

DTP システムの設計とそのフォント管理システムの実現

笹川重和, 門奈敦, 早川栄一, 並木美太郎, 高橋延匡

(東京農工大学 工学研究科 情報工学講座)

高解像度LBPの普及やイーサネットなどのネットワーク技術の発達により, ネットワークを利用した卓上電子出版(DTP)システムは身近なものとなった。しかし, 現在のシステムでは, ネットワークを単なる文書転送の手段として使用しているものが多く, 文書の入力段階から考慮した文書作成環境としてのDTP環境を考察し直す必要がある。

本報告では, 日本語文書作成環境としてのDTPシステムである「浄書システム第3版」の設計とそのシステムにおけるフォント管理システムの実現について述べる。このフォント管理システムでは, アルファベットにはない日本語特有の問題点を考察し, 日本語での組版を考慮したシステムとした。

Design of a DTP System and the Implementation of its Font Management System

Shigekazu Sasagawa, Atsushi Monna, Eiichi Hayakawa,
Mitarou Namiki and Nobumasa Takahashi

Department of Computer Science,
Tokyo University of Agriculture and Technology,
Koganei-shi, Tokyo, 184 Japan

With the spread of Hi-Resolution Laser Beam Printers, and the advances in Ethernet technology, Desktop Publishing systems relying on networks have become commonplace. However, the majority of present systems use networks only as a method of transferring documents, thus there is a need to reconsider DTP environments as documentation environments, beginning with the document input process. In this paper, we describe the design of a DTP system "JOSHO System Ver.3" a Japanese documentation environment, and the implementation of its font management system. In implementing this font management system we considered the problems of publishing that only occur in Japanese documents.

1. はじめに

レーザービームプリンタ（以下、LBP）の普及や計算機のコストダウンにより、ワークステーションなどで編集した文書を印刷会社や出版社を通さずに、直接LBPを利用して高品質の文書を出力しようとする卓上電子出版（以下、DTP）は身近な物となった。

また、近年、イーサネットに代表されるようなネットワーク技術の発達によって、Local Area Network（以下、LAN）の構築が手軽に行えるようになった。このようなLANは、様々な目的で利用されているが、高解像度LBPをネットワーク内で共有するプリントサーバとし、ワークステーションなどで編集した文書やプログラムリストをプリンタに転送し、出力するといった方法は、盛んに利用されている。

これらのシステムでは、基本的にネットワークを、文書転送の手段としてのみ利用しているが、ネットワークをより有効に活用し、文書の入力段階から文書の出力を考慮した文書作成環境としてのDTP環境を考察する必要がある。

これに対して、我々の研究室では、日本語文書出力環境の研究として「浄書」（JOSHO: Japanese Output Server with HOspitality）[1]と呼ぶ研究プロジェクトを行ってきている。その目的とするところは、インテリジェントプリントサーバシステムの開発である。

本論文では、日本語文書の出力を目的としたDTPシステム「浄書システム第3版」の設計とそのシステムにおけるネットワークを利用したフォント管理システムの実現について述べる。

2. 浄書システムについて

我々は、「浄書」の目的である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に対して、「浄書システム」と呼ぶプリントサーバを研究・開発してきた。浄書システムは、1983年10月にその基本構成を示し[1]、LBP-10を用いた初版システム、SL-1000（240DPI）を用いた第2版システムを実現してきた。浄書システムの共通な特徴は、次の通りである。

(1) 高解像度LBPおよびLBPコントローラマシンを備える

高品質の文書出力を行うために、高解像度LBPを採用する。また、LBPを制御するためのホストマシンを独自に持つ。

(2) フレームメモリの採用

文章、数式、図、イメージ画像などの様々な出力対象をより柔軟に扱うために、紙面イメージを格納するためのフレームメモリを採用する。また、出力の際にはこの紙面イメージをビデオインタフェースによりLBPに転送する。

(3) フォーマッタプログラムを備える

浄書システムのもつ機能を、種々のコンピュータから利用できるように、プリンタ側に文書を整形出力するフォーマッタプログラムを持つ。このフォーマッタとして、和欧混合組版処理を行う日本語文書フォーマッタ[2]やプログラムリスト出力フォーマッタなどを実装している。

3. 浄書システム第3版の構築目標と現状の問題点

浄書研究の開始当時、LBPは高価であったため、プリンタおよびソフトウェア資源の共有を中心目的として研究開発を進めてきた。また、当時のワークステーションの演算能力は低く、高価格であったことから、パーソナルなWYSIWYG環境の構築を目標とするよりは、システムのアーキテクチャを規定するためのバッチ型フォーマッタマシンの設計を中心研究課題としてきた。

しかし、前述したように、近年、LBPやワークステーションの高性能化、ネットワークの普及が進んできた。このことから、浄書システム第3版では、ハードウェア資源のコストだけでなく、いかに文書作成のコストを低減させるかをシステム構築の中心課題とする。特に単なる出力系としてのDTPシステムではなく、文書推敲支援、手書きインタフェースによるユーザインタフェースの向上、分散環境による文書作成の分業化、文書のデータベース化による文書の再利用などを目標とする。

以後、「浄書システム」とはネットワーク環境を含めたシステム全体のことを指し、プリンタとそのホストマシンを指すときには、「浄書マシン」と呼ぶことにする。

具体的に、我々は、浄書システムにおいて、次のような文書作成環境を提供したいと考える。

(1) 日本語ツールによる文書推敲支援

日本語ワードプロセッサやDTPの普及により、文書作成において、従来、印刷業者の行っていた作業を、文書の発案者が行うようになった。この結果、文章の誤字、脱字のチェックの手間が増加している。

我々の研究室では、文章の誤字、脱字の自動検出手法や文章の推敲支援に関する研究 [3] が行われている。この研究成果を利用し、ワークステーションなどで入力した文書を入力する前に、

- ・表記が統一されているかどうか
- ・誤字、脱字はないか
- ・漢字の含有率は多すぎないか

などの文章の訂正すべき箇所を計算機により検出し、推敲の支援を行う。

(2) 手書きインタフェースによるユーザインタフェースの向上

我々の出力したい文書には、文章ばかりではなく、図や表などが含まれる。これらの図や表を、マウスで描くのではなく、表示一体型液晶タブレットを用いて、手書き入力するシステムの研究が行われている [4, 5]。また、文書編集などにも手書きインタフェースを利用し、ユーザインタフェースを向上する。

(3) ネットワークを利用した分散文書作成環境

印刷会社での組版は、実際には、文章を入力する人、文章を校正する人、図を描く人、全体のレイアウトを行う人など、複数人で行われる。現在のWYSIWYGのシステムではこの作業をすべて一人で行わなければならないが、実際の組版を考えれば分かるように、校正は校正の専門家、レイアウトはレイアウトの専門家が作業した方がよりよい文書を作成できる。また、大学においても、論文などの文書作成の際には、文書作成者以外からの図・表の作成の手伝いや、教官らによる校正・コメント入れなどが行われる。このような作業を、分散環境上で行いたい。

(4) 文書データベース

我々が論文などの文書を書く際には、参考文献などからの引用や以前描いた図の再利用を行う。このような場合に、「～の原稿の～に関する図」といった具合で、文書データベースからその文献を探し

だし、必要とするデータを取得したい。

このような、文書作成環境を実現するためには、システムに最低限、次の機能が必要である。

- (1) 文書のアーキテクチャ（構造と要素）が定義されていること
- (2) 資源管理の機構が実現されていること

このうち、(1) の文書の構造とその要素である文章に関しては、浄書において既に定義している [6]。これは、文書の意味的構造を、“章”、“節”、“図”などの構成要素に分け、図1のような文書構造木として表現し、また、これを図2の例に示すような形式で記述するものである。

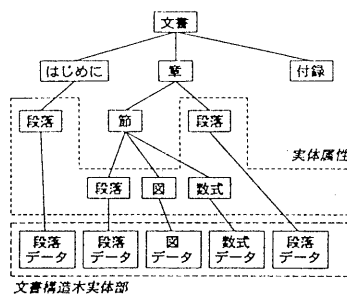


図1. 文書構造木

```
@ノンブル(-@[頁]-@)
@章(2.浄書システムについて@)
@和 我々は、「浄書」の目的である高度な日本語
「浄書システム」と呼ぶプリントサーバを研究・β
LBP-10を用いた初版システム,SL-1000 (240DPI)を
次の通りである。
@和
@図(文書構造木@)(0;23c;19.5c;10.5c@)
@和
```

図2. 浄書ファイルの例

文書要素の図に関しては、図形の構造が図の対象ごとに規定されることから、図3に示すような多階層の図形表現を定義した [7]。

このうち、システムとしては最も基本的な階層に属する文字と図形プリミティブの表現を規定することが必要であると考え。また、それらを実際に分

散環境下で、プリンタやディスプレイに共通に出力するための基本描画ソフトウェアの作成が必要である。

浄書第2版では、プリンタ側の基本描画ソフトウェアとして、Frame Memory Handlerと呼ぶOSの拡張ソフトウェアを実現していた[6]が、浄書第3版では、このような基本描画ソフトウェアを、分散環境下で実現する。

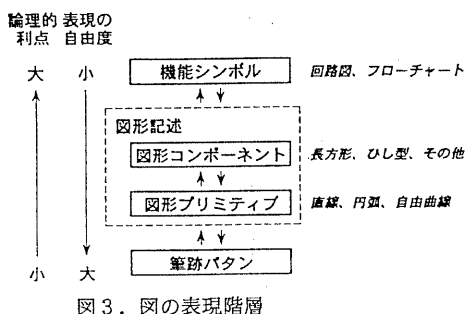


図3. 図の表現階層

4. 基本描画システムの設計方針

3章で示した「浄書システム」の構築目標から、浄書システムで扱う文書の文書要素の最もプリミティブを扱う「基本描画システム」を作成することとした。また、「基本描画システム」の設計方針を次のように定めた。

(1) システム全体の共通機能として基本描画システムを用意する

描画プリミティブを使用した図の入った文書をプリンタで出力する場合や、ワークステーションで文書のレイアウト処理やプレビューなどを行う場合などは、その紙面イメージを作成するために、フレームメモリへの描画を行い、作成した紙面イメージをそれぞれの出力デバイスに転送するといった処理を行う描画ソフトウェアが必要である。

このような描画ソフトウェアは、出力デバイスのそれぞれに特化したものを用意することも考えられるが、その際の描画のインタフェースも異なるのでは、それぞれの描画系に対してプログラムを個別に作成することになる。このため、プリンタ側とワークステーション側で同じ描画インタフェースを持った描画ソフトウェアを持つことが望ましい。

このことから、基本描画システムは、浄書システ

ム全体に共通の機能とし、共通の描画インタフェースを与える。

(2) 出力デバイスに依存しない仮想的なインタフェースを提供する

ドットなどの出力デバイスに依存した単位での描画は、出力デバイスの解像度の変更によって紙面イメージが変わってしまう。また、LBPなどの出力デバイスは年々高解像度化しており、これからも高解像度化が進むことが考えられる。

このため、基本描画システムでは、解像度に依存しない論理的な座標系と、一般的な単位によって位置の指定を行い、文字または図形の描画プリミティブを指定することで描画を行う。また、紙面イメージを格納するフレームメモリを、最終的に出力を行うおうと考えている用紙として仮想化する。

具体的には、ユーザは最終的に出力したい用紙(A4, B4など)をイメージし、用紙の左上から、横10cm、縦15cmの座標から、指定長の線を引く、指定サイズの文字を置くといった具合に命令することで、紙面イメージの作成を行う。

このような仮想化により、ユーザは出力デバイスが何であるかを意識せずに、紙面イメージの作成を行うことができ、プログラムの機種独立性を確保できる。

(3) 和欧混合組版をサポートするための機能を提供する

浄書では、日本の文化に根ざした計算機による文書作成環境を目的として、日本語特有の組版規則を用いた知的組版ソフトウェアの研究・開発を行ってきた。和文組版の例としては次のようなものがある。

- ・縦書き処理
- ・約物(、。、「」など)に関する処理
- ・禁則処理
- ・分離禁止処理
- ・ルビ打ち処理

これらの中には、欧文の組版時にも必要な処理もあるが、それらは和文組版と共通した処理ではなく、基本的には別の処理を必要とする。また、縦書き処理などは欧文ではまったく必要のない処理である。

これらの和文組版を行うためには、文字フォントの特殊な情報(文字の組み合わせによって変わる文字幅の情報など)の取得や縦書き時の特殊な文字フォントの描画方法(、一などの描画)が必要とされ

る。よって、基本描画システムではこれらの和文組版をサポートするための機能を用意する。

(1)、(2)の機能は、PostScript [8]などの既存の描画システムでも実現されている機能であるが、このような和文組版をサポートするための機能はシステムの設計時から考慮されていない。よって、我々はこれらの機能を実現するための研究が必要であると考えられる。

(4) 文字フォントをネットワーク上で共有する

基本描画システムで、任意のサイズの文字の描画を行う場合、出力デバイスの解像度に応じて、適切なドットフォントを作成する必要がある。例えば、400 DPIのプリンタに20級(5 mm角)の文字を出力するには、80ドット角のフォントを作成しなければならない。100 DPIのディスプレイであれば、20ドット角のフォントを作成しなければならない。また、DTPシステムとしては、任意のサイズの文字フォントを出力できる必要がある。このような、様々なサイズの文字フォントを作成するため、浄書システムでは、アウトラインフォントによる文字フォント生成を行う。

しかし、日本語アウトラインフォントのサイズは、1書体あたり、数Mバイトから数10Mバイトであり、複数の書体をシステムに装備する場合フォントの格納サイズが非常に大きくなる。このため、ワークステーション側にフォント格納用の大容量2次記憶装置を装備しなければならないが、効率が悪い。このような問題は、字種が多い日本語特有の問題であり、アルファベットでの出力を目的としたシステムにはない問題である。

この問題を解決するために、浄書システムでは、ネットワーク上にフォントを一括管理するフォントサーバを置き、フォントの共有を行う。このため、フォントの生成・共有などといったフォント管理を行う機能を基本描画システム内で実現する。基本描画システムを利用するユーザには、文字サイズ、文字コード、文字書体などのパラメータから成る仮想的な文字のインタフェースを提供する。このように、基本描画システムで文字のインタフェースを定義し、フォントの実体を基本描画システム内のフォント管理機構で処理することで、浄書システム全体で使用できるフォントに一貫性を持たせることができる。

5. 描画インタフェースの設計

4章の設計方針にしたがい、基本描画システムとそれを利用するユーザとのインタフェースを設計した。

基本描画システムは、ユーザに紙面イメージを作成するための描画コマンドを提供する。描画コマンドは、描画の際のモード設定、文字と図形の描画プリミティブなどから構成され、4章の設計方針にしたがった出力デバイスの解像度に依存しない仮想的なインタフェースを持つものとする。

この描画コマンドは、基本的に次の4つに分類される。また、この描画コマンドの一覧を表1に示す。

(1) 描画モードの設定に関するもの

作成する紙面イメージの用紙サイズ(A4、B4など)、用紙に対する論理座標系の設定(左上原点、左下原点など)、座標指定時の単位系(mm、cm、inchなど)などの紙面をどのように使用するか、紙面へのアクセスをどのようにして行うかといった紙面の仮想化を行うコマンドである。

(2) 文字の描画プリミティブに関するもの

文字フォントをフレームメモリに描画するコマンドである。描画プリミティブの他に、書体、サイズ、回転角度などの文字属性を設定するモード指定コマンドなどからなる。また、和欧混合組版を行うプログラムを作成するためには、文字の幅などのフォントに依存した情報が必要であるが、これらの情報はフォント書体のそれぞれに付随するものである。我々はこの文字に依存する情報は、フォントの実体と共にシステムで管理すべき情報であると考え、これらの情報を得るための命令や文字列をプロポーショナルして描画する命令を、描画コマンドとして実現する。

(3) 図形描画プリミティブに関するもの

直線、方形、曲線などの図形を描画するためのコマンドである。描画プリミティブの他に、線の太さ、実線、破線などのモード指定コマンドなどからなる。これらの図形コマンドに関しては、ユーザの図形描画に自由度を持たせるために、プリミティブとして最低限のものに押さえた。

(4) その他、デバイス制御に関するもの

フレームメモリのクリア、フレームメモリのデバイスへの出力など、デバイス制御に関するコマンド

である。

表 1. 描画コマンドの一覧

機能番号	機能	機能番号	機能
描画モード設定			
1	用紙サイズの指定	1 4	点の描画
2	座標系の指定	1 5	直線の描画
3	単位系の指定	1 6	方形の描画
文字描画プリミティブに関するもの			
4	文字の描画	1 8	円弧の描画
5	文字書体の設定	1 9	点列の線補充
6	文字サイズの指定	2 0	線の太さの指定
7	文字の太さの指定	2 1	線幅の指定
8	文字幅の指定	2 2	領域の塗りつぶし
9	回転角の指定	その他	
1 0	反転などの属性指定	2 3	システムの初期化
1 1	文字幅の取得	2 4	デバイスへの出力
1 2	文字列幅の取得	2 5	フレームメモリのクリア
1 3	文字列の描画		

6. 基本描画システムの設計

4章の設計方針にしたがい、基本描画システムの構成を図4のようにした。基本描画システムが管理する資源には、

- (1) フレームメモリおよび出力デバイス
- (2) フォントデータ

の2つがあるが、それぞれの資源を管理するモジュールとして、

- (1) 描画管理部
- (2) フォント管理部

がある。次に、各構成モジュールの機能について述べる。

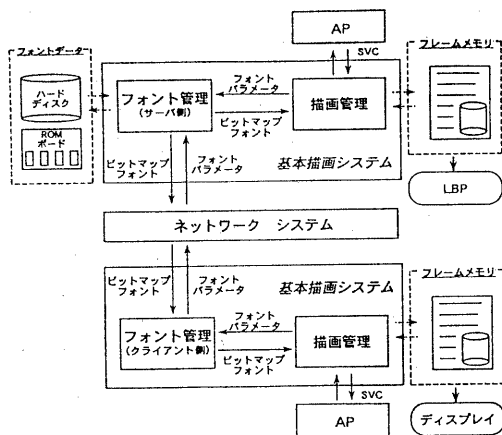


図 4. 基本描画システムの構成

6. 1 描画管理部

描画プリミティブを用いて紙面イメージの作成を行うモジュールである。紙面イメージを格納するフレームメモリ、紙面イメージを出力するデバイスの管理を行う。

和欧混合組版処理を行う日本語フォーマッタ、レビューア、レイアウトなどのユーザプログラムからの要求にしたがい、フレームメモリへの文字描画、図形描画を行う。文字フォントに関しては、フォント管理部からフォントを取得する。また、文字のサイズ、プロポーショナルなどの文字情報のユーザプログラムへの通知処理などを行う。

ユーザからは、5章で述べた描画コマンドとそのパラメータを要求として受取り、描画コマンドの結果を返す。

6. 2 フォント管理部

フォントの共有を行うためのフォント管理を行うモジュールである。フォント管理部では、アウトラインフォントなどのフォントデータを管理する。次に、フォント管理部の機能を、

- (1) フォント格納形態
- (2) 機能分担(サーバ、クライアント)
- (3) インタフェース

の3点に分けて説明する。

(1) フォント格納形態

フォント管理部は、フォントの形態について、実際には、アウトラインフォントだけでなく、ドットフォント、ラスタフォントの形態でも管理する。これは、システムで頻繁に利用されるようなサイズのフォントは、逐次アウトラインフォントから展開するのではなく、ドットやラスタの形態で持つ方が速度的に効率がよいからである。また、フォントデータの格納デバイス(ハードディスク、ROM、光ディスクなど)や格納サイズについても管理する。フォント管理部が実際に管理するフォント格納形態の情報を、表2に示す。

(2) 機能分担(サーバ、クライアント)

4章で述べたように、日本語フォントデータはそのサイズが巨大であるため、ネットワーク上のそれぞれのマシンで持たずに、フォントサーバで一括管理する。このため、フォント管理部は、実際にフォントデータの管理を行うサーバ側とサーバを利用して

フォントデータを取得するクライアント側の2種類がある。浄書マシンのようなサーバマシンにサーバ側のフォント管理部を置き、我々が文書作成などに使用するワークステーションマシンにクライアント側のフォント管理部を置く。

(3) インタフェース

フォント管理部と描画部、サーバ側フォント管理部とクライアント側フォント管理部の間では、取得する文字の級数、コード、書体などの入力データと取得された文字のドットサイズ、文字のベースラインデータ、プロポーショナル時の幅などの出力データでやり取りを行う。また、この入出力パラメータには、4章で述べたように、基本描画システムで和文組版をサポートすることを考慮し、縦書き時のフォント指定が行えるように、また、アルファベットばかりでなく約物などに対しても文字幅などを取得できるようにした。この入出力パラメータを、表3に示す。

表2. フォント格納形態の情報

情報	意味
ロード形態	メモリにロードするか/格納状態のまま使用するか
フォント種類	アウトライン/ラスター/ドットフォントの区別
ドット数	ドット/ラスターフォントの場合のドット数
格納デバイス	格納場所、ディスク/ROM/LAN
格納アドレス	格納デバイス中のアドレス
格納サイズ	格納デバイス中のサイズ

表3. フォント管理部との入出力データ

パラメータ	入出力	意味
書体名	入力	明朝体/ゴシック体などの書体名
文字コード	入力	文字のJISコード
文字サイズ	入力	取得したい文字の級数
反転属性	入力	反転などの属性指定
変形属性	入力	変形組の属性指定
横罫/縦罫属性	入力	横罫/縦罫用のフォント生成指定
解像度	入力	要求側の出力デバイスの解像度
文字ドット数	出力	生成された文字のドット数
オフセット	出力	ベースライン、左オフセットなどの位置
プロポーショナル幅	出力	プロポーショナル時の文字幅
文字ドット実体	出力	ドットフォントデータ実体

7. 実現

現在、浄書システムは、ハードウェアの中心となる高解像度LBPおよびそのホストマシンの世代が第2版から第3版へ移行している。ソフトウェアは第2版のソフトウェアの移植に加え、OSをモジュール化し、拡張性を向上することを目的としたハイパOS「礎」[9]の開発、LBPの高解像度化に対応するためのアウトラインフォントによる文字描画の拡

張などが行われ、毎月8000枚の出力を行い、研究室の文書出力を担っている。

基本描画システムは、我々の研究室で開発しているOS/omicon第3版を使用して実現する。開発言語は、OS/omicon上のC言語CATを使用する。また、本システムは、OSの拡張として、第3版のデバイスドライバの形式で実現され、ユーザから基本描画システムへの要求は、SVC(Super Visor Call)を用いて行われる。

現在、本システムの土台となるネットワークシステムの実現、および、このネットワーク上で動作するフォント管理部の実現は終了しており、現在は、描画管理部の実現を行っている。

フォント管理システムのサーバ側とクライアント間におけるフォントデータの入出力は、6章で述べたように、表3に示すようなパラメータで行われるが、このインタフェースを図5に示すような要求/結果バケットで入出力するように実現した。

また、サーバ側からクライアント側に返されるフォントデータは、基本的にはサーバ側でアウトラインフォントを展開し、その展開されたドットフォントのデータ形式で行われる。しかし、次のような場合は、クライアント側でフォント展開を行った方がよいと考え、サーバ側で展開せずに、格納時のフォントデータの形式でもデータを取得できるようにした。

(1) サーバ側の負荷が大きい場合

サーバに対するフォント取得要求が多い場合や格納時のフォントデータサイズよりも展開後のフォントサイズが大きく上回る場合などは、サーバ側は、結果バケット中のフォントデータ実体として、格納形式のフォントデータを返す。クライアント側では、結果バケットのフォントデータ実体の種別を判断して描画する。

(2) フォントを展開するハードウェアなどがクライアント側に装備された場合

クライアント側にフォント展開用のハードウェアなどが装備され、常にクライアント側で展開した方が速い場合は、クライアント側は、要求バケット中の“文字級数”を“0”にすることで、結果バケットのフォントデータ実体として、格納形式のフォントデータを返す。

現在、実現された部分のプログラム行数は、ネットワークシステムが5000行、フォント管理部が9000行となっている。

要求バケット

データ名	単位	内容
書体番号	2	取得する文字の書体番号
文字コード	2	取得する文字のJISコード
文字総数	2	取得する文字の総数
反転属性	2	取得する文字の反転属性の属性
変形属性	2	取得する文字の変形属性の属性
要求座標系	2	要求側の座標モード
要求単位系	2	要求側の単位系
要求解像度	2	要求側の出力デバイスの解像度

結果バケット

データ名	単位	内容
文字ドット数	2	生成された文字ドット数
ベース位置	2	文字のベースライン位置
オフセット位置	2	文字のオフセット位置
文字高さ	2	ベースラインからの高さ
アローシフト幅	2	プロポーショナル幅
データ実体長さ	2	この後に続くデータ実体の長さ
データ実体	n	文字データ実体(可変長)

図5. フォント入出力バケット

8. おわりに

日本語文書の出力を目的としたDTPシステム「浄書システム第3版」の設計を行い、そのシステムにおけるフォント管理システムを実現した。

今後の課題として、次のことを計画している。

(1) 描画部の実現、および、システム全体の評価

現在、基本描画システムのフォント管理部は実現されているが、描画部は実現途中である。今後、描画部および全体の実現を行い、実際の使用における速度評価などを行う。

(2) フォント管理部とのインタフェースの評価

基本描画システムを使用した日本語組版フォーマッタなどの実現を行い、今回採用したフォント管理部とのインタフェースの検討、評価、改良を行う。

(3) フォントキャッシュの実験

現在、クライアント側フォント管理部は、フォント取得のたびに、サーバを呼び出すが、スループット向上のために、クライアント側で一度取得したフォントのキャッシングを行い、ネットワーク通信の負荷を減らすことを計画している。また、クライアント側では、キャッシュの効果がシステムの描画速

度のカギとなる。このため、キャッシュアルゴリズムの変更による速度向上の実験を行なう。

(4) フォント生成のハードウェアによる高速化

文字フォント生成の速度を向上させるため、ハードウェアによるフォント描画を行う。

参考文献

- [1] Takahashi, N. et al. : 浄書: Japanese Output Server with HOspitality (JOSHO), ICTP' 83, pp. 29-34 (1983).
- [2] 関口他: 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
- [3] 下村他: 最小コストパス探索モデルの形態素解析に基づく日本文誤り検出の一方式, 情報処理学会論文誌, Vol. 33, No. 4, 1992
- [4] 曾谷他: 手書きユーザインタフェース, 情報処理学会第31回プログラムシンポジウム報告集, pp. 125-134, 1992
- [5] 風間他: 文房具メタファを用いた手書き作図インタフェース, 情報処理学会ヒューマンインタフェース研究会, 43-3, 1992
- [6] 里山他: 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, 情報処理学会論文誌, Vol. 30, No. 5, 1989
- [7] 中川他: 手書きインタフェースのための図形階層文法, 情報処理学会第44回全国大会, 7K-6, 1992
- [8] Adobe Systems Inc. : PostScript Language Reference Manual, Addison-Wesley, Reading, Massachusetts, 1985
- [9] 並木他: ビルディング・ブロック・システムのための共有ソフトウェアバス~礎~, 情報処理学会オペレーティングシステム研究会, 51-2, 1991