

1J-4

構造化文書(論理構造)処理システムの既存システムとの連携

諸星博司*, 山口哲生*, 長田格*, 川崎淳一**, 鬼頭隆志**
 富士通株式会社 基盤システム事業本部*, (株)石川富士通ソフトウェア開発 開発部**

1.はじめに

論理構造を持った文書を処理するために, FACOM G シリーズワークステーション上にEPODOCUMENT(構造化文書処理システム)¹⁾を開発した. この開発は, ワークステーション上に既に構築されている統合OAソフトウェア「EPOC-G」²⁾と連携して構造化文書処理を実現するように構築している.

本稿では, EPODOCUMENTとEPOC-Gの連携処理及び将来の展望について報告する.

2.システム構成

EPODOCUMENTとEPOC-G及びメディアファイル(文書ファイル等)の関係について以下に図1.に示す.

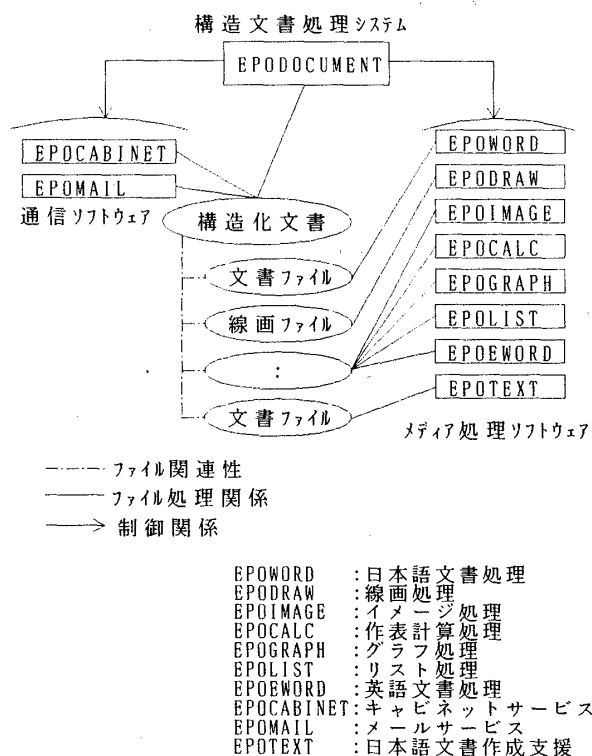


図1. システム構成

3.既存システムとの連携

EPODOCUMENTの開発に当たっては, 既存のソフトウェアとの整合性及び連携により, ソフトウェア及び文書ファイル等の既存の資産を有効に活用できるようにすることを前提にした.

このために, 木構造処理以外の処理に関しては, EPDOC-Gを徹底して利用するために新たなインタフェースを設けることにした. 次にメディアファイル種別ごとの詳細について述べる.

3.1 文書処理システムとの連携

文書処理システムとの連携は, 構造化文書処理システムとしても中心となる機能であり, 特に木目細やかな連携を行い自然なMMIを実現する必要がある. このために, 次のような点について特に考慮している.

(1)文書の連続性

構造化文書は, 複数の文書ファイルから構成されており, 木構造の終端ノードに対してそれぞれ別の文書が関係付けられている. そして, それらの複数の文書は連続性を持っているため, 独立した文書ファイルにあたかも連続した文書のように見せる必要がある.

・文書の作成 (図2.参照)

作成においては, 一連の終端ノードの文書(はじり)を作成する場合に, 上位ノード(本文)を選択する. すると, EPODOCUMENTが下位の木構造を解析し, EPOWORDに対して終端ノードに対する文書の作成指令が発行される. これにより, 文書の作成者は下位のノードの構造を意識することなく, 文書を次々と作成していくことができる.

・文書の更新/参照 (図2.参照)

更新/参照においては, EPOWORDにおいて処理状態にある時に, 文書の先端での前ページキー又は終端での次ページキーにより, EPODOCUMENTと通信を行い木構造上の前後の別文書ファイルへ移行することができる. 例えば, ノード“はじり”の終端において

次ページキーを押すとノード“参考文献”の文書の処理状態に移行する。

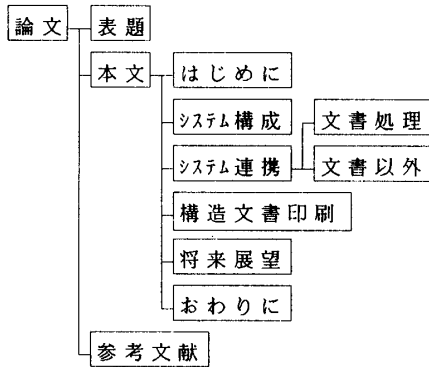


図2. 木構造の例

図3.に構造化文書処理画面と文書内容の処理画面の例を示す。

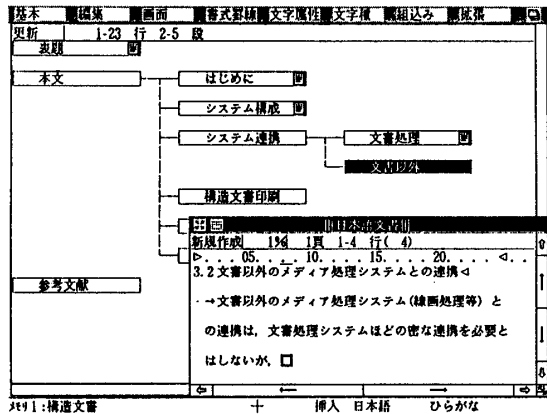


図3. 文書操作画面の例

(2)文書の書式の継承

既存の文書ファイルを、構造化文書として扱うためには、時として文書の書式（行当たり文字数等）が異なる場合がある。

EPODOCUMENTでは、構造化文書として各ノード毎に最低限の書式情報を持っており、これらは互いのノードとの関係により変更される可能性がある。

文書を処理する場合には、文書ファイル内の書式情報が無効となり、木構造上の書式情報がEPOWORDに継承されることにより、構造化文書としての書式で更新／表示を行うことができる。

3.2 文書以外のメディア処理システムとの連携

文書以外のメディア処理システム（線画処理等）との連携は、文書処理システムほどの密な連携を必要とはしないが、構造化文書処理システムとしては重要な構成要素である。このために、次のような点について特に考慮している。

(1)書式指定によるメディアサイズの決定

各メディアの作成時には、木構造上の書式情報がEPOC-Gに通知されることにより、構造化文書の書式に従ったサイズのメディアを作成することができる。

(2)メディアサイズによる書式の決定

既存のメディアを利用する場合には、木構造上の書式情報の範囲内でメディアサイズに従って書式が割り当てられることにより、適切な書式で扱うことができる。

4.構造化文書の印刷

作成された文書においては、必ず印刷が必要になる。これに対してEPODOCUMENTでは、ODA (Office Document Architecture)³⁾で規定されている厳密な割付け規則を、現状では適用していない。

構造化文書の各ノードには、前後のノード間の相対的な位置関係を定義する情報を持つ。これにより、内容を構成する各ファイルを印刷する場合に、その都度相対的な位置が決定されて印刷される。ゆえに、編集においては処理中のページが何処に割付けられるか規定することはできない。ただし、ページ単位の切り分けは可能である。

5.将来の展望

図1.で示したシステム構成の中で、キャビネットサービス、メールサービス、日本語文書作成支援及び英語文書については現在未開発であり、今後は各EPOCについて順次適応を進めていく。特に、日本語文書作成支援との連携による目次／索引作成システム等の開発が重要課題であると考えている。

また、ODAで規定している、割付に関する定義を取り込み、本格的な割付のサポート及びシステム間の交換に対する検討を行っていきたい。

6.おわりに

文書処理システムの周辺は、今後も様々な機能の要望が行われてくるであろうし、様々なシステムが開発されてくるであろう。構造化文書処理システムは、それらとの連携を充実させていくことで、より効率的な文書処理システムを構築していくことが必要と考えている。

[参考文献]

- 1) 川崎淳一, 鬼頭隆志, 諸星博司, 山口哲生, 長田格: 構造化文書(論理構造)処理システムの開発, 情報処理学会第38回全国大会(1989).
- 2) 幡鎌博, 古川利満, 長田格: オフィスシステムにおけるOA処理体系, 情報処理学会第37回全国大会(1988).
- 3) ISO 8613 : Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format (March 1988).