

# 日本語DTPのための基本描画システムの作成

4 L - 9

笹川重和, 門奈敦, 早川栄一, 下村秀樹, 並木美太郎, 高橋延匡

(東京農工大学)

## 1. はじめに

現在, 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, レーザビームプリンタ(以下, LBP)などの高解像度の出力デバイス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)は身近な物となっている。また, 出力デバイスであるLBPは, 年々高解像度化が進んでいる。

このような出力デバイスの高解像度化に対応するため, また, ディスプレイ上で出力紙面と同じイメージを表示しながら編集したいという要望などから, 多くのDTPシステム[1]では, 自由なサイズの文字フォントを生成できるアウトラインフォントによる文字描画を採用している。

しかし, 日本語での文書出力を考えた場合, 次のような問題点があげられる。

### (1) アウトラインフォントの格納サイズが大きい

日本語はアルファベットに比べて, 字数が多く, また, 筆画数も多いため, 文字フォント1文字あたりのデータサイズが大きい。複数の書体を使用することを考えると, フォントデータのサイズは巨大なものとなり, ワークステーション個別にフォントを持つことはコストがかかる。

### (2) フォントの生成速度に時間がかかる

アウトラインフォントでは, 輪郭線を $n$ 次曲線などで描画してから, 内部を塗りつぶすといった手法のために, フォントの生成に多くの時間を要する。また, 日本語文字は輪郭線が複雑であるためさらに時間を要する。

本報告では, これらの問題点を解決するための, DTPシステムにおける基本描画システムの作成について述べる。

## 2. 浄書システム

我々の研究室では, 日本語文書出力の研究として, 「浄書(Japanese Output Server with HOspitality)」[2]の研究を行なってきた。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トサーバ「浄書システム」を開発してきた。

浄書システム第2版では, プリンタ側の基本描画システムとして, Frame Memory Handler [3]と呼ぶOSの拡張ソフトウェアを実現していたが, アウトラインフォントは使用しておらず, 自由な文字サイズの出力を得ることはできなかった。また, このソフトウェアはプリンタ側のみ実現さ

れていたため, WYSIWYG (What You See Is What You Get) 環境は実現されていなかった。

これらのことから, 我々は, 新たに浄書システム第3版を開発することとし, そのシステムにおいて, 前述の日本語文書出力の問題点を解決するための基本描画システムを作成することとした。

## 3. 基本描画システムの設計方針

基本描画システムの設計方針は次のとおりである。

### (1) 分散環境下で共通の描画システムとする

ネットワーク中の全てのマシンにおいて, 同じ描画体系を持った描画システムをシステムソフトウェアとして用意し, マシンの持つ出力デバイスに応じて描画を行う。また, 描画体系は, 出力デバイスの解像度に依存しないものとする。これにより, WYSIWYG環境を実現する。

### (2) 基本描画システム内においてフォント共有を行う

巨大なサイズとなるアウトラインフォントをネットワークを利用して共有する。共有の方式は, フォントをファイルとして共有するのではなく, 基本描画システムが管理・生成するオブジェクトデータ(アウトラインフォントデータ, ビットマップデータ)として共有する。これは, フォントの形状や形状の持つ意味などを把握している描画システムのレベルで共有を実現した方が速度効率が良いと考えたからである。

このような, フォント共有を実現するために, 描画システムをフォントの一括管理・生成を行なうサーバ側システムとサーバからフォントを取得するクライアント側システムの2つに分ける。

### (3) システムのハードウェアによる高速化を考慮する

フォント生成速度の向上のために, 日本語アウトラインフォントの展開ハードウェアなどを持った別プロセッサ上で基本描画システムの実行を行うことを考慮し, 描画の処理依頼の送付およびその処理結果の受取りといった通信にかかるオーバーヘッドを減少させるため, 複数の描画依頼の送付および複数の処理結果の受取りを一括して行えるようにする。

## 4. 基本描画システムの構成

設計方針にしたがい, 基本描画システムの設計を行った。基本描画システムの構成は図1のとおりである。以下に基本描画システムのインタフェースについて説明する。

4. 1 描画コマンド

基本描画システムは、フレームメモリに紙面イメージを作成するための描画コマンドをユーザに提供する。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の描画命令などから構成される。描画コマンドに付随する位置指定などのパラメータは、解像度に依存しない単位で行なわれる。描画コマンドの一覧を表1に示す。

4. 2 コマンドポート

基本描画システムでは、描画コマンドの依頼およびその結果を受け取るためのコマンドポートと呼ぶ仮想的な通信路を提供する。ユーザは、あらかじめ基本描画システムに対し、コマンドポートをオープンし、コマンドポートを通して描画コマンドを送付する。また、描画コマンドを実行したことによって得られる結果をコマンドポートを介して受け取る。

このコマンドポートには、次の2つの種類がある。

(1) フレームメモリ描画用コマンドポート

描画コマンドを受取り、指定されたオブジェクトデータ(ビットマップフォントなど)を生成する。このオブジェクトデータをフレームメモリに描画する。また、結果として描画コマンドの返値を返す。

(2) 描画オブジェクト取得用コマンドポート

描画コマンドで指定されたオブジェクトデータを生成し、結果として生成したオブジェクトデータと描画コマンドの返値を返す。

(1) のコマンドポートは、通常アプリケーション(以下、AP)から、基本描画システムに対して使用される。(2) のコマンドポートは、ネットワーク中の基本描画システム間でフォント共有を実現するために使用される。つまり、クライアント側描画システムは、APから受け取った描画コマンドの内、自分では処理できない描画コマンドをサーバ側描画システムに送り、オブジェクトデータを生成してもらい、自分のフレームメモリに描画を行う。このように、コマンドポートによる通信モデルをネットワーク間での通信に適用することで、フォントの共有を実現できる。

基本描画システムを別プロセッサ上で動作させる場合は、ネットワークによる通信と同様に、コマンドポートをプロセッサ間の通信を用いて実現する。このような並列システムでは、通信にかかるオーバーヘッドをいかに減少させるかが重要となる。このため、描画コマンドは一度に複数のコマンドを依頼し、結果も複数のコマンドの返値と複数のオブジェクトデータを返すことができる。このため、コマンドポートの通信は、パケット単位で行われる。

5. 基本描画システムの実現

基本描画システムは、我々の研究室で開発しているOS/omicron第3版[4]上で、OSの拡張機能として実現した。コマンドポートの通信は、OSのメッセージ通信機能とネットワーク通信機能を使用して実現した。また、基本描

画システムを別プロセッサ上で動作させる場合には、プロセッサ間通信機能を利用して実現した。

6. おわりに

日本語文書出力における問題点を考慮し、フォント共有およびハードウェアによる高速化を実現したDTPシステムのための基本描画システムを作成した。今後は、サーバ側から取得したオブジェクトデータ(特にフォントデータ)のクライアント側でのキャッシング手法などを検討していく予定である。

参考文献

[1] Arthur Naiman:The Macintosh Bible, second edition, Japanese-language version, Gijyutsu Hyouron Inc., 1990  
 [2] Takahashi, N. et al.: 浄書:Japanese Output Server with HOspitality (JOSHO), ICTP' 83, pp. 29-34 (1983).  
 [3] 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  
 [4] 岡野他: 並列処理用OSカーネル“Omicron V3”とハイバOSによる共有メモリ型マルチプロセッサへの実装, 情報処理学会論文誌, Vol. 32, No. 6, 1990

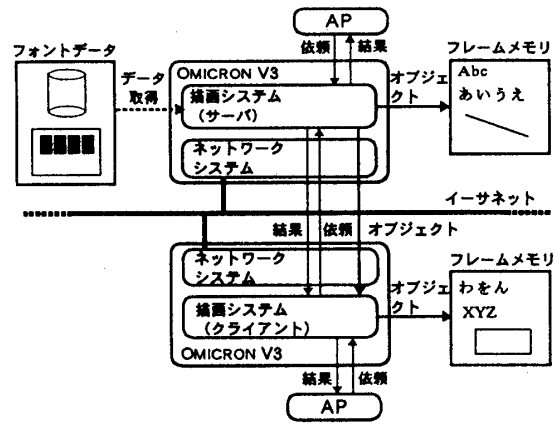


図1.基本描画システムの構成

表1. 描画システム コマンド一覧

機能番号	機能	機能番号	機能
描画モード設定		図形プリミティブに関するもの	
1	用紙サイズの指定	14	点の描画
2	座標系の指定	15	直線の描画
3	単位系の指定	16	方形の描画
文字描画プリミティブに関するもの		17	円の描画
4	文字の描画	18	円弧の描画
5	文字書体の設定	19	点列の線補完
6	文字サイズの指定	20	線の太さの指定
7	文字の太さの指定	21	線種の指定
8	文字幅の指定	22	領域の塗りつぶし
9	回転角の指定	その他	
10	反転などの属性指定	23	システムの初期化
11	文字幅の取得	24	デバイスへの出力
12	文字列幅の取得	25	フレームメモリのクリア
13	文字列の描画		