16. ソフトウェア情報のシステム化について

九州大学 竹内 章, 大槻説乎
中央計数施設 Akira Takeuchi Setsuko Otsuki

1. はじめに

TSSの普及に伴い、自室において各所の計算センターを利用する機会が増加してきた。このような場合、マニュアルや利用の手引き等の資料が手元に揃えてあるとは限らず、不便を感ずることがある。これらの資料が計算機内部に貯えられており、必要に応じて情報の検索を行い、端末に表示できることが望まれる。

さらに、マニュアル等に限らずCAIテキスト形式のものを含めて、知識の伝達を目的とした各種の情報が計算機に用意される場合が増えるであろう。これら多くの情報が個々に計算機に記録されているのでは、利用者が自分に必要な情報を選び出し正しく利用することは容易ではない。はたして、必要な情報が貯えられているのかどうかを知ることすら容易ではなかろう。

BOOKHシステムは、TSS端末のための知識伝達を目的とした情報の作成、管理、利用手段を与えるものである。知識伝達を目的とした情報とは、具体的には会話形式プログラムであり、以後「テキスト」と呼ぶことにする。

個々に作成されたテキストは自動的に組織化され、利用者はBOOKHの利用法さえ知っていれば全テキストを通じて情報検索を行うことができる。

BOOKHの特徴

BOOKHは次のような特徴を持っている。

(1) テキストは、その作成者にも利用者にも親しみやすい形態となることを考慮し、外形は我々が幼い時から親しんでいる「本」と同様に題名、目次、内容、索引から成る構造を持っている。

(2) テキストの作成は、BOOKと呼ぶ編集言語によって行う。テキストにつけられるキーワード及び索引は、BOOK処理系により自動編集される。

(3) 特定のテーマを扱ったテキストについては、内容の仕様の入力によってテキストを自動生成する。

(4) テキストの目次はインデンテーションをつけて記述し、これによってそのテキストに含まれている概念相互の階層関係を表わす。これを目次の階層と呼ぶ。さらに、上位概念のテーマについて書かれたテキスト中で、下位概念のテーマを記述したテキストを呼び出すことができるようになっている。このような関係は概念の階層構造に従う一連のテキストの半順序集合を作る。

このような半順序関係で結合しているテキストの集合は一般に複数種類できるが、それぞれのテキストの集合の最大元は、「INFORMATION」と呼ぶテキストの下位概念として記述される。従って、すべてのテキストは一つの半順序集合として表現される。テキストINFORMATIONは、その計算機のシステム情報の全種類を統合表示するテキストとなる。

(5) 個々に作成されたテキストの情報を組織化するため、テキストの階層、目次の階層に従い、テキストのキーワードや索引項目を用いて、検索情報の自動生成を行う。これによって、全テキストを通じて情報検索・質問応答が可能となる。

(6) 検索情報の作成・変更はBOOKHが自動的に行うので、テキスト作成者が検索シ

第21回プログラミング
シンポジウム 1980. 1

ステムの存在を意識する必要がない。同時に、情報管理の完全性が保障される。

2. システムの設計と概要

2.1 設計方針

設計に当っては、テキストの作成者、利用者、管理者の3つの立場から検討する必要がある。

(A) テキスト作成者の立場からの要請

(A1) 情報提供者は、情報処理の専門家でない場合が多い。従って、作成手段が容易に会得できなければならず、新しい問題向言語を導入することは抵抗が多い。

(A2) テキスト実行時には、作成した内容をそのまま表示する場合と、表示された内容に関して利用者との会話を通し、情報処理を行う場合が考えられる。作成した内容を「画面」と呼び、画面に対して処理を行うことを「画面の運用」と呼ぶ。画面の作成に際しては、作成者が画面自身を目で確認しながら入力できることが望ましい。

(A3) 画面の運用を行うためには、作成者の意図するアルゴリズムを自由に表現できる必要がある。

(A4) テキスト作成者は、個々のテキスト作成に専念すべきで、組織化に関しては特別の注意をはらうべきでない。

(A5) 情報提供者が不特定多数となる可能性のあるテキスト（例えば、プログラムライブラリやTOOL BOX使用法など）については、簡単なコマンドによってテキストの部分作成が可能でなければならない。

(B) 利用者の立場からの要請

(B1) 全てのテキストを通じて、操作の一様性を保つ必要がある。作成者によって使用手段が変ることは混乱の原因となる。

(B2) テキストを始めから終りまで読むだけでなく、必要な部分のみ選出し調べることができなければならない。

(B3) ライブラリ使用法等に関するテキストは、読みやすさの観点から、作成者によらず内容の形式が同一でなければならない。

(B4) 検索したい事柄が利用者自身漠然としている場合にも対応できるように、上位概念から詳細事項へと検索を進めることが可能でなければならない。

(B5) 個々のテキストは異なる作成者により独立に作成されたものであるが、組織化され階層化された全テキストを通して質問応答を行う場合、全体があたかも一つの情報源のごとく振舞わねばならない。

(C) 管理者の立場からの要請

(C1) テキストの作成・変更の記録が残ること。

(C2) テキストの作成・変更に応じて、検索情報が自動生成・修正されること。

(C3) テキストの階層構造を管理し、階層関係に矛盾したテキストの作成・変更を防止する。特に、テキストを消去する権利は、管理者にのみ与える。

(C4) 個々に作成されたテキストは組織化され、一つの検索系を形作るが、一つのテキストの変更を行う場合、検索系全体に影響を与えることなく該当部分のみの修正が可能であること。

(C5) 後日のテキスト修正のため、実行形式となったテキストとそのソースの間に矛盾があってはならない。

これらの要請を満たすためには、BOOKHに次のような構造を与えればよい。

(1) A2を満足するために、画面の作成と画面運用のためのプログラムの作成とを独立に行う。画面の作成とその編集・運用が一つのプログラムとして混在すると、プログラムの構造が判りにくい上、画面を直接見ながら作成できないので作成者の負担が非常に大きくなる。

(2) A1, A3の要請に対しては、作成者の得意とする汎用プログラミング言語を用いるのが最善である。しかし、汎用言語はテキストの編集・運用に本来固有であるはずのデータ型，操作，制御をアルゴリズムとして表現しなければならないという不便さがある。これを解決するために、汎用言語を親言語とし、その上に固有のデータ型や文を加えた言語BOOKを設計した。

一般に親言語方式をとった場合の欠点として、親言語に翻訳した後、親言語処理系から出されたエラーメッセージともとのプログラムとの非整合性があげられるが、この欠点を補うため、画面の操作を単位とするpageという構造を導入した。

(3) B1, B2を満たすために、利用者の操作命令をBOOKに導入した。上記のpage構造は、利用者の操作とテキストの構造を整合させる働きをも兼ねている。

(4) A1, A5, B3を満たすため、特定の問題に対するテキスト作成には専用のサブシステムを用い、内容の仕様記述により画面の作成を行い、同時にBOOKによる運用プログラムの生成を行う。これによって、BOOK文法を知らなくともテキストの作成が可能となる。

(5) A4, B4, B5, C2, C4, C5を満たすために、情報管理のためのサブシステムによって個々に作成されたテキストを階層に従い半順序をつけて登録し、目次の階層に従う検索情報の自動生成を行いテキストを組織化する。同時に、実行形式プログラムとソースプログラムを対にして管理下におく。

(6) C1, C3を満たすために、情報管理のためのサブシステムは、管理者を除いてはBOOKコンパイラを通してのみアクセス可能とする。

2.2 システムの概要

BOOKHでは、図1に示すように全てのテキストは「INFORMATION」という名前のテキストを最大元とする半順序集合の中に組みこまれる。「INFORMATION」の下には、各種のシステム情報が階層に従う半順序関係によって連っている。例えば、ライブラリに関するテキストの階層、コマンドに関するテキストの階層、教育用テキストの階層等が存在する。

テキストの画面及び運用ソースプログラ

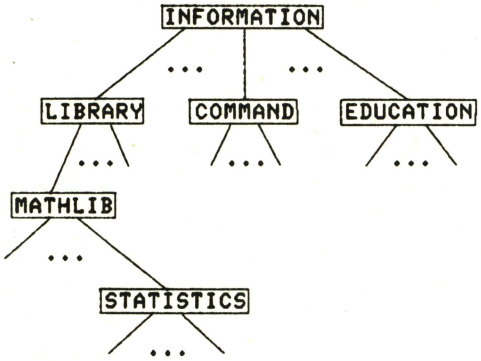

図1 テキストの階層

テキスト編集サブシステム	テキストの種類
BOOKHエディタ	INFORMATION
ライブラリエディタ	LIBRARYとその下位
コマンドエディタ	COMMANDとその下位
BOOKEサブシステム	EDUCATIONとその下位
⋮	⋮

表1 テキストと編集システムの対応

ムの作成・変更を行うために、表1に示すテキストの種類に対応してテキスト編集サブシステムがある。BOOKHエディタ、ライブラリエディタ、コマンドエディタはテキストの内容の仕様記述から、表示用画面とそのBOOKによる運用プログラムを生成する。BOOKEサブシステムは、CAI等教育用教材テキスト作成を行うものである。[1)]

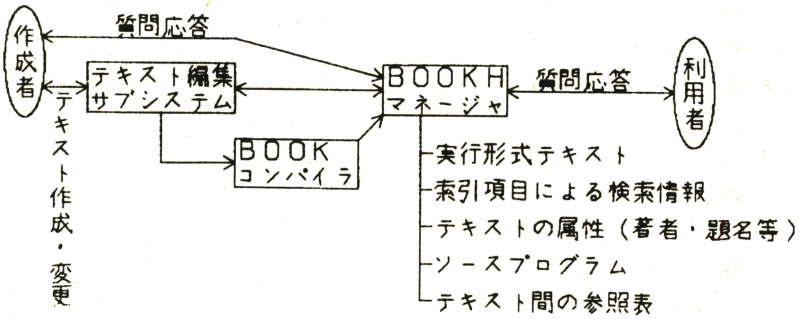

図2　サブシステムの関係

図2にサブシステム間の情報の流れを示す。テキスト編集サブシステムによって作成したBOOKプログラムをBOOKコンパイラに入力することによって、実行形式テキストが生成され、BOOKHマネージャによりテキストの集合への登録、検索情報の生成が行われる。BOOKで記述したソースプログラムもBOOKHマネージャに登録され、作成者が仕様の変更を行う場合にテキスト編集サブシステムに渡され、仕様の変更に応じてテキストの変更が行われる。

利用者は、題名、著者名、キーワード等によってテキストの検索を行うこともできるし、項目索引によって必要な情報に対する質問応答を行うこともできる。

3. 運用記述言語BOOKとその処理系

3.1 特徴

BOOKは、別に作成された画面を編集し運用するための問題向言語であり、次のような特徴を持っている。

(1) 記述単位「page」の導入

テキスト作成者にとっては、利用者に中断されることなく継続して表示したい一連の画面や、利用者との会話を行うための手続きがある。このような一連の運用を記述したプログラムの単位を「page」と名付け、「page label」と呼ばれる名札で区別する。pageの内部は、ブレイクボタンで中断される以外は、作者の意図に従ってプログラムに記述された通りに運用される。pageの終りには必ず利用者の介入を認める入力促進記号が表示され、利用者の意志による操作命令 —— F（次の頁へ）, R（今の頁を反復）, B（直前の頁へ）, C（目次の表示）, J（別の頁へ、頁数も入力する）, E（テキストの実行終了）, I（索引項目表示） —— が許される。pageの導入は、作成者がテキストの論理的組み立てを容易に行うためだけでなく、後述するように、プログラム記述の単位を小さくして、テキストの記述順序と編集順序を独立にし、挿入・削除の容易さ、プログラムの見易さ、記述の簡易さ及び親言語との整合性を目指したものである。

(2) テキストの記述順序、編集順序、実行順序

BOOKで記述したプログラムは宣言部、目次部、運用部から成る。運用部はpageの集まりであるが、page相互間の記述順序は任意で、テキストの編集順序とは無関係である。編集順序は目次部に記述されたpage labelの順番に従う。テ

キストの実行順序は利用者の操作命令で決まる。利用者が最初からpageを進めていけば、テキストは目次通りに実行される。

(3) 階層構造の導入

pageは利用者に情報提供する際の概念の単位である。この論理的組み立てを明確にするため、目次部には編集順序のみでなくpageの階層関係をも記述する。pageの記述部で他のテキストを呼び出した場合は、そのpageは呼び出したテキスト内のpageの上位概念となる。

(4) 親言語形式の採用

画面の運用記述に際して、テキスト作成者が複雑なアルゴリズムを本人の得意とする言語で自由に記述することができるように、既存の汎用言語を親言語とし、その上にテキスト作成に固有なデータ型、操作、制御を記述するための表現を付加して言語BOOKを構成する。BOOKプログラムと親言語に翻訳したプログラムとの間の構造的な整合性を保つためには、page単位で一つの手続きを構成するように翻訳すればよい。

(5) BOOKに固有な文

BOOKには表2に示すような固有のデータ型、データ操作文、制御文がある。

3.2 記述例

図3はFORTRANを親言語とするBOOKの記述例である。BOOKプログラムはBEGINBOOK文で始まり、ENDBOOK文で終る。BEGINBOOK文中のMATHLIBはテキスト名、

	型名	備考
データ型	SCENE	画面名の宣言
	PAGELABEL	pagelabelの宣言
	PAGESET	pagesetの宣言
	STORE	個人別ファイルに値が保存される変数の宣言
	PROGRAM	テキスト中で作成・変更・実行するプログラム名の宣言
	データ操作文	被操作データの型
順序づけ	CONTENTS ENDCONTENTS	例題参照
	JUMP	SCENE又はPAGELABEL又はPAGESET
	INSERT	PAGELABEL又はPAGESET
	ON BREAK	親言語のラベル
画面表示	DISPLAY	SCENE
	ERASE	整数(行), 整数(字)
	HALT	整数(秒)
	HINT	SCENE
一致	MATCH	SCENE
プログラム作成	NEW	PROGRAM
	PUT	PROGRAM, 行番号, 文字列
	GET	PROGRAM, 行番号, 文字列
	DELETE	PROGRAM, 行番号,
	LIST	PROGRAM
	RUN	PROGRAM
結合	SYSTEM	システム名,"パラメタの列"
	BOOK	テキスト名[,著者名]
	BEGINBOOK ENDBOOK	テキスト名[,著者名]

表2 BOOKに固有な文

```
      BEGINBOOK MATHLIB,A.TAKEUCHI
C
C *** DECLARATION PART
      PAGELABEL EX1,EX2,P10
      SCENE P1,P2,P3,E1,P4,E2
      PAGESET PSET1(P3,EX1),PSET2(P4,EX2)
      PROGRAM PROG1
      CHARACTER ANSWER*3
C
C *** CONTENTS PART
      CONTENTS
        1 CHAPTER1 (ﾋﾞｾｷﾌﾞﾝﾎｳﾃｲｼｷ)      P1
          2 SECTION1 (ﾋﾞﾌﾞﾝﾎｳﾃｲｼｷ)      P2
            3 SLRKT  (ﾙﾝｹﾞｸｯﾀﾎｳ)        PSET1
            3 SLHAM1 (ﾊﾐﾝｸﾞﾎｳ)          PSET2
              .
              .
              .
        1 CHAPTER10 (ﾄｳｹｲｹｲｻﾝ)         P10
      ENDCONTENTS
C
C *** MANAGEMENT PART
 EX1: DISPLAY E1
      WRITE(6,1)
    1 FORMAT('0Do you want to execute this program?'/
     &        ' Please answer "YES" or "NO".')
      READ(5,*) ANSWER
      IF(ANSWER .EQ. 'YES') RUN PROG1
 EX2: DISPLAY E2
      .
      .
      .
 P10: BOOK STATISTICS
      ENDBOOK
```

図3 BOOKプログラムの例

A.TAKEUCHIは著者名を表わし、これらの情報は処理系を通してBOOKHマネージャに渡され、検索情報の一部となる。宣言文のPROGRAMは、PROG1という名前のプログラムを作成・修正又は実行することを示す。この例では、EX1と名付けたpage中でPROG1の実行が行われる。目次部の各行は、目次に加えて、左端に目次の階層を表わすためのインデンテーション数を、右端にその目次の運用を記述したpageのpagelabelを書く。一つの見出しに対応するpagelabelが複数個ある場合は、まとめてpageset宣言をする。見出しに対応する画面の運用が単に一画面を表示するだけならば、pagelabelの代りに直接画面名を書く。この場合は運用部は不要となる。例題中ではPSET1とPSET2がpageset、P1, P2は画面名である。pageset宣言中のP3、P4は画面名である。E1、E2はpagelabel EX1, EX2の中で運用される画面名である。P10というpageでは、STATISTICSというテキスト名のテキストを呼び出す。

3.3 BOOKコンパイラ

BOOKコンパイラは表2にあるようなBOOK文を親言語に翻訳するもので、次のような機能を備えている。

(1) 目次によるページの編集機能

ソースプログラムの目次部に記述されたpageset, pagelabel, 画面名に頁数を割り当て、この数字を用いて目次画面を作成し、利用者がこの頁数を指定すると、対応するpageの運用が始まるように目的プログラムを生成する。

(2) 索引項目の自動編集機能

画面作成時に、画面の属性として作られた索引項目を集め、頁数と対応させて辞書順のインデックス画面を作成する。利用者が語句を問い合わせた場合はこのインデックスリストを索引し、該当頁をその時点での利用頁に置き換えて表示し、該当頁の運用が終了した時点でもとの頁に戻るような目的プログラムを作る。これらの索引項目は目次の階層、頁数、下位テキストの有無と共に、BOOKHマネージャに渡される。

(3) 実行順序検査機能

テキストの実行順序は、他のテキストの呼び出しを除けば、目次部と、運用部中のJUMP文・INSERT文で構成される連結グラフによって決まる。このグラフ構造は処理系によってチェックされ、無限ループや孤立点が検出される。

表2のBOOK文による他のテキストの呼び出しはBOOKHマネージャに通知され、テキスト間でのループが検査される。

4. テキスト編集サブシステム

テキスト編集サブシステムは、表示のための画面及び索引項目の作成、BOOKを用いた運用プログラムの記述を行うものであり、BOOKHエディタ、ライブラリエディタ、コマンドエディタ、BOOKEサブシステム等がある。

4.1 特徴

(1) BOOKHを有効にするためには、テキストの半順序関係が概念の階層と対応していなければならない。その意味で、作成者は自分の作るテキストが半順序関係のどこに位置するのが最も適当かを知り、作成したテキストを呼び出すpageを上位概念のテキストの目次部で指定できなければならない。従って、テキス

ト編集サブシステムは一つのテキストの編集機能を持つだけでなく、BOOKHマネージャとの間にテキストの階層を辿るためのインタフェイスを持つ。

(2) BOOK言語は多様なテキストの記述が可能であるように設計されている。しかし、画一的テーマのテキストを作成するのであればBOOKの全ての機能を知る必要はなく、作成するBOOKプログラムも当然画一的な形でよい。ここで紹介するのはこのような問題を扱うテキスト編集サブシステムで、表示内容の仕様から画面と運用プログラムを自動的に作成する。

4.2 BOOKHエディタ

BOOKHエディタは、テキスト集合の最大元である「INFORMATION」という名前のテキストの編集を行う。このテキストの内容は、画面の表示、下位テキストの呼び出しの2種類のpageからできている。BOOKHエディタはBOOKモードと画面モードの2つから成り、次の機能を持っている。

(1) BOOKモードの機能

目次の作成・追加：目次のインデンテーション数、目次、対応するpageの内容（画面表示かテキスト呼び出しか）を指定して、目次の作成を行う。画面表示を指定した場合、エディタは画面モードになり、対応する画面の作成を行う。テキスト呼び出しの場合には、呼び出すテキスト名、そのテキストの編集に用いるサブシステム名を指定する。目次に対応する運用プログラムは自動的に作成される。

目次の変更：目次のインデンテーション数、タイトル、編集順序の変更を行う。

テキストの登録：登録を指定すると、BOOKコンパイラを呼び出し実行形式テキスト、ソースプログラム、検索情報の登録を行う。

他のテキスト編集のためのサブシステム呼び出し：対応するpageが他のテキスト呼び出しである目次の一つを指定し、そのテキストの編集開始を指定すると、あらかじめ登録されていたテキスト編集用サブシステムが呼び出される。

画面モードへのモード変更：画面表示の目次の一つを指定し、対応する画面の変更のために画面モードに変更する。

(2) 画面モードの機能

表示用画面の作成・変更：テキストエディタにより、画面の作成・変更を行う。

索引項目の作成・変更：画面の属性となる索引項目の作成・変更を行う。

4.3 ライブラリエディタ

ライブラリエディタはBOOKHエディタの下位に位置しており、「INFORMATION」から呼び出されるテキスト「LIBRARY」及びその下位にあるテキストの作成・変更を行うもので、プログラムライブラリ説明のためのテキストを扱う。

このエディタで扱うテキストのpageの内容は、画面の表示、下位テキストの呼び出し、プログラムライブラリの説明と例題実行の3種類である。

ライブラリエディタは、BOOKモード、画面モード、仕様入力モード、管理モードの4つのモードを持っている。BOOKモード、画面モードの機能はBOOKHエディタと次の点を除いて同様である。すなわち、BOOKHエディタでは、他のテキスト編集のためにサブシステム呼び出しを行った際に下位のエディタが呼ばれるのに対し、ライブラリエディタでは自分自身が呼ばれる。

(1) 仕様入力モードの機能

このモードは、ライブラリの仕様入力から画面と運用プログラムを生成する。図4は仕様の入力例であり、下線付きは計算機からの出力を示す。sequenceには呼び出し手順を、explanationには説明を入力する。続いて、sequence入力に従ってパラメータが表示されるので、それぞれの型、種類、説明を入力する。conditionは使用条件を示す。parameterの説明をERROR INDICATORとしたものについては、その意味を入力する。以下、注意事項、参照手続き、作成者を入力する。exampleでは、使用例とその説明を与える。使用例の入力は使用言語のサブシステムを呼び出し、そのサブシステムの下でプログラムを作成することにより行う。

sequenceで入力した手続き名、explanationのmethod, object, answerの各語、supplement中の" "で囲んだ語は、この画面に対する索引項目となる。objectにある「:」は、:に続く語が前の語の関連語であることを示しており、索引項目が検索系に組み入れられた際に、:に続く語で検索することも可能にする。なお、BOOKHマネージャにより検索情報を生成する場合、目次部の深さに加えて、各語には次のような深さの相対関係が与えられる。

1 手続き名
　2 methodの語
　　3 objectの語
　　3 answerの語
　2 supplementの語

```
sequence ----CALL SLRKT(FN,X0,Y0,HH,NS,X,Y,IERR)
explanation
  method ----ﾙﾝｹﾞｸｯﾀﾎｳ
  object ----1ｶｲｼﾞｮｳﾋﾞﾌﾞﾝﾎｳﾃｲｼｷ:ｼﾞｮｳﾋﾞﾌﾞﾝﾎｳﾃｲｼｷ
  answer ----ｶｲ
  supplement-y'=fn(x,y)ｦ ｼｮｷｼﾞｮｳｹﾝ yo=y(xo)ﾉ ﾓﾄﾃﾞ ﾄｸ。
parameter
  FN      -FUNCTION
      explanation-y'ﾉ ﾃｲｷﾞ。
  X0      -REAL,INPUT
      explanation-xﾉ ｼｮｷﾁ。
  Y0      -REAL,INPUT,yﾉ ｼｮｷﾁ。
    .
    .
    .
  IERR    -INTEGER,OUTPUT,ERROR INDICATOR
condition -NS>0 , HH<>0
error indicator  IERR
  IERR =0 ,ｴﾗｰ ﾅｼ。
  IERR =-1 ,NS<=0
  IERR
remarks
  necessary routine -FUNCTION FN(X,Y)
  necessary routine -
  necessary declaration -EXTERNAL FN
  necessary declaration -
  others -
referd routine -
author -
example
  explanation-y'=x/(y+1)ｦ ｼｮｷｼﾞｮｳｹﾝ xo=0,yo=0ﾃﾞ ﾄｸ。
              =
  example
    ===ENTER TO FORTRAN SUBSYSTEM===
*AUTO
*010 DIMENSION X(10),Y(10)
      .
      .
      .
*150 END
*160
*DONE
```

図4　ライブラリ仕様の入力例

```
name ------ SLRKT
sequence -- CALL SLRKT(FN,X0,Y0,HH,NS,X,Y,IERR)
explanation ﾙﾝｹﾞｸｯﾀﾎｳﾆﾖﾘ 1ｶｲｼﾞｮｳﾋﾞﾌﾞﾝﾎｳﾃｲｼｷﾉ ｶｲｦ ﾓﾄﾒﾙ。
            y'=fn(x,y)ｦ ｼｮｷｼﾞｮｳｹﾝ yo=y(xo)ﾉ ﾓﾄﾃﾞ ﾄｸ。
parameters
  parameter type        kind    function
  FN        FUNCTION            y'ﾉ ﾃｲｷﾞ。
  X0        REAL        INPUT   Xﾉ ｼｮｷﾁ。
  Y0        REAL        INPUT   Yﾉ ｼｮｷﾁ。
   .
   .
   .
  IERR      INTEGER     OUTPUT ERROR INDICATOR
condition   NS>0 , HH<>0
error indicator   IERR
  IERR =0 ---- ｴﾗｰ ﾅｼ。
  IERR =-1 --- NS<=0
remarks
  necessary routine ---- FUNCTION FN(X,Y)
  necessary declaration  EXTERNAL FN
referd routine --------- NONE
```

図5A　図4から作成された画面

図4の入力から図3に示す運用記述が生成される。pageの運用を行うと、図5A,図5Bの表示が行われる。始めの頁は説明の表示で、それに使用例の表示と実行の頁が続く。使用例を実行するか、という問い合わせに"YES"と答えると、表示されているプログラム（画面作成時に入力したプログラム）がその場で実行される。

(2) 管理モードの機能

sequenceとparameterの部分の入力形式は、ラインプリンタの使用言語によって異なる。従って、これらの入力形式はあらかじめ言語ごとに登録される。この登録を行うのが管理モードである。エディタを使う場合、最初に使用言語の指定を行う。

図4に示したFORTRANの例では、sequenceの入力形式として、

'CALL' name ['('{parameter↑','}')']

が登録されている。sequenceの入力形式は、CALLという文字列で始まり、次に手続き名がくる。[ ]で囲まれた部分は省略可能、{ }は1回以上の繰返し、↑は繰返しからの出口を示す。上記以外の超記号として、択一を示す<…|…|…>がある。

4.4 コマンドエディタ

コマンドエディタはBOOKHエディタの下位にあり、テキスト「INFORMATION」から呼ばれるテキスト「COMMAND」及びその下位のテキストの作成・変更を行うもので、コマンドの使用法のテキストを扱う。

このエディタで扱うテキストのpageの内容は、画面の表示、下位テキストの呼び出し、コマンドの説明の3種類である。

コマンドエディタにはBOOKモード、画面モード、仕様入力モードの3つがあるが、前者2つはラインプリンタエディタと同一である。

(1) 仕様入力モードの機能

コマンドの仕様入力から画面と運用プログラムを生成する。図6は仕様の入力例である。...は繰返しを示す。選択項目内に省略時のデフォルト値があれば" "で囲む。大文字及び記号はその文字自身に意味があり、小文字、カタカナはパラメータである。definitionでは、syntax中のパラメータのうちさらに定義すべきものを、functionにはこのコマンドの説明を入力する。operandではsyntax, definition中の文字列及びパラメータの説明

```
example
  y'=x/(y+1)ヲ ショキジョウケン xo=0,yo=0デ トク。

    010 DIMENSION X(10),Y(10)
          .
          .
          .
    150 END

Do you want to execute this program?
Please answer "YES" or "NO".
=YES

          IERR=  0

          X=    0.1000          Y=    4.9876E-03
          X=    0.2000          Y=    1.9804E-02
          .
          .
          .
```

図5B 図4から作成された画面と運用

```
syntax ------CATA [ファイル|カタログ[*|[,op]...[,**]]]
definition --op::=<<A|C|L>(m-d-y)|A|R|S|n|FIRST/name/>
                =
function ----"ファイルゾクセイ","カタログゾクセイ"ノ ヒョウジ。
           ====
operand
  ファイル          -ヒョウジスルベキ ファイル。
  カタログ          -ヒョウジスルベキ カタログ。
  A                 -
  C                 -
    .
    .
    .
auxiliary
  -A(m-d-y):=yネン dガツ mニチ イコウニ アクセスシタ モノノミ ヒョウジスル。
    .
    .
    .
remarks -パラメータヲ スベテ ショウリャクスレバ UMCヲ シテイシタト ミナス。
        =
```

図6 コマンド仕様の入力例

```
command  CATA
function ファイルゾクセイ,カタログゾクセイノ ヒョウジ。
syntax   CATA [ファイル|カタログ[*|[,op]...[,**]]]
       where
          op::=<<A|C|L>(m-d-y)|A|R|S|n|FIRST/name/>
operand
  ファイル ---- ヒョウジスルベキ ファイル。
  カタログ ---- ヒョウジスルベキ カタログ。
    .
    .
    .
  A(m-d-y)  yネン dガツ mニチ イコウニ アクセスシタ モノノミ ヒョウジスル。
    .
    .
    .
remarks --- パラメータヲ スベテ ショウリャクスレバ UMCヲ シテイシタト ミナス。
```

図7 図6から作成された画面

を、auxiliaryは補足説明をする。コマンド名およびfunction中の" "で囲んだ語は、次のような相対的な階層関係を持った索引項目となる。

1 コマンド名

2 function中の語

図7は図6を用いて作成された画面を運用表示したものである。

5 BOOKHマネージャ

BOOKHマネージャは、BOOKHにおける全ての情報を管理しており、コンパイラモード、テキスト編集モード、利用者モード、作成者モードの4部分から成る。

(1) コンパイラモードの機能

BOOKコンパイラから渡されたテキスト名、BOOKソースプログラム、実行形式テキスト、著者名、題名、索引項目及びその深さ、キーワード、テキストが引用している他のテキストの情報に従ってテキストの登録を行い、検索情報を作成する。同時に、テキスト間の参照関係を調べて、ループを発見し警告を出す。

検索情報の作成・変更はテキスト単位で行われる。従って、他の部分に影響を与えず、テキスト単位の更新が可能なように考慮されている。

テキスト作成者はこのモードの存在を意識する必要はなく、BOOKコンパイラのみがこのモードに対するアクセス権を持つ。

(2) テキスト編集モードの機能

テキスト編集サブシステムによってテキストの作成・変更を行う場合に、コンパイラモードで登録されたソースプログラムとテキストの階層構造に関する情報を受け渡す。

(3) 利用者モードの機能

テキスト検索：テキストの題名、著者名、種類が、著者名あるいは題名の辞書順に表示される。題名、著者名、キーワード、種類の条件を指定して表示を制限することもできる。又、テキストの参照関係の表示も行う。

項目検索：項目を指定すると、検索情報から該当する語をみつけ、題名、著者、該当する頁数が表示される。同時に指定した語の下位の語が表示される。題名、著者名、種類を指定して表示を制限することもできる。

テキストの実行：テキスト検索によって調べたテキストを呼び出して実行したり、項目を指定して該当テキストを呼び出し、該当頁を部分的に実行する。

(4) 作成者モードの機能

利用者モードのテキスト検索、項目検索の表示内容に加えて、テキストの参照回数、作成日、修正記録、索引項目の引用回数等の表示が可能となる。テキストの削除は、このモードで管理者用のパスワードを用いて行われる。

6. おわりに

BOOKHは九州大学のACOS-600上に実現されている。今後、仕様記述の方法、日本語表示や処理の問題等検討すべき点が多いが、記述の形式化とテキストの自動生成はこのような修正・再構成を容易にするという点でも意味があると考える。

参考文献

[1] 大槻，古川，竹内「計算機端末を用いた教育と教材の作成」信学技報ET79-4

本 PDF ファイルは 1980 年発行の「第 21 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 〜 2021 年 3 月 19 日
掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html