた場合には，この方式の利点は大きい.

3. 日本語フォーマッタの設計目標

日本語フォーマッタは，フォーマッティングマシンの中核をなすアプリケーションソフトウェアである.次に日本語フォーマッタの設計目標をあげる.

（1） 文書構造を反映した書式化機能

一般に書式化の機能と言えばページのサイズ，上下左右の余白，段組数や行形式などといった用紙やページの物理的要素に対する機能であった.本日本語フォーマッタでは，文書の論理構造に対する機能を加える.つまり，章，節，項，脚注，ノンブル，柱，見出し（タイトル）といった文書の要素に対しても書式の設定を行えるようにする.また，設定された書式に従って，章番号，図番号や目次の自動生成，索引，参考文献などの出力を可能にする.

（2） 文書のきりばり機能

文書の編集中，“きりばり”の対象（例えば図）の挿入削除を繰り返しているうちに，それを参照している文から遠く離れてしまうことがある.この“きりばり”の対象をできるかぎり自動的にレイアウトする機能を持たせる.

我々は“きりばり”の対象を“箱”と呼んでいる.図，表などのように他の部分を構成する文字列と区別され固定した領域を持つものや，新聞の見出しのように特殊な組み方（これを箱組という）をするものを箱として出力する図形整形言語などを実現する.

（3） 禁 則 処 理

和欧混合組版に従ったスペーシング処理，行頭，行末，分離などの禁則処理，プロポーショナルピッチ，右ぞろえ（right justification）を行う.

（4） 文書作成支援機能

作成した文書中に現れた各文字の頻出度，表記のゆれなどを調べる機能や，作成年月日，作成者名，文書ファイル名，その他コメントなどを出力する機能があると便利である.また，参考文献をデータベース化し，そこから引き出せるようにする.こういった文書の作成，管理を支援する機能を設ける.

4. システム構成

我々が開発用に実現したフォーマッティングマシンを特に浄書マシンと呼ぶことにする.

4.1 ハードウェアシステムの構成

浄書マシンは，レーザビームプリンタをマイクロプロセッサ MC 68000 によってインテリジェント化を

行った．これにはフロッピディスクコントローラ（2台），2Mバイトのメモリボード2枚，ビデオインタフェースボード，漢字発生ボード（32×32，24×24 dotの2枚．2枚ともJIS第1水準，第2水準，JIS非漢字577種を備えている）を実装した．メモリボードの1枚はフレームメモリとして使っている．これは紙1枚分の画素データを展開できるように用意されたメモリである．

4.2 ソフトウェアシステム構成

4.2.1 オペレーティングシステム

我々は浄書マシンの開発と並行して，OS/oという独自のオペレーティングシステムの開発を進めている[6]．現在の浄書マシンはCP/M-68Kを開発用オペレーティングシステムに採用しているが，日本語の扱いなどで不都合が多いため，日本語オペレーティングシステムであるOS/oへ移行することになっている．このため，浄書マシンの標準的な入力ファイルはOS/oのコードである全2バイトコード（JIS X 0208（旧C6226）で規定される漢字コード．ただし，制御コードは先頭バイトにNULを付加し，2バイトに整合している）としている．

フレームメモリ，漢字フォント発生ボードおよびレーザビームプリンタなどのハードウェア資源を制御する機能が浄書マシンのオペレーティングシステムに要求される．これらのハードウェア資源にアクセスするためのドライバを作成した．これをFMH（Frame Memory Handler）と呼ぶ．これを利用しやすくライブラリ化したものをFML（Frame Memory Library）と呼ぶ．FMLは座標変換，文字フォントや線の描画，書体の変更などの機能を持つ関数で構成される言語Cをサポートするライブラリである．

```
@ページ設定（B5;縦長横組;1;60m;60m;40m;30m;0;片@）
@章設定（0;14;細明;0.3m;6m@）
@節設定（0;14;細明;0.3m;6m@）
@ノンブル設定（常下中;10;細明;0.2m@）
@頭書設定（常;左;10;細明;0.3m@）
@副頭書設定（常;右;10;細明;0.3m@）
@脚書設定（常;中;10;細明;0.3m@）
@要旨設定（0;10;細明;0.2m;5m@）
@章見出（　;左;14;太明;0.3m;5m;0;3;0;無@）
@節見出（　;左;14;太明;0.3m;5m;0;0;0;無@）
@　　（　;中央　　0.3m　　1;無@）
```

図1（a） 浄書の書式ファイル
Fig. 1 (a) An example of JOSHO format file.

FMLは仮想的な座標系と単位系によるインタフェースを持ち，浄書マシン上のアプリケーションプログラムの機種独立性を高めている．また，FMLにはタイマ機能も用意した．

4.2.2 文書変換システム

漢字コードは，利用するホスト側計算機によって異

```
@要旨（梗概@）@副頭書（梗概@）
@頭書（浄書の出力例@）
@ノンブル（情報処理学会論文予稿　No.@［頁］@）
　本論文は，高品質な文書出力と高度な文書清書機能を持つ清書システム「浄書」（JOSHO：Japanese Output Server with HOspitality）の設計と実現について述べたものである．浄書は，レーザービームプリンタをインテリジェント化し，清書機能や文書交換機能を持たせている．このため，パソコンやワードプロセッサの清書マシンとして利用できる．浄書には，日本語フォーマッタや英文フォーマッタ，リスティングツールなどのソフトウェアが用意されているが，本論文では日本語フォーマッタについて述べた．
@和　浄書の持つ日本語フォーマッタは，章や節，見出し，図といった文章の内部構造を書式化に反映しており，それらの文書構造に対して書式を設定するようになっている．また，和欧混合組版規則に基づく行頭，行末，分離禁則処理やスペーシング処理を実現している．
@和　浄書はすでに研究室において実用され，論文や研究報告書，プログラムとそのドキュメント，議事録などの印刷に利用されている．
@章（まえがき@）
　計算機を文書作成に利用し始めたのは，プログラムの作成の他にそのドキュメントの作成のためであった．欧米において，計算機科学に従事する研究者は，マニュアルや仕様書を計算機で管理する目的や自らの論文や報告書を秘書にタイプしてもらうコストを軽減するためあるいは，印刷屋に頼めないような文書を作成することを目的として，様々な文書
"　　・も作り上げ　　"大学のD.E.　　Knut
```

図1（b） 浄書のソースファイル
Fig. 1 (b) An example of JOSHO source file.

浄書の出力例　　梗概

梗概

本論文は，高品質な文書出力と高度な文書清書機能を持つ清書システム「浄書」（JOSHO：Japanese Output Server with HOspitality）の設計と実現について述べたものである．浄書は，レーザービームプリンタをインテリジェント化し，清書機能や文書交換機能を持たせている．このため，パソコンやワードプロセッサの清書マシンとして利用できる．浄書には，日本語フォーマッタや英文フォーマッタ，リスティングツールなどのソフトウェアが用意されているが，本論文では日本語フォーマッタについて述べた．

浄書の持つ日本語フォーマッタは，章や節，見出し，図といった文章の内部構造を書式化に反映しており，それらの文書構造に対して書式を設定するようになっている．また，和欧混合組版規則に基づく行頭，行末，分離禁則処理やスペーシング処理を実現している．

浄書はすでに研究室において実用され，論文や研究報告書，プログラムとそのドキュメント，議事録などの印刷に利用されている．

まえがき

計算機を文書作成に利用し始めたのは，プログラムの作成の他にそのドキュメントの作成のためであった．欧米において，計算機科学に従事する研究者は，マニュアルや仕様書を計算機で管理する目的や自らの論文や報告書を秘書にタイプしてもらうコストを軽減するためあるいは，印刷屋に頼めないような文書を作成することを目的として，様々な文書作成システムを作り上げた．スタンフォード大学のD.E.Knuthが，自分の著書の印刷を印刷屋に任せることに我慢ができずTEXを作った話は有名である．

情報処理学会論文予稿 No. 1

図1（c） 浄書の出力例
Fig. 1 (c) An example of JOSHO output.

なることが多い．そこで，独自のアーキテクチャを保ちつつ，プリントサーバとしてさまざまな計算機システムから利用できるようにするため，統一した文書変換を行う文書変換システムが必要になる．

文書変換システムは，ホスト側のファイルを浄書マシンの標準ファイルに変換する．JIS X 4001 (旧 C 6237) で規定されるファイル仕様の文書ファイルから浄書マシン内部のファイル仕様への変換，および，現在，多くのパーソナルコンピュータが採用している MS-DOS からの文書変換を実現した．これらの変換はファイル形式とコードの変換であるが，エスケープシーケンスなどによる文字修飾などの拡張表現は，変換ツールによって日本語フォーマッタの指令語に変換することができる．

4.2.3 アプリケーションソフトウェア

日本語フォーマッタ，英文フォーマッタやプログラムのソースリストを出力するツールなどである．ユーザが FML を利用して作成した出力ツールも含まれる．

5. 日本語フォーマッタの実現

5.1 概　　要

図 1 (a)に本日本語フォーマッタの書式ファイルの例を，図 1 (b)にソースファイルの例を，図 1 (c)にその出力例を示した．これらの図に示すように本日本語フォーマッタでは，文書を構成する要素の範囲を指定し，その要素の書式をまえもって設定しておくことで書式化を行っている．本日本語フォーマッタが実現した組版処理規則は文献 7) を参考にした．

ソースファイルはテキストと指令語で構成され，書式ファイルは書式設定の指令語をまとめたものである(書式設定の指令語がソースファイルのどこに現れてもかまわない．最後に現れたものが有効になる)．

5.2 指　令　語

指令は，@で始まる文字列である．@をテキストとして出力する場合は@を重ねる．

指令語には大きく構造指定指令，文字列修飾指令，書式設定指令，印刷制御指令に分類できる．

5.3 文書構造木の生成

最初にすべてのソースファイルを読み込み，構造指定指令に従って**図 2** に示したような文書構造木の構造部と実体属性部を生成する．この文書構造木によって，出力される文書の書式の適用範囲が決まる．

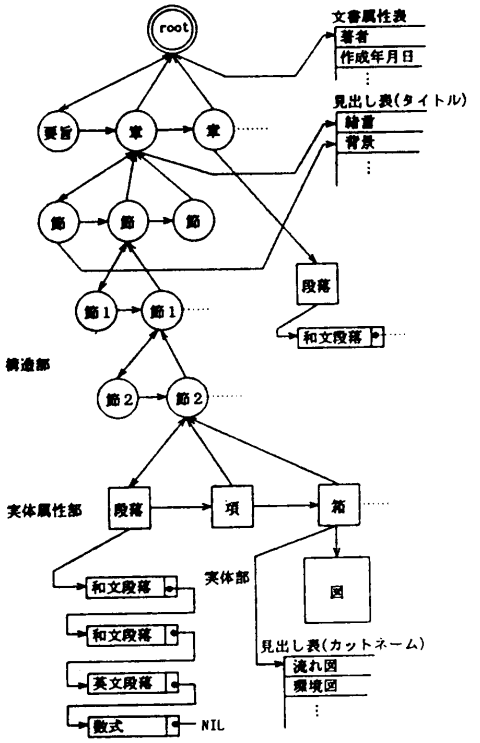

図 2 文書構造木の構成
Fig. 2 Data structure for the logical structure tree of a document.

(1) 構造部と見出し表

構造部は，文書構造上の論理的な階層構造を表したものである．例えば章の始まりに次のような構造指定指令がくる．

@章 (緒言@)

(@)の中は見出しである．階層構造の指定には，“まえがき”，“要旨” や “節” などを用意した．見出しは，章題などのタイトルであり，すべて見出し表に登録し，目次の出力や章番号の参照に利用する．

(2) 実体属性部

文書には中央ぞろえ（センタリング）や脚注といったような様式の異なる文書を出力する場合がある．また，図や表のように固定した領域を持つものもある．

このような文書を出力形態で分けて一塊にしたものを実体属性部という．例えば実体属性部指定には次のような構造指定指令がくる．

@図 (流れ図@) {;; 100m; 100m; @}

@脚書 (情報処理学会@)

“流れ図” は図のカットネームであり，見出し表に登録される．“情報処理学会” は脚書として出力する．{ @} の中は {x 座標; y 座標; 横のサイズ; 縦のサイズ; 拡張用@} である．この例の場合，座標は無

指定なのでフォーマッタ側で適当な位置に出力する(5.6 節参照).

(3) 実 体 部

実体部は段落か箱の実体部で構成される. 実体部指定の構造指定指令は, 例えば, 以下のように行う.

@和 段落を指定する.

また, 文字列は次のような文字列修飾指令で文字修飾も指定できる.

@下線/@級"10"/アンダラインを引く @/@/

段落は, 段落を構成する文字列が和文か英文か数式かを識別するコード, テキストバッファへのポインタ, 次の段落へのポインタで構成される. テキストバッファは, 文字列と各文字の属性を表す文字属性表へのポインタから構成される. 文字属性表は文字の書体コード, 修飾コード, 相対座標などで構成される.

箱の実体部は図や表などの識別コード, 座標やサイズ, カットネーム表へのポインタなどで構成される.

5.4 書式表・印刷指定表

書式指定指令によって書式表が生成される. 書式表は, 要旨, 章, 節といったすべての文書の階層クラスや頭書, 脚書, 脚注, ノンブル, 図, 表といった実体属性, 章や節のタイトル, 図や表のカットネームといった見出しなど, それらにあった書式が(例えば, タイトルは前後の行あきの指定など)別個に生成される.

5.5 行の書式化処理

行の書式化処理とは, 段落を禁則処理を施しながら行に分割していく処理である. 次に述べる処理により, 和欧混合組版規則に基づく行の書式化を実現した.

5.5.1 行の印字領域の管理

段落を行に分割するまえに, 図表がページ上に配置されていたり, 2段組みが指定されている場合などの条件を考慮し, 行のサイズと位置が計算される. このとき, 袋閉じや見開き印刷のために2ページ分の印字領域を管理している. 例えば, 図3のようなページが構成された場合, 図の数字の順序で行を展開する. この展開アルゴリズムを図4に PAD で示す.

5.5.2 スペーシング処理

行の長さと位置が決まると, 段落を行に分割する. この分割は段落を構成する文字の幅を計算し, その合計が行長より長くなる前の文字で行われる. 文字幅とその合計の計算をスペーシング処理という.

スペーシングは, ひとつ前に印字した文字とこれから印字しようとする文字の組合せによって決まる. そこで, 文字を幾つかのカテゴリに分類し, カテゴリの組合せでスペーシング処理を決める行列を考えた. これをスペーシング行列と呼ぶ. スペーシング行列のカテゴリは〈漢字〉, 〈英数字〉, 〈空白〉, 〈記号〉, 〈句読点〉, 〈起こしの括弧〉, 〈受けの括弧〉, 〈行頭記号〉の8種類に分類される. 行頭記号は, 行バッファの先頭に置かれる特別な記号で, 行の先頭であることを示す. 例えば〈英数字〉と〈英数字〉の組合せはプロポーショナルピッチを施し, 〈受けの括弧〉と〈句読点〉は, 半角詰めるなど, スペーシング処理を分類するのに使う. プロポーショナルピッチが必要な場合は, プロポーショナルピッチが格納された英数字と英数字の組合せによる2次元配列から得る.

表1にスペーシング行列を示した. 表1の〈半〉では, 例えば"」"に"."が続くとき, ベタ詰めにする. 〈頭〉では, 例えば"「"が行頭にきても, 左端

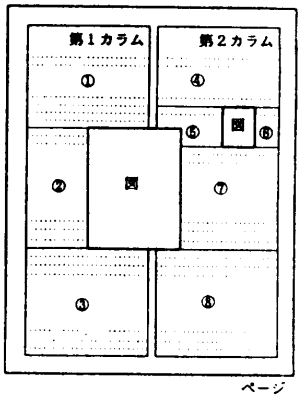

図3 ページ構成の例
Fig. 3 An example of page layout.

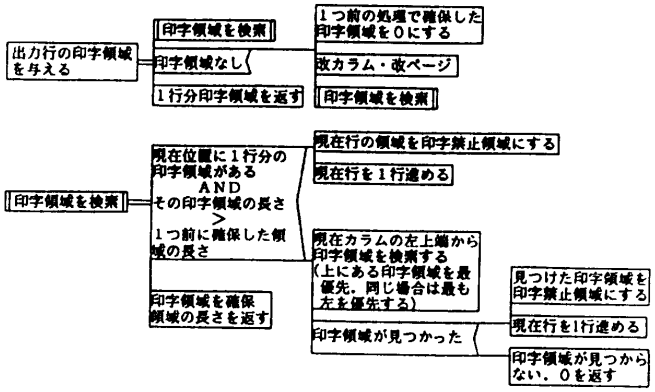

図4 印字行の決定アルゴリズム
Fig. 4 Algorithm to determine next line position.

に余分なアキがでないようになる．

5.5.3 禁則処理と右そろえ

段落を行に分割するためには，さらに禁則処理と右ぞろえの処理が必要である．ここでは，和欧混合組版規則に基づく禁則処理とその実現方法について述べる．

禁則処理は，行頭禁則処理，行末禁則処理，分離禁則処理の三つに分けられる．

本日本語フォーマッタでは，行頭禁則文字として，前節で分類したカテゴリの中の〈受けの括弧〉と〈句読点〉，および，〈記号〉のカテゴリから，促音記号，コロン，セミコロン，疑問符，中黒，感嘆符（JIS コード 16 進で，2126～2145）など合計 47 種類を設定している．また，行末禁則文字として前節で分類したカテゴリの中から，〈起しの括弧〉を設定した．

分離禁則は，リーダ（…）ダッシュ（—）どうしの場合や組合せ数字の場合（03 や 0423 など），連数字（86,000 や 9 億 7,000 万など）の場合，小数点の前後（96.58，3.14 など）の場合，記号（kg，mm，sec，No，$，¥，%，@ など）と数字の組合せの場合などに適用される．

本日本語フォーマッタでは，分離禁則処理を拡張し，行末禁則文字は，その次の文字と分離禁止とし，行頭禁則文字はその前に印字する文字と分離禁止とすることで，行頭，行末禁則処理を実現した．

図 5 に行の禁則処理と右ぞろえ処理を PAD で示す．

5.6 箱の処理

図や表といったものは箱として処理される．例えば図の場合，まず，ページ上のどの位置に配置するか計算する．次にカットネームに番号を振って出力し，最後に図の内容を処理する．図形の処理は現在，開発を進めている．

箱を配置する座標が指定されていない場合，自動的に適当な領域に展開する．この場合，箱は指定指令が現れた位置の最も近くで，展開するのに十分な広さを持つ領域にレイアウトされる．ただし，箱の横幅がカラム幅より狭い場合，カラムの右端にそろえ，逆の場合，カラムの左端にそろえる．

表 1 スペーシング行列
Table 1 Matrix for spacing.

	行頭	起括弧	受括弧	句読点	英数字	記号	漢字	空白
行頭	ERR	頭	ERR	ERR	0	0	0	0
起括弧	ERR	半	全	全	プ	全	全	全
受括弧	ERR	半	半	半	プ	半	全	全
句読点	ERR	半	半	半	全	全	全	全
英数字	ERR	プ	プ	プ	プ	プ	全	半
記号	ERR	全	全	全	プ	全	全	全
漢字	ERR	全	全	全	全	全	全	全
空白	ERR	半	全	全	半	全	全	全

（一つ前に印字したカテゴリを縦に，これから印字するカテゴリを横に並べている）

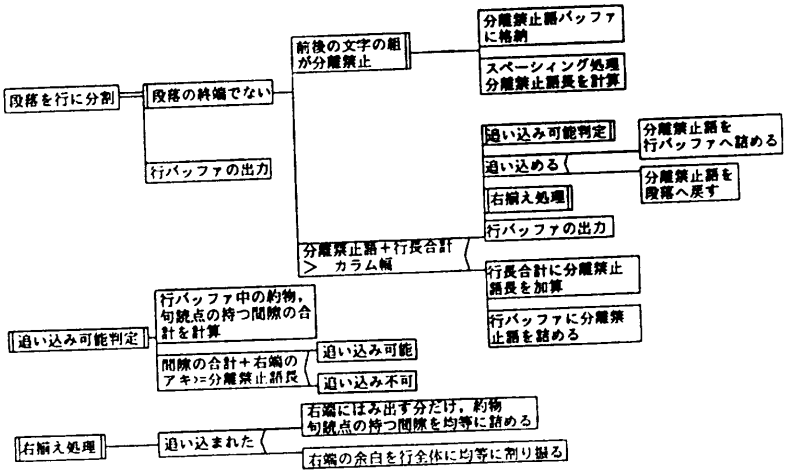

図 5 禁則処理と右ぞろえ
Fig. 5 Japanese typesetting and right-justification.

$$\sum_{n=1}^{\infty} \frac{n}{\sqrt{n+1}}$$

図 6 数式の出力例
Fig. 6 An example of arithmetic expression formatting.

5.7 数式の処理

数式は次のような数式整形言語によって記述する．

@数 Σ from *n*=1 to ∞ [*n* over sqrt *n*+1]

このように指定すると，図 6 のような出力が得られる．

5.8 組版処理に関する問題

実現した組版処理で問題になった点を次にあげる．

（1） 文字だけで分離禁則が判定できない場合

（2） 大きな箱が連続して何個も指定され，次々に後ろのページに追い出された場合

（1）の問題は，例えば円周率「3.14」の点が，小数

点なのかピリオドなのかといった判定である．たまたま，前のセンテンスが数字で終わり，次のセンテンスが数字で始まる場合と判定できない．また，「ROM 8個で構成する.」という文の場合，ROMと8は分離禁止にする必要はなく，逆にROMと8の間はアキをとるのが普通である．ところが，ROMを8個なのか，ROM 8という単語なのか判別できない．これらを判別するには，文脈や単語の意味を情報として持っている必要がある．これは，今後の研究課題となる．

(2)の問題は，人間がレイアウト処理を行っても生じる問題である．箱の外の部分にあたる文字列を移動して，箱の再配置処理を行うと，例えば図の場合，図が存在するページより後方のページに図を参照する文がきてしまう可能性がある．また，図と図を参照する文の最適な位置関係を計算する処理の自動化は不可能に近い．このため(2)の問題については，自動的なレイアウトは目安として使い，利用者がレイアウトを考え，箱の印字座標を指定するようにした．そして，自動レイアウトの結果をディスプレイで確認できるようなプレビューの機能（箱と行のページ上での位置関係が分かる程度）を本日本語フォーマッタでは備えた．

6. 研究室における実用とその効果

高品質な文書出力環境は，ソフトウェア作成者にプログラムやドキュメントを良くしようとする気を起こさせ，我々のモラルと士気を高めることになった．特に日本語によるコメントと高印字品質により，読みやすくなった．プログラムは完成したものから，浄書マシンで印刷し製本している（**図7**参照）．このようなソフトウェア開発環境の実現も一つの成果である[8]．

研究室で作成される学会の予稿や修士論文など，日本語ワードプロセッサやパーソナルコンピュータで作成した文書を，本日本語フォーマッタで出力できるようになった．また，英文フォーマッタでは**図8**のような出力を得ることができる[9]．

```
ファイル名：    a:demo.lst                          1989年  4月  8日  19時  51分

/*************************************************************/
/*              ファイル名:demo.c                            */
/*              作成者     :大島                             */
/*              作成日     :1987/04/30  14:00                */
/*(モジュールの仕様) : 文書ファイル中の文字をクラス          */
/*                   ごとに分類するプログラム :              */
/*************************************************************/
/***    ヘッダファイル                  ***/
#include        "type.h"
#include        "stdio.h"
#include        "ismacro.h"

/***    外部参照関数の宣言              ***/
FDCL_GLBL       INT             printf();
FDCL_GLBL       FILE            *fopenc();
FDCL_GLBL       INT             fclose();

/***    外部定義関数の宣言              ***/
FDCL_GLBL       INT             main();

/***    静的関数の宣言                  ***/
FLOCAL          AVOID           分類処理関数();
FLOCAL          AVOID           表示処理関数();

/***    外部静的データの定義            ***/
DLOCAL          FILE            *fp;
DLOCAL          INT             漢字文字数;
DLOCAL          INT             片仮名文字数;
DLOCAL          INT             平仮名文字数;
DLOCAL          INT             英字文字数;
DLOCAL          INT             数字文字数;
DLOCAL          INT             ギリシャ文字数;
DLOCAL          INT             その他;

/***    外部関数の定義          ***/
/*-----------------------------------------------------------*/
/*(関数名) : main :                                          */
/*(引数  ) INT     argc;    : コマンド行の引数の数 :        */
/*         CHAR    *argv[]; : 引数を含む文字列の配列を      */
/*                            指すポインタ :                 */
/*(機能  ) : プログラムの制御を行う. :                       */
/*-----------------------------------------------------------*/
ENTRY   main(argc, argv)
INT     argc;
CHAR    *argv[];
{
        if (argc != 2) {
                puts("使用方法 : [demo] [入力ファイル名]");
                exit(0);
```

図7 プログラムの出力例
Fig. 7 An example of program listing.

7. むすび

フォーマッティングマシンのアーキテクチャをとることにより，浄書マシンにおいて，以下の利点が得られた．

(1) 清書用プリントサーバとしてパーソナルコンピュータやワードプロセッサなどで作成した文書をフロッピディスクを介して清書印刷することができる．

(2) 高解像度プリンタの高品質な文書出力を提供するうえで，パーソナルコンピュータや日本語ワードプロセッサなどのホスト側に負担を掛けない．

(3) 日本語フォーマッタ，英文フォーマッタ，リスティングツールなどのソフトウェアにより高い付加価値を付与できた．また，必要な出力ツールが浄書マシン上でプログラミングできるように，印刷に必要な機能をライブラリ化した．このため，柔軟性の高い利用形態が可能となった．

日本語フォーマッタで実現した特徴的な機能を次にまとめる.

(1) 章や章題, 節, 脚書といった文書の論理構造に対して書式設定が行える書式化機能.

(2) 図形等の固定した領域をページ上に確保する箱の指定機能と箱の自動レイアウト機能.

(3) 和欧混合組版規則に基づく, 行頭, 行末, 分離禁則処理およびスペーシング処理機能.

浄書の研究グループは日本語オペレーティングシステムの研究グループと密接な関係を保ちつつ, 日本語フォーマッタの改良や拡張のほか, 図形作成言語, 参考文献データベースの開発を進めている. これらのほかに表作成言語や会話的な環境を実現するための浄書用エディタの開発などは今後の課題である.

6. Exponential and logarithmic functions

Two types of programs have been implemented for exp(x) and log(x) by Dr.Hamada and STL[6].

The STL program is based on the following principle:

$$e^x = \prod_k (1+q_k 2^{-2}) \quad (6.1)$$

where

$$x = \sum_k q_k \log(1+2^{-k}), \quad q_k = 0 \text{ or } 1 \quad (6.2)$$

and

$$\log x = \log 2 - \sum_k r_k \log(1+2^{-k}) \quad (6.3)$$

where

$$x = \frac{2}{\prod_k (1+r_k 2^{-k})}, r_k = 0 \text{ or } 1 \quad (6.4)$$

図 8 英文の出力例
Fig. 8 An example of English text formatting.

謝辞 1987年夏のプログラミングシンポジウム「ヒューマンフレンドリなシステム」において, システム構成の問題点, 高品質出力がもたらす効用などについて, 貴重な御意見を頂いた木村泉教授, 大岩元教授に深謝する. 日立製作所多賀工場の佐々木道甫氏, 榎本順一氏, 堀内雄一氏には, ハードウェア作成において御協力頂いた. ここに感謝の意を表す.

参 考 文 献

1) Kernighan, B. W. et al.: UNIX Time-Sharing System: Document Preparation, *Bell Syst. Tech. J.*, Vol. 57, No. 6, pp. 2115-2135 (1978).
2) Knuth, D. E.: *The TEXbook*, Addison-Wesley (1983).
3) Reid, B. K.: A High-Level Approach to Computer Document Formatting, *7th ACM Symposium of Principles of Programming Languges*, pp. 24-31 (1980).
4) 長谷部紀元ほか: 和欧混合文書組み版システムの試作と組み版規則の検討, 情報処理学会日本語文書処理研究会資料, 21-4 (1985).
5) Takahashi, N. et al.: 浄書: Japanese Output Server with HOspitality (JOSHO), *ICTP '83*, pp. 29-34 (1983).
6) 高橋延匡ほか: OS/o のアーキテクチャと第一版の実現, 情報処理学会オペレーティングシステム研究会資料, 24-11 (1984).
7) 写研・写植ルール委員会: 組み NOW, (株)写研 (1975).
8) 並木美太郎ほか: 日本語ワードプロセッサのソフトウェア生産への応用, 情報処理学会ソフトウェア工学研究会資料, 47-1 (1986).
9) Oxford University Press: *Hart's Rule for Composition and Readers*, 182 p., The University Press Oxford, Oxford (1983).
10) 里山元章ほか: 日本語文書出力システム「浄書」の基本設計と開発システムの実現, 情報処理学会ヒューマンフレンドリーなシステムシンポジウム報告書, pp. 181-191 (1986).
11) 関口 治ほか: 和欧混合組版機能を持つインテリジェントプリンタ第1版の実現, 情報処理学会日本語文書処理研究会資料, 9-1 (1986).
12) 石田晴久: PostScript インタプリタの開発とその日本語化の問題点, 情報処理学会マイクロコンピュータの現状と将来シンポジウム論文集, pp. 95-104 (1987).

(昭和62年11月10日受付)
(平成元年7月18日採録)

里山 元章 (正会員)

昭和37年生. 昭和62年東京農工大学大学院修士課程 (数理情報工学) 修了. 同年(株)日立製作所に入社. 現在, 同社システム開発研究所においてオフィスオートメーション, マンマシン・インタフェースなどの研究に従事. 電子情報通信学会会員.

中川　正樹（正会員）

昭和52年東京大学理学部物理卒業．昭和54年同大学院修士課程修了．同大学院在学中英国 Essex 大学留学（M. Sc. in Computer Studies）．昭和54年東京農工大学工学部数理情報助手．平成1年同助教授．同年4月より電子情報助教授．オンライン手書き日本語入力，日本語計算機システム，文書処理の研究に従事．理学博士．

高橋　延匡（正会員）

昭和32年早稲田大学第一理工学部数学卒業．同年4月(株)日立製作所中央研究所入社，HITAC5020 モニタ，TSS の開発などに従事．昭和52年東京農工大学工学部数理情報教授．平成1年同大工学部電子情報教授．オペレーティング・システム（OS/omicron），日本語情報処理（JOLIS，JOSHO）などの研究，教育に従事．理学博士．ACM，電子情報通信学会，ソフトウェア科学会，計量国語学会各会員．