

## 4J-9

# G4クラス3FAXのための ミクストモードデータエディタの実現方法

倉持 稔、橋本 隆子、山本 倫子、斉藤 裕、宮崎 章二  
(株) リコー ソフトウェア事業部

## 1 はじめに

ISDN (サービス総合デジタル網) の発達にともない、今後、ファクシミリは現在の主流機種である G3 機から G4 機へ移行することが予想されている。

G4 ファクシミリは厳密には3つのクラスに分類することができる。現在一般に G4 機と呼ばれているものは、この分類で述べればクラス 1 に相当する。これは従来の G3 機と同様にイメージ情報のみを扱うものである。これに対してクラス 2、クラス 3 機はイメージ情報の他にテキストをコード情報として扱うという機能を持つ (クラス 2 は受信のみ可)。このようにイメージ情報とコード情報を混在して扱うモードをミクストモードと呼び、これに関する標準勧告 [1] が 1988 年に CCITT (国際電気電話諮問委員会) によってなされている。今回我々は G4 クラス 3 のミクストモードを扱うエディタと、それを送受信するためのシステムを開発した。本稿では、ミクストモード原稿作成の実現方法について述べる。

## 2 ミクストモード原稿の特性

### 2.1 レイアウト特性

ミクストモード原稿の個々のページはブロックと呼ばれる最大 31 個の透過的な矩形領域から構成される。各ブロックの内容はテキストまたはイメージデータとなり、ブロック間の重ね合わせが可能となっている (図 1)。

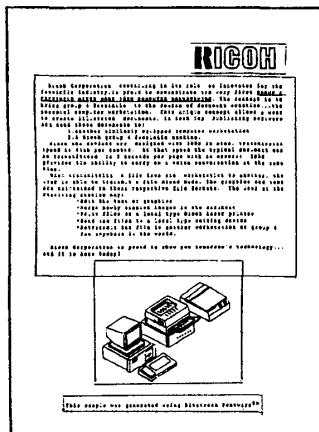


図 1: ミクストモード文書例

### 2.2 レイアウト構造

ミクストモードにおいては、スペシフィック・レイアウト、ジェネリック・レイアウトと呼ばれる2つのレイアウト構造が並列して存在している (図 2)。それぞれは、ルート/ページ/ブロックという階層を持ち、ジェネリック・レイアウトは同じ階層のスペシフィック・レイアウトから参照される。

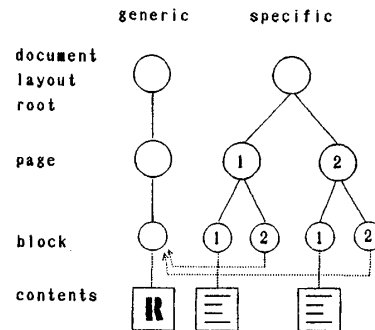


図 2: レイアウト構造

## 3 ミクストモードエディタ

今回、我々は上述したミクストモードの特性に従い、1) 重ね合わせ可能なブロックを扱い、2) スペシフィック/ジェネリック・レイアウト構造を持つエディタの開発を行った。動作環境としては、IBM-PC/AT 上の MS-DOS + MS-WINDOWS (Ver2.1) を使用した。

### 3.1 エディタの構成

本システムにおいては、テキスト/イメージデータを作成するエディタと、それをページ上に配置するエディタとに役割を分担し、1) テキストエディタ、2) イメージエディタ、3) レイアウトエディタ、という3種類のエディタを作成した。

### 3.2 ブロックの取り扱い

一般に、イメージとテキストを混在させてページ上に配置する DTP ソフトでは、イメージを配置した残りの領域にテキストを流し込むようなユーザインタフェースが考えられる (図 3, 1)。この場合、テキスト領域は必ずしも矩形にはならないのでミクストモードのデータ形式にするにはシステム側で複数の矩形領域に分割する等の

処理が必要となる。また、この方式の場合、意識的にイメージとテキストを重ねて配置したい時の指定方法が複雑になる可能性がある。

そこで、今回はミクストモードのレイアウト特性を正確に表現できることを第一に考えて、操作時にユーザがブロックという概念を意識して矩形領域を定義するというアプローチをとった(図3.2)。この場合、他のブロックと重なる時でも指定されたブロック内いっばいにテキストが流し込まれる。上述のレイアウトエディターにおいてはブロックの新規作成や移動、削除等の操作が行えるようになっている。

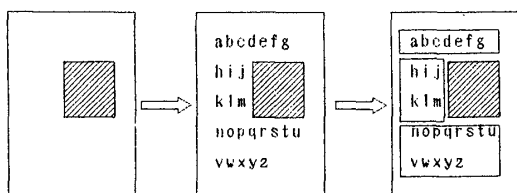


図 3.1: 流し込み方式

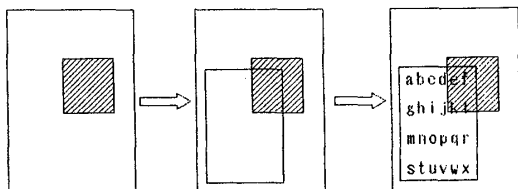


図 3.2: ブロック定義方式

### 3.3 エディター間の通信

今回エディターを3つに分けるに当たっては、個々のエディターを別々なアプリケーションとして意識せずに最終的にミクストモード原稿が作成できることを考慮した。そのため、レイアウトエディター上でブロックを定義後、レイアウトエディターから直接テキスト/イメージエディターを起動するユーザインタフェースを構築した。

また、これらのエディター間では MS-WINDOWS のメッセージ通信の機能を用いて、ブロックのサイズや、そこに入るデータのファイル名等の情報を交換している(図4)。この機能によって、例えばテキストエディターが起動された時には指定されたブロックと同じサイズになり、一行の入力幅もブロックサイズに固定される。テキストエディターが終了すると、ただちにブロック内にテキストが貼り込まれる。

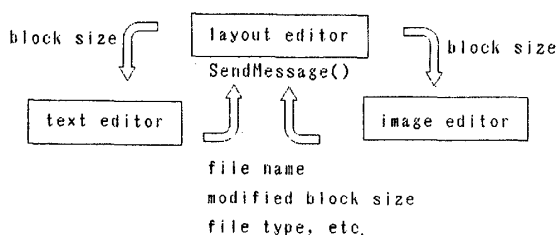


図 4: エディター間のメッセージ通信

### 3.4 エディター内のファイル管理

本エディターにおいて、各ブロックの内容となるデータファイルは『部品ファイル』もしくは『一時ファイル』として管理される。部品ファイルは、データの再利用を念頭におき、ユーザが明示的にファイル名をつけてホストのディスクに保存しておくファイルである。部品ファイルはテキスト/イメージエディターでセーブ操作を行うことによって作成され、レイアウトエディターの貼り込み操作を行うことでページ上に配置される。また、一度しか使用しないため明示的にセーブしないデータは、レイアウトエディター側が一時ファイルとして管理を行うようにした。一時ファイルは現在編集中の原稿に対する処理を終了させた時点で自動的に削除される。また、原稿全体を格納するためのファイルは、各ページのレイアウト構造を持ち、各ブロック内のデータファイルはアーカイブされた形で保存される。

### 3.5 ジェネリックレイアウトの実現方法

ジェネリックレイアウト構造を用いると、重複して使用したデータを繰り返し送信する無駄がなくなる。これを実現するために、編集中の原稿において同一の部品ファイルを何回使用しているかをレイアウトエディター内の管理テーブルによってチェックし、2回以上使用された部品ファイルをジェネリックレイアウト構造に組み込むことにした(図5)。

FAX 送信時において、ジェネリックなデータ本体は、使用回数にかかわらず一度だけ送り、その後そのデータを使用している各ブロックの位置等の情報を送るという形式をとる。これによって、送信データ量の大幅な節約が可能となった。

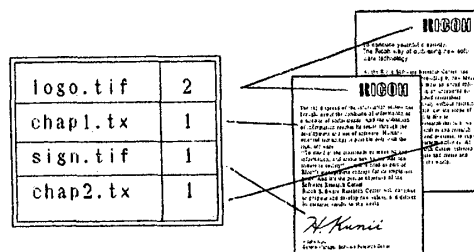


図 5: ジェネリックレイアウトの実現方法

## 4 おわりに

G4 クラス3 ファクシミリデータの形式であるミクストモードデータを作成するエディターの実現方法を述べた。テキストデータのコード化とジェネリックレイアウトの実現により、ファクシミリ送信データの量を大幅に減少させ、より高速な送信が可能になった。なお、今後は、ODA (Open Document Architecture) に従った論理的文書構造のサポート等も検討していきたいと考えている。

### 参考文献

- [1] CCITT Proposal for new T-series Recommendation, Mar., 1988