ソフトウェア工学　46－22
（１９８６．２．７）

ソフトウェア設計文書生成のための汎用レポータ

寺嶋　祐一
日本電気ソフトウェア株式会社
第一応用システム事業部

紫合　治
日本電気株式会社
ソフトウェア生産技術研究所

１．はじめに

システム開発において作成される技術文書は、一般の文書に比べて、定型的で、大量であることが多い。また、図や表で表現される部分が比較的多い。このような技術文書は、通常、手書きか、ワードプロセッサなどを用いて作成されるのが普通であるが、作成量が多いと手間が掛かり不都合である。また、コンピュータを用いた文書出力機構としては、ＵＮＩＸ上で動作するｒｏｆｆ機構［１］や書式（フォーマット）とデータを分離した埋め込み展開機構［２］などがあり、前者はページ制御、後者は表処理の点でそれぞれ優れているが、前者は技術文書で使用されることの多い表操作が弱く、後者は書式から出力イメージを想像しにくい、複数ページの文書処理が弱いなどの欠点がある。出力形式に表を多く含み、定型的で、作成量が多い技術文書の生成には、効率面や将来の変更容易性から、これらの長所をうまく統合したアプローチが効果的であろう。

ここで報告する汎用レポータは、設計文書をはじめとする技術ドキュメントの整形、編集を主目的とした文書生成ツールである。このツールでは、文書の内容に相当するデータと、実際の出力イメージに似せた出力フォーマット、および必要に応じて指定できる簡易な編集コードを組み合わせて、任意の形式のドキュメントを容易に生成できる。特に、別々に出力した文書の一部同士を連結し、改めてページ番号や章節番号をふり直したり、目次や一覧、索引等を自動生成する機能を有する。

本報告では、この汎用レポータの機能とその特徴、およびユーザがカストマイズできるフォーマットや結合指示等の指定方法について述べる。

また、このツールの応用として、ソフトウェアＣＡＤを指向した設計支援システム（ＳＤＭＳ）［３］の設計文書生成サブシステム（ＳＤＭＳ／レポータ）を紹介し、実際に詳細設計書を出力した例を示す。これらの説明および適用評価を通じて、ソフトウェア設計文書生成のための支援について考察する。

２．汎用レポータの機能要求

文書生成支援として一般的に求められる機能は、小は活字のポイントや種類の選択機能から、大は冊子形式の編集機能に至るまで、きわめて多岐にわたる。これらのすべてを網羅するというのも一つの考え方であるが、目的に応じて取捨選択し、できるだけコンパクトに仕上げることも、もう一方で重要であろう。ここでは、基本的には後者の考えに従っている。

ここでの前提条件および目標は、次のとおりである。

（１）主として技術文書の編集を対象とする。
（２）文字列の編集の他、特に、表形式の編集を強化する。
（３）章節やページの番号付けを柔軟に、かつ、動的に制御する。
（４）大量の文書を効率良く扱えるように、出力形式の設定を工夫する。
（５）出力形式の設定、変更は自由に行なえるようにする。特に、設定時点で出力イメージが容易に想像できるよう工夫する。
（６）文書の部分作成と、それらの統合編集を可能とする。
（７）ワードプロセッサなどでは実現できない冊子形式の編集を強力に支援する（そのまま綴じれば文書が出来上がる程度のレベル）。
（８）自動収集または自動生成できる情報は極力自動編集する（目次、索引など）。

（9）出力装置としては、当面、ごく普通のプリンタを想定しておく。

3．汎用レポータの機能概要

上記の機能要求に基づく文書生成支援は、一口でいえば、例えば、

「与えられた文書データを、（ユーザによってあらかじめ定められた）出力イメージに似せたフォーマットを用いて、表や文書の形式で展開編集し、そのようにしてできた部分文書を連結したり、部章節番号およびページ番号をふり直したりして、冊子形式で出力する。その際、指定位置に、目次、索引を自動的に生成し、また、生成文書の部分印字も行なえるようにする。」

というものである。

このような考えに基づいて設計された、汎用レポータの機能の一覧を表1に示す。

表1．汎用レポータ機能一覧

1	基本処理	文字詰め、中断、空白行挿入
2	ヘッダ、フッタ	3部タイトル、ヘッダ定義、フッタ定義、出力抑止
3	ﾌｫｰﾏｯﾄ付ﾃｷｽﾄ展開	フォーマット名指定、データ部定義
4	見出し、枠線処理	見出し引き継ぎ／抑止、クリア 枠線（閉じ、調整、連結）
5	ｻﾌﾞﾌｧｲﾙの挿入	別のファイル内容をコピー挿入
6	ページ幅と字下げ	ページ幅設定、字下げ、一時字下げ設定
7	ページ制御	ページ長、ページ番号、改ページ、ブロッキング、空白ページ挿入
8	章立て	部タイトル（番号付き／なし）、章節タイトル（番号付き／なし）
9	索引	索引項目、グループｉｄ定義
10	ﾕｰｻﾞ定義レジスタ	数レジスタ設定／解除、文字レジスタ設定／解除
11	整形	ページ禁則実施／抑止、中央揃え、一連番号付き段落、横線挿入
12	結合指示	ファイル連結、目次生成（サイズ）索引生成（サイズ）、タイトル
13	番号の再付与	部番号再付与、章節番号再付与、ページ番号再付与

汎用レポータは設計文書生成支援の核部分として設計されたが、その特徴をまとめると以下のようになる。

（1）簡潔なフォーマット定義
- 出力イメージに近く直観的にわかり易い
- 表形式指定も容易
- フォーマットを入れ子にして用いることができ、複雑な指定にも対応可能
- 縦方向を圧縮した簡潔な表現（一行表現）で、データ長に応じた自動伸張を実現
- フィールド毎にデータの埋め込み方法をきめ細かく設定可能：
  長いデータの＜折り返し／切り捨て＞
  短いデータの揃え＜左／中央／右＞
  禁則＜なし／日本語／ASCII＞

（2）目次、索引の自動生成（ページ情報付き）
- 生成位置を指定可能
- 索引（または一覧）は同時に複数種類の指定が可能

（3）容易なカストマイズ
- 出力の形式、位置、順序等の変更のほとんどはエディタで即座に対処可能
- 高度なユーザインタフェース（データやフォーマットと混在自由な編集指定など）の提供により、新コマンドの追加等に容易に対処可能

（4）強力な編集機能（簡易な編集指示）
- フォーマットを利用した効率の良いテキス

ト展開。埋め込みフィールド毎に折り返しや切り捨てを自動処理。表の各ボックスの伸縮、枠線始末なども自動調整可能。

－複数の部分文書の結合

－ヘッダ、フッタ、改ページ見出し処理

－章立て（部章節番号）制御

－ページ番号制御

－番号付きパラグラフ、字下げ処理

－（システム定義／ユーザ定義）レジスタ処理

（５）暗黙の自動処理（デフォルト処理。処理の指定があれば、そちらを優先）

－出力行の文字詰め（fill処理）

－データが短い時は左詰め、長い時はフィールド幅に従って折り返し

－ページ禁則

<BB>=(b1 b2)

項目	説　明
＊＊＊＊	＊＊＊＊＊＊＊＊

（ａ）表フォーマットパターン

```
.fm BB
.(t
(  "abc"
   "xxxのデータ"
   "MD－1985"
   "1985年の統計資料"
   "１２MM"
   "出力フォーマットのパラメータ"
.)t
```

（ｂ）文書データ

項目	説　明
abc	xxxのデータ
MD－1985	1985年の統計資料
１２MM	出力フォーマットのパラメータ

（ｃ）埋め込み展開後のイメージ

図１．表展開の例

－表形式での枠線自動閉じ処理

（表データ終了時、表の途中での改ページ時）

－埋め込みデータのないフィールドへの空白詰め処理

目　次

－ 1 －

図２．自動生成目次の例

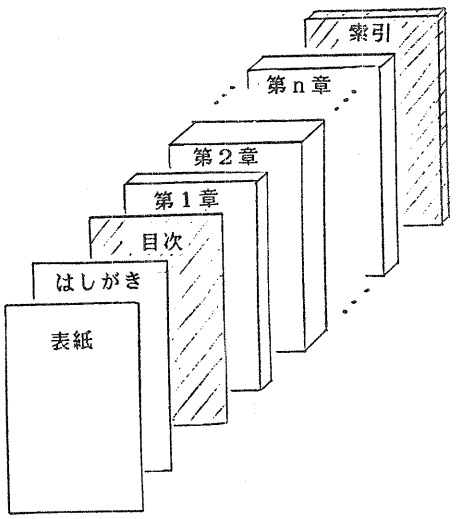


図３．冊子形式編集のイメージ

図１－３に上記機能および処理イメージの例を示す。図１に表展開の例、図２に自動生成される目次の例、図３に冊子形式編集のイメージ、をそれぞれ示す。

４．汎用レポータの処理の流れ

図４に汎用レポータの処理の流れを示す。汎用レポータの処理は大きく、展開処理、結合処理、印字処理の３つのフェーズからなる。

展開処理は文書データを入力し、指定されたフォーマットパターンの埋め込み位置に、データを埋め込みながら展開し、所定の大きさのページに分割して出力する。表１に挙げた各種の編集や制御は、編集コードの形で、展開処理の入力として与える。この編集指定は、必要に応じて、文書データ中およびフォーマットパターン中のどちらにも記述することができ、効果も同じである。

展開処理では、同時に、次の結合処理あるいは印字処理のための文書情報制御表を生成し、文書本文の先頭に付加して出力する。図で、展開処理の出力ｆ１に施した斜線部分がそのイメージである。

ここで生成される制御情報は、ページ情報、目次（部章節項目）情報、索引（一覧）情報などであり、それぞれ次のような項目とその値を管理する。

（１）ページ情報部：
　　相対ページ、絶対ページ、ページオフセット（先頭／埋め込み）など

（２）目次情報部：
　　部章節番号、表題、ページ番号、埋め込みオフセットなど

（３）索引情報部：
　　索引項目、索引グループｉｄ、参照ページ番号など

図５に文書情報制御表つき文書の構造を示す。

結合処理は、展開処理で生成された制御情報表つき文書のいくつかを結合し、制御情報と本文を別々にマージして、一つの制御情報つき文書にまとめて出力する。また、このような文書の連結処理と並行して、部章節番号やページ番号を通し番号にしたり、部分的にふり直すことができる。部章節番号やページ番号のふり直しは、一つあるいは複数の入力文書単位で行なえるし、いくつかの

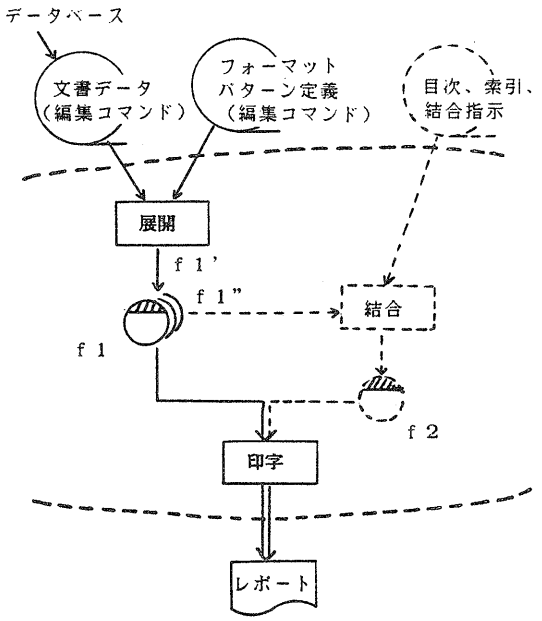


図４　汎用レポータの処理の流れ

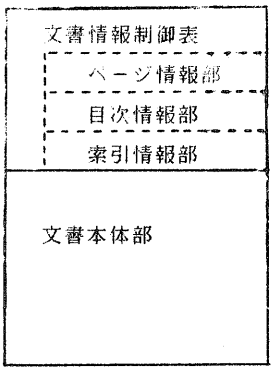


図５　文書情報制御表つき文書の構造

指定を同時に行なうこともできる。これらの指示は、すべて結合処理の一方の入力である結合指示にて行なう。

結合処理の出力ｆ２は、入力ｆ１、ｆ１’、ｆ１’’．．．と同一の内部構造を持つので、ここでの出力を再び別の結合処理の入力として用いて（図の破線の流れ）、結合処理を繰り返して行なうことができる。

結合処理は、また、結合指示で指定される任意の位置に、目次、索引、各種一覧などを自動生成して挿入する機能を有する。目次は、部章節番号とそのタイトルを自動収集し、ページ番号つきで出力する。索引および一覧は、展開処理の入力データあるいはフォーマットであらかじめ指定されたデータ項目を、グループ別に収集し、指定に応じて、索引や一覧の形式で、ページ番号つきで出力する。目次および索引の生成は、その内容に、該当項目の章節番号あるいはページ番号を含むという性格上、これらの番号がすでに確定していることを前提としている。したがって、目次および索引の生成指示に限っては、もしあるとすれば、一連の結合処理の最後の結合指示中でなければならない。結合指示の内容については次章で説明する。

実際の文書作成では、節など小単位で作成した文書を、とりあえず章の単位でまとめておき、全体がそろった時点でそれらを一つにまとめて、ページ番号や章節番号を通しでふり直すといったことを行なうが、ここでの処理はそれら一連の手続きを自然に支援している。

印字処理は、文書情報制御表つき形式の文書を印字する。従って、展開処理、結合処理のいずれの出力も任意のタイミングで印字できる。通常は全文出力であるが、指定により、目次のみ、索引のみ、あるいは指定ページ（範囲指定可）のみの出力が可能である。

５．カストマイズ情報

汎用レポータにおいてユーザに開放されるカストマイズのための情報は、フォーマットパターン定義と結合指示の２つである。これらの情報は普通定数ファイルとしてシステムに与えられるが、そのためカストマイズ情報の変更はエディタ操作だけで容易に行なえ、プログラムを変更する必要はない。

（１）フォーマットパターン定義ファイル

展開処理で用いるフォーマットパターンをこのファイルにあらかじめ定義しておく。一回の展開処理で用いるパターンはすべて同一のファイルになければならない。一般的なデータ埋め込みパターンのほか、そのまま出力する固定パターンも定義できる。図６にフォーマットパターン定義の例を示す。図で、＊で示される部分にデータが埋め込まれる。

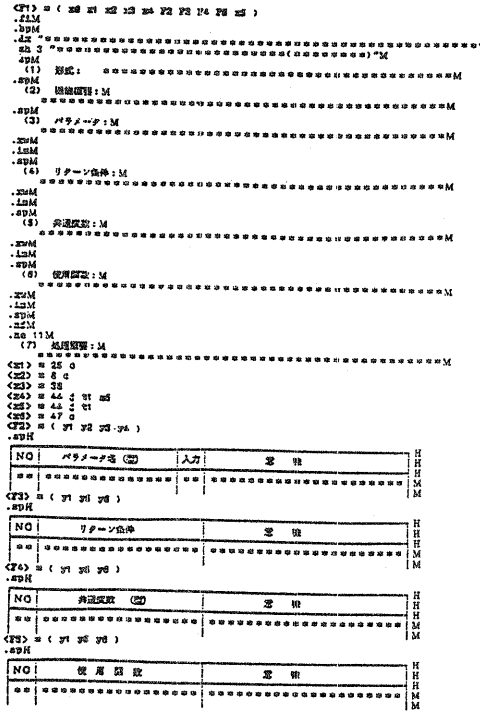


図６．フォーマットパターン定義の例

フォーマットが複雑な場合には、まず大局的な位置と大きさを決めておき、次に詳細なフォーマットを決めるといった段階的な設定が可能である。具体的には、フォーマット設定時に入れ子を許すことで実現している。例えば、図6でフォーマットF1はF2，F3，F4．．．などからなるが、F2，F3，F4．．．は、それぞれ別に詳細な定義がある。

（2）結合指示ファイル

結合処理のための指示をこのファイルに定義しておく。指示は、

a）文書の連結

b）部章節番号の再付与

c）ページ番号の再付与

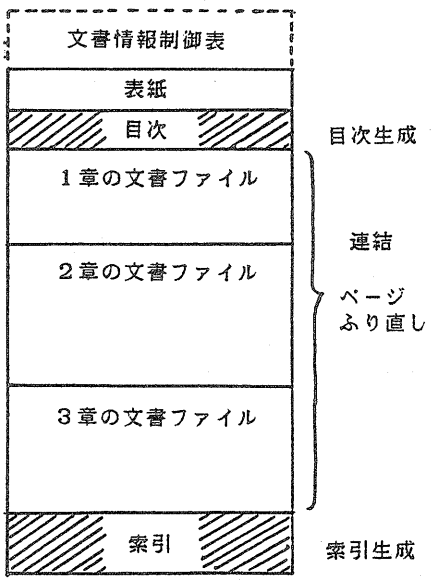
d）目次の自動生成指示

e）索引（一覧）の自動生成指示

の5種類であり、目次以外は複数回指定可能である。図7に、結合指示の例とその出力イメージを示す。

```
. l f  表紙
. c n  目次 <---(目次生成指定)
. (p  1  <---(ページふり直し指定)
. l f  1章の文書ファイル名 }
. l f  2章の文書ファイル名 }(文書ファイ
. l f  3章の文書ファイル名 }ルの連結)
. ) p      <---(ページふり直し終わり)
. x p  索引 <---(索引生成指示)
```

(a) 結合指示



(b) 出力イメージ

図7．結合指示とその出力イメージ

6．設計文書出力

－ソフトウェア設計支援システムへの応用－

汎用レポータの応用として、ソフトウェア設計支援システム（SDMS）において、設計文書生成サブシステム（SDMS／レポータ）を実現した。以下で、その利用例を紹介する。

SDMSは、設計情報についてのデータベースを持つが、SDMS／レポータはそこから設計文書に必要な各種の設計情報を検索し、汎用レポータの入力形式に整形する。汎用レポータでは、あらかじめ、設計書出力に必要な、フォーマットパターンを定義しておき、整形されたデータを受けて、それらを展開し、編集出力する。もし、文書の連結の必要がなければ、目次、索引を生成して、一気に印字まで行なうこともできる。

図8－11に一連の応用例を示す。図8は設計文書生成サブシステムの構成、図9はデータベースから取り出された文書データ（汎用レポータの入力形式）の例、図10は文書本体部の出力例、図11は自動生成された索引（ここでは一覧）の例、をそれぞれ示す。自動生成目次については、図2に示したので、ここでは省略する。

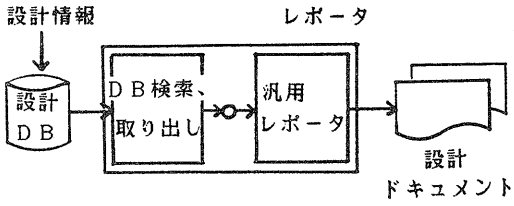
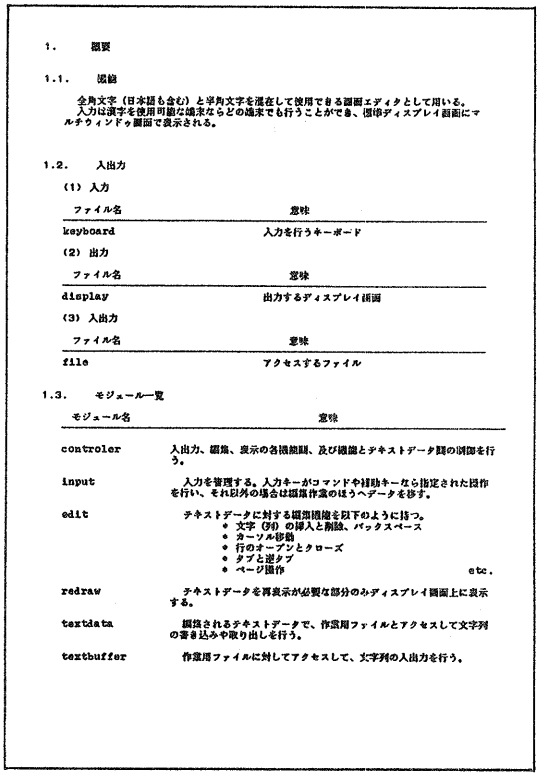


図8．設計文書生成サブシステムの構成

```
.fm F1
.(t
("aefsave (  textdata  ) 指定されたファイルからの情報のセーブ"
"指定されたファイルからの情報のセーブ" "aefsave"
"int        aefsave();"
"  ページテーブル、行情報テーブルとバッファテーブルとを参照して、
最新の情報を指定されたファイルへ順次書き出す。"
("1""char     *fn""I""セーブファイル名")
( )
("1""llast""  テキストの最終行の値を保持し、ファイルセーブ時の最終行の設定や、
文字列の1行入出力時の既存行かどうかの判定に用いる。"
"2""adebtbl""  バッファテーブルで、入出力文字列の一時的なバッファとして
用いる。")
("1""static  int      inbuf();""  バッファテーブル上に指定された行があるか
どうかを調べ、あればそのバッファテーブル上の行番号を、なければ－1を返す。"
"2""static  int      getfile();""  作業用ファイルから、指定された行の文字列
を取り出し、出力バッファにコピーする。")
)
.)t
```

図９．文書データの例



1. 概要

1.1. 機能

全角文字（日本語も含む）と半角文字を混在して使用できる画面エディタとして用いる。
入力は漢字を使用可能な端末でも行うことができ、標準ディスプレイ画面にマルチウィンドウ画面で表示される。

1.2. 入出力

(1) 入力

ファイル名	意味
keyboard	入力を行うキーボード

(2) 出力

ファイル名	意味
display	出力するディスプレイ画面

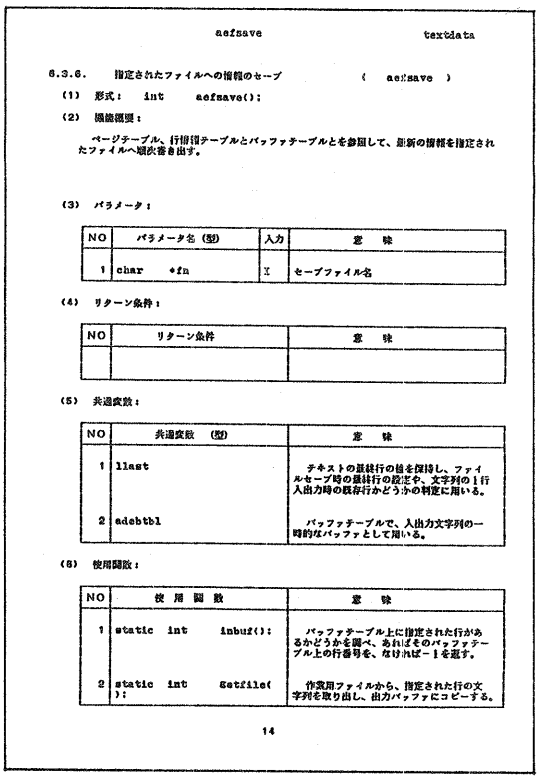
(3) 入出力

ファイル名	意味
file	アクセスするファイル

1.3. モジュール一覧

モジュール名	意味
controler	入出力、編集、表示の各機能間、及び機能とテキストデータ間の制御を行う。
input	入力を管理する。入力キーがコマンドや移動キーなら指定された処理を行い、それ以外の場合は編集作業のほうへデータを渡す。
edit	テキストデータに対する編集機能を以下のように持つ。 ・文字（列）の挿入と削除、バックスペース ・カーソル移動 ・行のオープンとクローズ ・タブと逆タブ ・ページ操作 etc.
redraw	テキストデータを再表示が必要な部分のみディスプレイ画面上に表示する。
textdata	編集されるテキストデータで、作業用ファイルとアクセスして文字列の書き込みや取り出しを行う。
textbuffer	作業用ファイルに対してアクセスして、文字列の入出力を行う。

（ａ）基本仕様書の例



aefsave　　textdata

6.3.6. 指定されたファイルへの情報のセーブ　　（ aefsave ）

(1) 形式： int　aefsave();

(2) 機能概要：

ページテーブル、行情報テーブルとバッファテーブルとを参照して、最新の情報を指定されたファイルへ順次書き出す。

(3) パラメータ：

NO	パラメータ名（型）	入力	意　味
1	char　*fn	I	セーブファイル名

(4) リターン条件：

NO	リターン条件	意　味

(5) 共通変数：

NO	共通変数　（型）	意　味
1	llast	テキストの最終行の値を保持し、ファイルセーブ時の最終行の設定や、文字列の1行入出力時の既存行かどうかの判定に用いる。
2	adebtbl	バッファテーブルで、入出力文字列の一時的なバッファとして用いる。

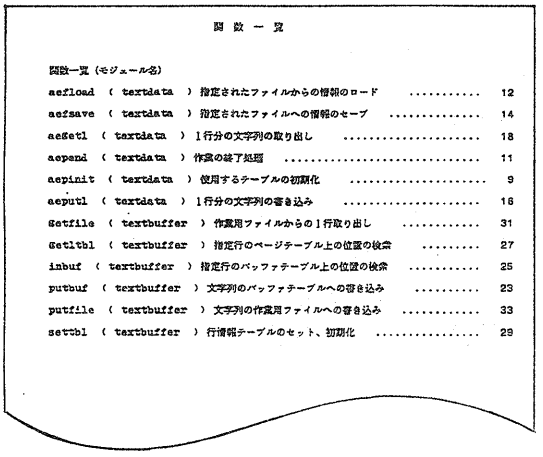
(6) 使用関数：

NO	使　用　関　数	意　味
1	static int inbuf();	バッファテーブル上に指定された行があるかどうかを調べ、あればそのバッファテーブル上の行番号を、なければ－1を返す。
2	static int Getfile();	作業用ファイルから、指定された行の文字列を取り出し、出力バッファにコピーする。

14

（ｂ）詳細設計書の例

図１０．文書本体部の出力例

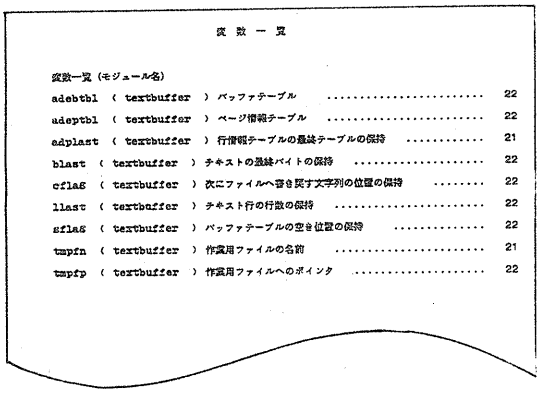


関 数 一 覧

関数一覧（モジュール名）

aefload （ textdata ） 指定されたファイルからの情報のロード 12
aefsave （ textdata ） 指定されたファイルへの情報のセーブ 14
aefsetl （ textdata ） 1行分の文字列の取り出し 18
aepend （ textdata ） 作業の終了処理 11
aepinit （ textdata ） 使用するテーブルの初期化 9
aeputl （ textdata ） 1行分の文字列の書き込み 16
fsetfile （ textbuffer ） 作業用ファイルからの1行取り出し 31
fsetltbl （ textbuffer ） 指定行のページテーブル上の位置の検索 27
inbuf （ textbuffer ） 指定行のバッファテーブル上の位置の検索 25
putbuf （ textbuffer ） 文字列のバッファテーブルへの書き込み 23
putfile （ textbuffer ） 文字列の作業用ファイルへの書き込み 33
settbl （ textbuffer ） 行情報テーブルのセット、初期化 29

（a）関数一覧の例

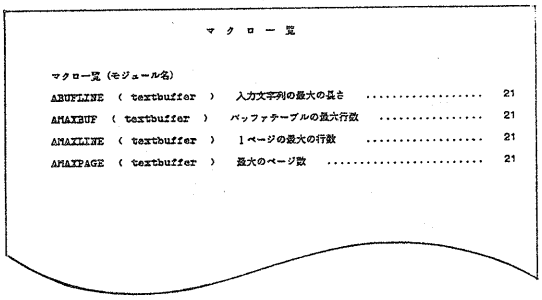


変 数 一 覧

変数一覧（モジュール名）

adebtbl （ textbuffer ） バッファテーブル 22
adeptbl （ textbuffer ） ページ情報テーブル 22
adplast （ textbuffer ） 行情報テーブルの最終テーブルの保持 21
blast （ textbuffer ） テキストの最終バイトの保持 22
cflag （ textbuffer ） 次にファイルへ書き戻す文字列の位置の保持 22
llast （ textbuffer ） テキスト行の行数の保持 22
sflag （ textbuffer ） バッファテーブルの空き位置の保持 22
tmpfn （ textbuffer ） 作業用ファイルの名前 21
tmpfp （ textbuffer ） 作業用ファイルへのポインタ 22

（b）変数一覧の例



マ ク ロ 一 覧

マクロ一覧（モジュール名）

ABUFLINE （ textbuffer ） 入力文字列の最大の長さ 21
AMAXBUF （ textbuffer ） バッファテーブルの最大行数 21
AMAXLINE （ textbuffer ） 1ページの最大の行数 21
AMAXPAGE （ textbuffer ） 最大のページ数 21

（c）マクロ一覧の例

図１１．自動生成索引

7．おわりに

汎用レポータを、設計文書生成支援ツールに組み込み、実際に設計文書を生成することで、一通りの評価を終えた。機能面では、一般の文章と表からなる文書について通常要求される処理はほぼ実現できる見通しがついた。

ただし、

－図形を含む文書の処理について

－ユーザがカストマイズするフォーマットパターンや結合指示の簡易化について

などの点が不十分であり、これらについては今後さらに検討していきたい。

汎用レポータは、もともと設計文書生成を支援する目的で、フォーマット展開機能のほか冊子形式の文書の編集機能などを強化してきたが、あるユーザでは、このシステムを設計文書ではない、複雑な単票のフォーマットとデータの管理に応用している。このような事例からも、

－独立したツールとしての汎用レポータの洗練について

別途、再検討したいと考えている。

参考文献

［１］UNIX PROGRAMMER'S MANUAL.Volume 2A. 1979

［２］山城他「ソフトウェアのドキュメント作成支援環境の構築」、情処ソフト工学研究会、85-SW-44-3.1985

［３］岸、紫合「ＳＤＭＳにおける設計情報画面入力機能について」、情処30回大会、1985.3