5P－6

浄書参の設計と実現

狩野敦，鈴木未来子，曽谷俊男，並木美太郎，高橋延匡
（東京農工大学　工学部　情報工学講座）

1．　はじめに

1982年に、開始した浄書（Japanese Output Server with HOspitality）プロジェクト研究は、ソフトウェアとして日本語フォーマッタを始め、リスティングツールや議事録出力ツールを備えるにいたった。それらは毎日、研究室の文書作成環境として利用されている。

しかし、浄書計画の当初の目的は研究室、しいては学科内で要求される文書を出力することであった。つまりマニュアル、プログラムリスト、実験用解説図、会議議事録、仕様書等の出力である。そこには、日本語文章、英語文章、図、絵、数式、イメージ、グラフ、表等種々の形態が含まれる。

また、仕様書、マニュアル等の不備がソフトウェア工学的に問題となってきているが、日本語文章とリスト、図等が同時に同一紙面上に表現できなくては、見やすく分かりやすいドキュメントは作成できない。

そこで我々は、これら多岐にわたる文書を総合的に管理し、統一的に出力できる環境の実現を目指している。

2．　浄書第2版の問題点

1987年に実現した日本語フォーマッタは和欧混合組版処理を実現しており、論文等の出力に数多く利用されてきた [1]。

我々はその日本語フォーマッタを実際に利用し、機能拡張を行ってきた。その過程で我々は以下のような問題があることに気づいた。

1）要求される文書形態（図、表、化学式等）は多様であるが、展開処理を行う部分には限りがある。

2）出力の多様性（文書形態、レイアウト）を増すためには、過去から存在するプログラムについて、すべてを把握しなければ拡張できない。

これらは大学の研究室という次々に人が入れ替わる環境において、数年にわたるプロジェクトを行っていることによるものである。また、より美しいものを、より凝った物を得たいという人間の感性によるものである。これらの問題は、本質的に奥の深い問題ではあるが、現実的な解決策を考え、浄書第3版の設計思想とした。

3．　浄書参の概要

浄書第3版（以後：浄書参）はこれまでの研究課題に加え、多様な出力に対する拡張性と、そのためのページレイアウトにも重点をおいた。

また処理の部品化を考え、文書形態の展開処理ごとに処理単位をとり、レイアウトとその処理を別にした。各処理単位をうまく紙面上で配置することにより、多様な出力に対応できる。概要を図1に示す。

4．　設計思想

4．1　浄書計画の設計思想

我々は文書の入力系と出力系を切り放し、文書の出力処理において自動化できる部分はすべてプリンタ上で行おうと考えた。それにより以下のような利点が得られる。

1）資源の共有

2）処理の高インテリジェント化

である [1]。

日本語の出力処理には多くのハードウェア資源やソフトウェア資源を消費する。その資源をプリントサーバという形で多くの入力計算機から共有できる。

また、分離したことにより組版処理や、高品質な出力のための処理を入力計算機の能力によらずに行うことができる。将来、浄書マシンは組版のエキスパートシステム等を持った高度にインテリジェント化されたプリントサーバとなる。

現在でも和欧混合組版等を実現しており、実際ユーザはパソコン上のワープロソフトやワープロ専用機で作成した文書ソースを浄書マシンにより整形し、出力を得ることができる。本論文も浄書によって出力した。

4．2　浄書参の設計思想

1）入力と出力を分離する

入力系と出力系を切り放した環境は個人でパソコン等を持っているような我々の研究室では、利点の多い環境であった。自分の入力環境で初期入力を行えば、出力はプリンタ上の整形処理により行われ、高品質の出力を得ることができる。浄書参も入力と出力を分離して考える。

2）多種の文書形態を出力可能にする

図を含んだ日本語文書を処理するプログラムと、図を含んだ英語文書を出力するプログラムが、別のプログラムとして存在するような環境では、利用者は混乱をきたす。ユーザが直接幾つものツールを扱うことなしに、多形態な文書を出力可能とする。

3）拡張性を考慮する

我々の学科内を見回すと、作表プログラムや、楽譜出力等が要求される可能性もある。ユーザの要求する出力形態は、多様化していく。その際、既存のプログラムに

Design and Implementation of JOSHO V3．0
Atsushi KANO, Mikiko SUZUKI, Toshio SOUYA, Mitarou NAMIKI, Nobumasa TAKAHASHI
Depertment of Computer Science, Faculty of Technology, Tokyo University of Agriculture and Technology

はできるだけ手を入れたくない。そこで各処理対象を扱うプログラム間のインターフェースさえ注意すれば、それら後発の処理部も同一紙面上で描画させることが可能になるように考慮する。

5. 開発方針

我々は浄書参を開発するにあたって以下のような開発方針を定めた。

1) 各展開処理を独立とする

紙面上で多様な出力を行う際に、各展開処理部の干渉はできるかぎり避けたい。他を意識するとプログラミングが煩雑になる。そこで各処理プログラムは独立な部品として、他との干渉はできるだけ抑えるようにした。

2) 各展開処理部に紙面を意識させない

展開処理部が紙面に対しどの位置に展開を行うかを意識すると、部品としての独立性が低くなる。各展開処理部は自分に与えられた領域（箱）のみを意識して展開処理を行う設計とした。

3) 紙面の抽象化はユーザ拡張部で行う

我々はハードウェアの抽象化に対し、FMH（Frame Memory Handler）を用いている。ここでフレームメモリや漢字フォントの扱いを抽象化しているが、さらにこのFMHと展開処理部の間に紙面と箱の仮想化部を設け、展開処理を仮想化する。

4) OS／omicronを用いる

上記の様な特殊な処理を実現するために、われわれはOSとしてOS／omicronを選んだ。OS／omicronは当研究室で開発されたフル2バイト表現のアプリケーション指向OSである。そのOS／omicronのタスクフォース機能とユーザ拡張部の機能を利用している。また機能ではないが、内部コードが2バイトで表現されていることも、データの扱いに際し大きな長所となっている。

6. 処理の実際

6.1 ファイル構造

実際どの様なテキストを入力とするかを示す。本システムは基本的にバッチ型のシステムである。処理テキストのソースの構成を図2に示した。

文書というものはその文書を通して、一定のフォーマットによって形作られるものである。そこで初めに文書の骨格となる柱の位置、ノンブルの位置、それぞれの形式、用紙サイズ、標準となる文字書体等の文書形式の記述を行う。

次に文書の本体がくる。文書本体は1つの内容を1つのまとまりとして記述する。2種類以上のテキストを紙面上で表現したい場合は、図の第1展開内容の子展開内容のように並べて書く。また1つのテキストの中に別のテキスト（図等）を内包させたい場合は、入れ子にして表記する。このように紙面上のイメージと近い形でファイルとして記述する。

6.2 指令語

指令語はすべて@で始める。ノンブルの設定は@ノンブル設定といった具合いである。また、各展開処理部が独立であるため、現在の文書で標準として扱うフォントの種類なども記述する。

処理対象の本文のテキストには次のコマンドを初めと終わりに付け、囲む。

@（処理名）（箱指定子）{＜本文＞@}

＜本文＞の部分が展開処理部に渡される部分である。よってこの部分に展開処理部に対する文書ソースやコマンドを記述する。

処理名の部分には処理を行わせたい処理ルーチン名を書く。当然漢字名も自由に使える。マクロで抽象化させることも考えている。

また、箱指定子の部分には紙面上での展開位置、サイズ、出現ページ等を書く。1つまたは複数の箱を指定することができる。

7. おわりに

本稿では浄書参のアウトラインを示した。我々は今後、処理名と処理対象を一つのファイルに併記したことの評価や、複数形態の文書の混在によるドキュメントの存在価値等を評価していきたいと考えている。また、ビットマップイメージを考慮した入力系の強化を図っていきたい。

参考文献

[1] 里山元章，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．9，pp．1126－1134（1989）．

[2] 狩野敦，鈴木未来子，曽谷俊男，並木美太郎，高橋延匡：“文書処理システム浄書Ⅲ”，卓上出版シンポジウム報告集，(1988)．

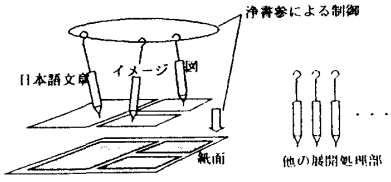

図1 浄書参の概要

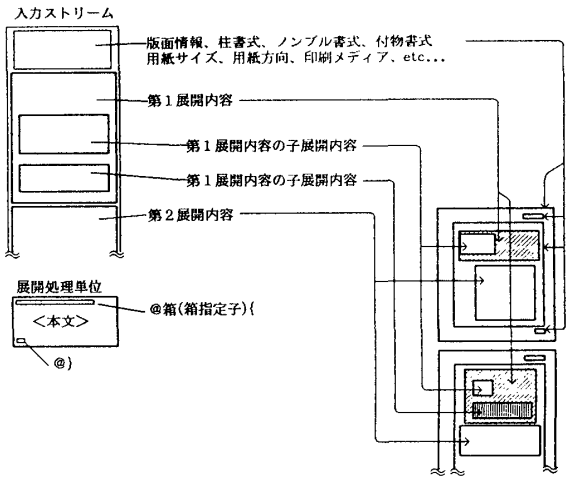

図2 浄書参に入力するファイルと展開仕様