

自社製プリンタ言語インタープリタ Emperor の開発

Development of the Emperor PDL Interpreter

藤原 葉子*
Fujiwara, Yoko

Mabry Dozier **
Dozier, Mabry

秋山 文人*
Akiyama, Fumihito

能西 豊茂*
Nohnishi, Toyoshige

要旨

オフィスMFP（Multi Function Peripherals：多機能複合機）におけるプリント機能の重要性は年々増加しており、より高速・高画質・高機能のプリント機能を安価に顧客に提供することが要望されている。

それらの要求に応えるため、自社のプリント機能インタープリタ“Emperor”を速度・画質面で改善し、更にMFPシステムとの最適化、漢字フォント搭載などを実施した。

また、機種カスタマイズ部とコア部にソースコードを分離し、コア部を各機種共通とした。これにより、プリンタからMFPまでの幅広い製品群への対応が可能となった。

今後はEmperorコア部をエンハンスして、新規プリンタ言語の追加などを行う。

本稿では、これらの技術内容を紹介する。

Abstract

Today, the printing function in office MFPs (multi-function peripherals) is more important than ever, and demand is strong for higher speed, better image quality, greater variety of printing functions, and reasonable cost. To meet these demands, Konica Minolta's Emperor PDL (printer description language) interpreter has been significantly improved. The Emperor's implementation is well-optimized on MFP platforms by the interface established with MFP systems, and a kanji font is available for both PS and PCL. By separating the product-customization module from its common core code, the Emperor has been applied to both printers and MFPs. In the future, the Emperor's common core code will be enhanced to support new printer description languages. This paper introduces technical advances behind the Emperor's development.

1 はじめに

コピー・プリント・スキャン及びファックスの機能を一体化したMFP（Multi Function Peripherals：多機能複合機）において、プリント機能の重要性は年々高まってきた。プリント機能の基本は、プリンタ言語という形式でPC等から入力される描画データをインタープリタと呼ばれるモジュールでビットマップに変換して印刷することである。そこで使用されるプリンタ言語は、Adobe社のPostScript（以下、PSという）、HP社のPCLが世界標準となっている。

これらPS・PCLでは、複雑なグラフィックや文字の描画をプリンタ言語で記述し、インタープリタで言語を解釈して描画するため、処理の負荷が増大する。従って、高画質ではあるが、低価格で高パフォーマンスを実現することが難しかった。

そこで、オフィス市場向けのカラープリンターでは、コスト面から途中でビット数を削減して処理の負荷を軽減することが行われており、画質を十分に確保できないという課題があった。

また、画質を重視するPSでは、ビット数を維持して完全多値処理を行う製品があるが、専用の資源（メモリ、CPU）を使用しており、価格的には非常に高いものとなっていた。

このように、オフィス市場でのプリント機能に関して、機能・性能・価格の全ての面で十分にユーザーニーズに応えきれていない状況であった。

本開発では、コニカミノルタの自社製インタープリタ“Emperor”を活用し、これらの課題を解決してオフィス市場のニーズに合ったプリント機能を提供することを目的とした。具体的には、最新のPostScript 3エミュレーション、PCL 6エミュレーションのインタープリタをMFPに標準搭載し、高画質かつ高速のインタープリタを幅広い製品群に共通展開して開発効率を向上することを目標とした。

本稿では、この目標を達成するための技術内容を

- 1) 高画質化
- 2) 高速化
- 3) 製品群への共通展開

という3つの点から説明する。

* コニカミノルタビジネステクノロジー株式会社
制御開発本部 制御第2開発センター 第21開発部
** Konica Minolta Systems Laboratory Inc. Core Technology Division

2 Emperorの概要

2.1 MFPシステムのソフトウェア構成

Fig. 1にMFPシステムのソフトウェアのブロック図を示す。本システムは、入出力制御を行うネットワーク制御部、ビデオ入出力を行うエンジン制御部、システム制御部から構成される。

システム制御部では、コピー、ファックス、スキャン、プリントの各機能別にジョブ制御が行われており、インタープリタは、プリントプロセス部から制御されるアプリケーションのとして動作する。

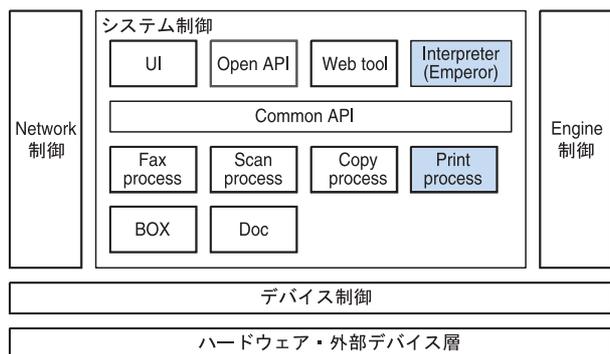


Fig.1 MFP system software block diagram

2.2 Emperorの構成

Fig. 2にEmperorのソフトウェアブロック図を示す。大きくは、言語解析部とスタライザ部に分かれている。

言語解析部では、入力データであるプリンタ言語を解析してDisplayListと呼ばれる中間データに変換する。

ラスタライズ部では、中間データを読み込み、Frame空間に描画処理を行う。

プリンタ言語はPS, PDF, PCL 5, PCL 6と、画像フォーマットであるTIFF, JPEGのダイレクト印刷をサポートし、言語解析部にはそれぞれの解析を行うブロックがある。

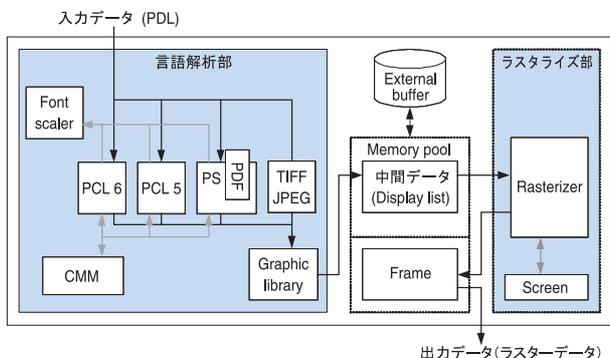


Fig.2 Block diagram of Emperor

3 開発技術

3.1 高画質化

Emperorでは、以下に紹介する2つの技術を搭載することにより、オフィス向けインタープリタとしての高画質を実現した。

3.1.1 多値処理化

オフィス向けプリンタでは、内部処理を簡略化するために言語指定の画像品質に対し、制御ビット数を制限して処理する構造が一般的である。そのため、解像度や階調性の情報が完全に保持できず、PSの高画質な画像処理やPDFで要求されるTransparency機能への対応が困難であった。Emperorではこれらの問題に対処するため、中間データの構造、ラスタライザの構造を改良し、完全多値処理とした。

Fig. 3に言語データ入力からエンジン出力までのデータフローを示す。プリンタ言語の形式で入力されたデータは、言語解析され、中間データに変換される。さらにラスタライズ処理され、Frame空間に描画処理された後、スクリーン処理を経てプリントエンジンに出力される。Emperorでは、すべてのFrame空間に各色8bitを維持して処理することにより、階調性と解像度を落とすことなく処理可能とした。

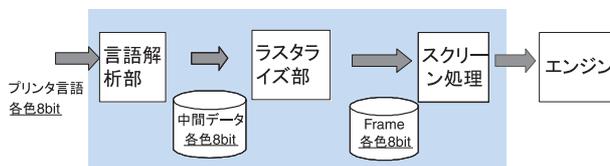


Fig.3 Emperor data flow

3.1.2 オブジェクト分離

Emperorでは、DisplayList (中間データ) に保持した描画オブジェクトを文字、グラフィック、イメージの3種類に分離する。Fig. 4にオブジェクト分離の処理フロー (PCLの場合) を示す。

プリンタ言語の入力データから解析したオブジェクトの種類を検出し、画素ごとのオブジェクト種類の情報としてDisplayListを出力する。その情報に基づき、色変換やスクリーン処理の切り替えが行われる。具体的には、以下の通り。

- ①色変換処理では、文字・グラフィックは鮮やかな色再現、イメージは忠実な色再現となるよう色変換モードが切り替え。
- ②スクリーン処理では、文字・グラフィックは高解像度スクリーン、イメージは高階調スクリーンを使用するよう切り替え。

Fig. 5にサンプル画像を用いた改善結果を示す。比較の

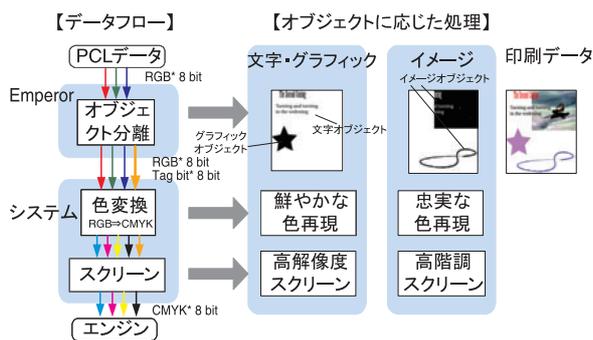


Fig.4 Object type segmentation processing in color MFP

ために、色変換、スクリーン処理をそれぞれ2種類のモードで処理した結果を示す。○で囲んだ部分在实际に選択した処理である。

オブジェクト分離を行わない場合、写真画像で背景部分の階調がつぶれる場合や、階調再現を重視して高階調スクリーンを使用すると、文字のエッジのギザギザ(ジャギー)が目立つ等の問題があった。

これらの課題に対し、同一ページ内に異なる種類のオブジェクトがあった場合でも、オブジェクト分離結果に応じて画像処理を最適な制御を選択し切り替えることにより、文字・グラフィックとイメージの最適化を行い、各々の高画質処理を両立させることが可能となった。

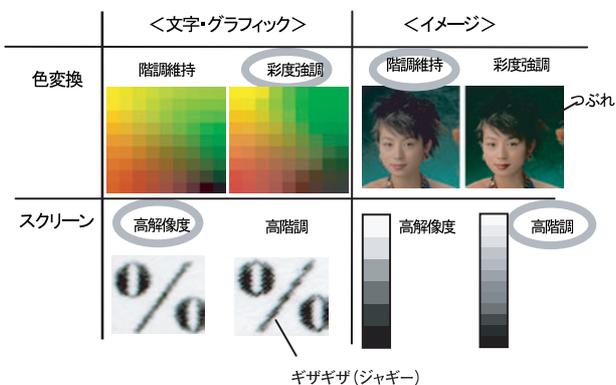


Fig.5 Images acquired by the test of object type segmentation

3.2 高速化

高画質化の目的で多値処理を導入したことに起因し、ソフト処理の負荷が増大している。これらに対し、様々なアルゴリズム改善を積み重ね、高速化を図っている。ここでは、その中の2つを取り上げる。

3.2.1 カラー/モノクロバンドの判別

従来の処理では、入力の色空間がRGBのモノクロページの場合に、R = G = B 3面のFrame空間に描画処理され処理時間が遅いという問題があった。

Fig.6に改善前後の処理を示す。生成した各ページのFrame空間は、バンドと呼ばれる単位で処理される。

改善前(PCLの場合)では、ページ全体をカラーで処理するため、モノクロのバンドもカラーのバンドもRGB計3回ラスタライズの描画処理が行われた。

それに対し改善後では、言語解析処理でバンド毎にカラーかモノクロかを事前に判別し、中間データに判別結果を保存することとした。ラスタライズ処理時には、中間データに保存されている判別結果からモノクロか否かを判別し、モノクロの場合はFrame空間にK成分のみをラスタライズし、1回だけの処理とすることで高速化を行った

また、描画データの存在しないFrame空間に対しては、ブランクバンドを定義し、処理を削減して高速化を図った。

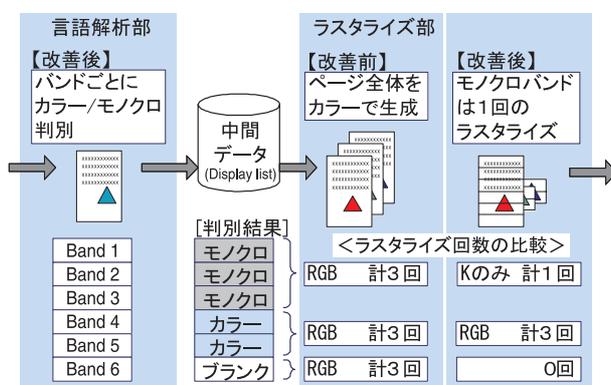


Fig.6 Color/monochrome band detection processing

3.2.2 イメージ処理フローの最適化

従来の処理では、入力したイメージ画像はFrame描画する出力解像度に変倍された後に中間データに保存されていた。従って、イメージ拡大を指定するような描画データの場合は、中間データのサイズが大きくなるという課題があった。

改善後は、イメージ拡大が必要な画像データに対し、元となるイメージ情報のまま、解像度を変倍せずに中間データに保存する。この後、ラスタライズ処理時に拡大・縮小の処理を行った。イメージ縮小が必要な画像

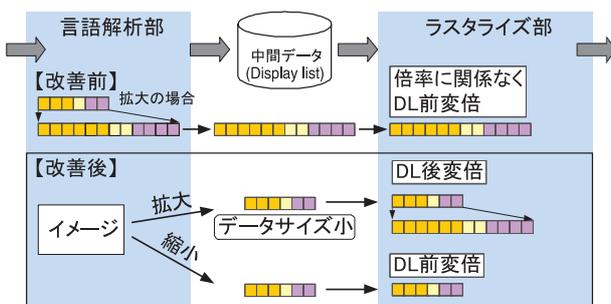


Fig.7 Image rendering process according to magnification ratio

データに対し、縮小して中間言語に保存するようにした。即ち、倍率に応じて変倍処理のフローを最適に切り替えることにより、中間データのサイズを削減し、パフォーマンスを改善することができる。

Fig. 7に改善前と改善後のイメージの処理フローを示す。

3. 2. 3 パフォーマンス比較

これまでに紹介したアルゴリズム改善の他、様々な改善の結果、以下の高いパフォーマンスを達成した。Fig. 8に同クラス他社のカラーMFPのパフォーマンス比較を示す。

Fig. 8では、各製品のPPMを100%とした換算値でパフォーマンスを示した。Emperor搭載機種bizhubC450では、完全多値処理を導入しながら、標準的な印刷データをエンジンスピードで処理している。他社同クラスの製品と比較しても良いパフォーマンス結果を得た。

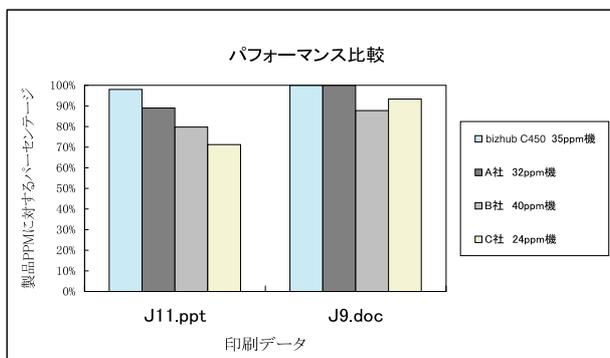


Fig.8 Performance comparison with other MFP products

3. 3 製品群への共通展開

インタープリタを短期間に製品展開するためには、同一の安定したソースコードを共通的に適用することが重要となる。本項では、以下の4つのポイントを説明する。

- 1) MFPシステムとの共通インターフェイス策定
- 2) MFPシステムハードウェアへの最適化
- 3) 海外・日本向けの共通ソース適用(日本版:漢字付)
- 4) 機種カスタマイズ部の分離

同一ソースコードによりプリンター製品からMFP製品までの幅広い製品群への共通展開を実現した。

3. 3. 1 MFPシステムとの共通インターフェイス策定

MFPでは、インタープリタはコピー・スキャン・ファックスなどの機能を制御するMFPシステム上で動作する。

Fig. 9にカラーMFPにおけるプリント時のインタープリタとMFPシステムとの画像データ送信のシーケンスの一部を示す。このようなシステムとのシーケンス制御に対応し、機種間の共通性を確保する必要がある。EmperorとMFPシステム間のIFを機種間共通に定義し、システムへの移植性を高めた。

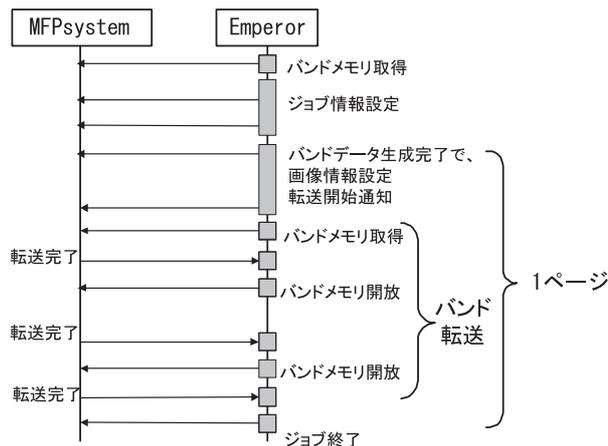


Fig.9 Example of interface between MFP system and Emperor interpreter MFP system

3. 3. 2 ハードウェア最適化

カラーMFPでは、膨大な画像データ(例えば、A3サイズ、600dpi、CMYKのFrame空間データ量は約270MB)を扱うため、画像圧縮が必須であり圧縮ハードウェアを搭載している。Emperorはこのハードウェアを効率的に活用し、高速かつ高メモリ効率な描画処理を実行している。

Fig.10にEmperorの作成したFrame空間(バンド)をMFPシステムへ転送する処理フローを示す。この処理では、1ページのFrame空間をバンドに分割して作成する仕組みになっており、図の1サイクルめでは、バンド1のFrame空間を生成し転送用バッファに書き込む。2サイクルめでは、1サイクルめで転送されたバンド1をハードウェアを活用し圧縮する処理と、バンド2を生成して転送する処理が並行処理される。同様に、2サイクルめで転送されたバンド2を圧縮する処理と、バンド3を生成してシステムへ転送する処理が並行処理される。

このように、ソフトウェア(Emperor)によるラスタライズ処理と、ハードウェア(MFPシステム)による圧縮処理を並行処理することにより高速化を図った。更に、生成したバンドを、順次圧縮してメモリの効率化も図っている。

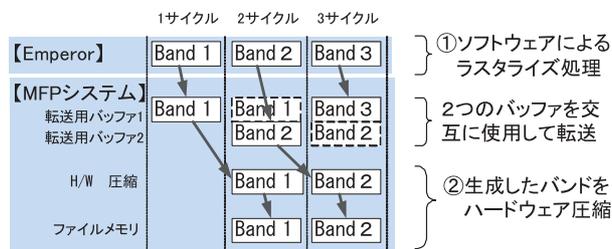


Fig.10 Parallel processing of S/W rendering and H/W compression

1つの例として、色変換処理のフローの最適化をFig.11に示す。

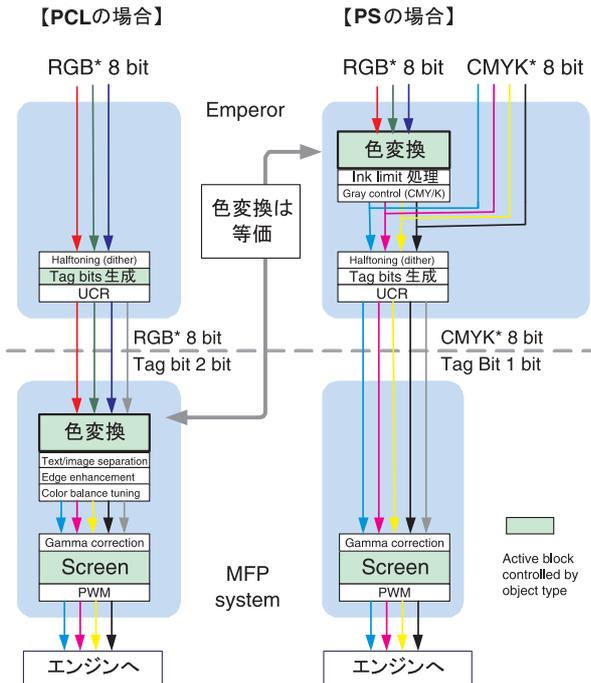


Fig.11 Implementation of color correction in PCL and PS

カラーMFPシステムでは、コピー画像処理の高速化を図るため、RGB→CMYK色変換のハードウェアを搭載している。

プリント機能でも色変換ハードウェアを共通使用することにより、コストを抑えて高速化を狙った。しかし、電子写真式プリントエンジン出力は一般にCMYKであり、ラスターライズ部の描画後にはCMYKのFrameを生成する。そこで、本EmperorではRGB出力の処理を新たに追加した。

上記処理はPCLの場合に活用する。

しかし、PSの場合には、CMYK直接描画処理や、複数の色空間指定（LAB, XYZなど）が含まれるため、K成分の保持が必須要件となる。よって、この場合は、CMYK描画処理となる。

この方法により、MFPシステムのもつハードウェアを共通使用しながら、PCLの高速化とPSの高画質を両立させることが可能となった。

3. 3. 3 海外・日本向けの共通ソース適用

日本市場では、高速化のために漢字内蔵フォントが必要とされている。Emperorでは、Fig.12に示すような形態で漢字フォントを搭載しているが、フォント以外のソースコードは共通化処理とした。

自社開発であることを生かし、以下のような特長がある。

- ・PS, PCLで字母データを共通とすることにより、メモリを削減（外部フォント導入では、共通字母データの実現は困難）

標準的なPS・PCL言語で対応し、海外・国内向けのソースコードを共通化（日本語フォント搭載は、各社独自言語で対応している製品が多い）
 このように、漢字内蔵フォントを搭載することにより日本語対応し、それ以外は共通ソースコードを使用した。

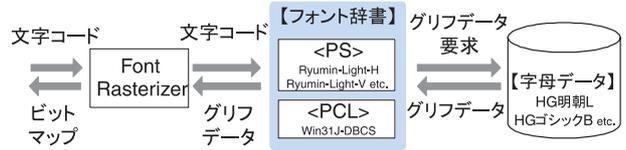


Fig.12 Implementation of resident kanji font

3. 3. 4 製品カスタマイズ部の分離

Emperor搭載対象機種仕様の仕様は、カラー8bit, モノクロ1bit, 出力解像度600dpi, FAX機能で200/300/400dpiなどと、幅広いものとなっている。

また、製品ごとにPJLのサポート、デモページのデザインなども異なる。

これらの機種対応にカスタマイズすることを主眼とし、Fig.13に示すとおり製品カスタマイズモジュールを定義した。このモジュールを分離することにより、機種対応が容易となる。また、このカスタマイズ部を分離するために、共通Core部との間はAPI関数を定義し、Core部の独立性を高めた。

これにより、カスタマイズ部のモジュール修正のみで、コア部のソースコードの変更なしで機種対応を可能にした。

これにより、Magicolor5450, Magicolor7450などのプリンタ製品からbizhubC450, bizhub750などのカラー及びモノクロのMFPまで幅広い製品群への搭載が可能となった。

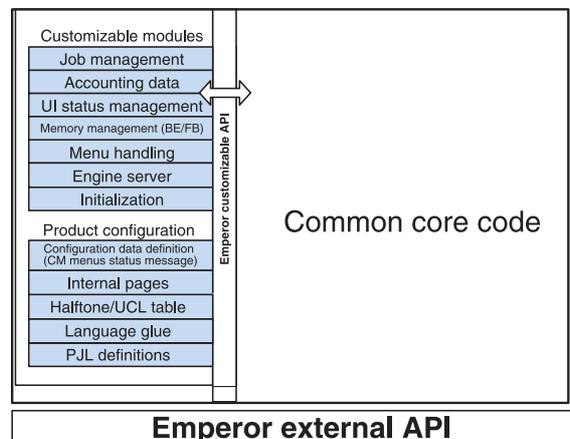


Fig.13 Emperor customization block

4 まとめ

以上、自社開発である利点を生かし、高画質かつ高速のインタープリタ“Emperor”を開発し、コアのソースコードの共通化を図って開発効率を向上することにより、プリンターからMFPまでの幅広い製品群への展開を実現した。

今後は、XPSなどの新規プリンタ言語への対応、プロダクション領域で要求される機能開発に取り組む所存である。

5 謝辞

本開発にあたってご協力いただいた、制御開発本部、第11開発部、第12開発部、第13開発部、第23開発部、ソリューション開発部の関係者の皆様に感謝いたします。

- ・PCLは、Hewlett-Packard Companyの登録商標です。
- ・Adobe, PostScript 3, PDFは、Adobe Systems Incorporatedの登録商標です。
- ・XPSは、Microsoft Corporationの登録商標です。