

オンライン・マニュアルの効果的な利用

村井純, 大和喜一 (慶應義塾大学理工学部)

1. はじめに

昨今のソフトウェア全般の発達にとよない、ソフトウェア工学の分野ではソフトウェアライフサイクルを支援する各種のシステムの研究が行なわれてきている [1, 2, 3, 4]。ソフトウェアのライフサイクルを支援するシステムを考えると、ライフサイクルを開発と運用/保守の段階に分割してとらえることができる [5]。ここでは、ソフトウェアの運用/保守を支援するシステムとして開発中の MOMS (数理工学科オンラインマニュアルシステム) について報告する。MOMS はソフトウェアの運用/保守を統合的に支援するいくつかのツールと、データベースとして用いるオンラインマニュアルのファイルの集合体であり、利用者のインタ・フェースのすぐれている UNIX [6, 7, 8] 上で実現されている。ソフトウェアのライフサイクルを支援するシステムは、開発から運用/保守までを一貫して支援できることが理想であり、MOMS は、その第一段階を実現している。

2. マニュアルの情報とその利用

2-1 マニュアルの機能

計算機のソフトウェアを利用する際に参照するマニュアルの類は、機能別に分けて次の3種類がある。すなわち、特定のソフトウェアに関する開発動機や理論的背景など広範囲に記述した解説書の類、そのソフトウェアを実際に利用するために必要な情報を記述した使用手引書の類、そして、言語やコマンドの文法に関する詳しい定義を記述した文法書の類、である。実際のマニュアルの中には、これらの3つ以上の機能を兼備したものが少なくない。

解説書の類は、実際には、そのソフトウェアを使っている概念に関する論文的な性格が強く、これを参照しただけでは、そのソフトウェアを利用することはできない。また言語処理系においては、プログラミングには文法書の類が必要不可欠であるが、実際に処理系を動かすためには充分でないことが多い。一方、使用手引書の類には、実際のソフトウェアの利用に必要な情報がはいっている。親切なものであれば、実行例なども記述されていて、そのソフトウェアの実行、運用に関しては十分な情報を得ることができる。そのため、手引書の量が増大することが多く、効率の良い管理をする必要が生じてくる。

したがって、一般にあるソフトウェアを十分に理解するには解説書と文法書が必要であり、プログラミングに関しては文法書、そして、実行、運用に関しては利用手引書が必要であるといえる。

2-2 オンラインマニュアル

マニュアルの形態は、参照する手段という観点から印刷された文書として存在する形態と計算機を使って参照できる形態で存在する形態のものと分類することができる。前者は、計算機を直接使用しなくても、情報を得ることができるが、情報の探索は人間に任されている。後者には、磁気テープ等でマニュアル類を保存するものも含まれるが、特にそれらのデータがオンラインで計算機に接続され、常時参照が可能になっていけば、単にマニュアルとして参照するだけでなく、

いろいろな加工をプログラムで行なうことができる。また、会話型の環境では、このようにして得られた情報をさらに多様に利用できる。ソフトウェアやデータ構造についてのマニュアルの情報からデータと、それを加工するマニュアルの参照プログラムが一体となったシステムとして存在すれば、そのシステム内のソフトウェアを利用すると、即座にそれらに関する情報を得ることができるようになる。このような会話的に扱えるオンラインで計算機に接続されたマニュアルの形態をオンラインマニュアルとよぶ。

2.3 情報の内容と探索

オンラインマニュアルは、計算機を利用したソフトウェアの開発、作成、利用等いろいろな現場で参照する。したがって、それらの持つ情報は、プログラミングやソフトウェアの実行に必要な情報、すなわち文法書の類と使用手引書の類が主となる。オンラインマニュアルは、通常オペレーティングシステムのコマンドによって実行される参照プログラムを使うで参照する。このようなコマンドは、一般に引数として参照したいコマンド名を入力して、マニュアルを出力することができる。また、なんらかの文法（例えば引数を書かないなど）により、参照可能なコマンド名の一覧を出力する機能を備えている。参照コマンドの中には、2つ以上の引数を与えることによって、マニュアルファイルの中の見出しを指定し、情報の一部分だけを参照することが可能なものもある[9]。

3. オンラインマニュアルへの要求

3.1 所持すべき情報

ソフトウェアの開発、運用/保守の環境で必要な情報は全てオンラインマニュアルにして容易に参照できることが望ましい。下記の端末での作業において必要な使用手引書の類に属する情報としては、次のようなものを挙げることができる。

- 1° オペレーティングシステムにおいて利用可能なコマンドに関する手引書。
- 2° 同じく、利用可能なライブラリ、サブルーチンに関する手引書。
- 3° システムマクロ、システムコール等に関する手引書。
- 4° システムで決まっているファイルの形式、ディレクトリの構造等、プログラムの作成に必要なシステムの情報。
- 5° システムが出力するメッセージに関する情報。

言語処理系に関しては、その他文法書に類するものが必要である。含まれていなければならない情報は種類ごとにそれぞれ異なる。1°~3°に示した種類においては、各エントリについて下記の要素が含まれていることが望ましい。

1) 探索キーワード (複数)

エントリを参照するためのキーワード。従来のオンラインマニュアルでは参照するためのキーワードはコマンド名と同じでなければならぬ場合が多い。しかし、一般にコマンドやサブルーチンの名前がその機能や内容をその名で表現しているとは限らない。また、それらの名前の語源的ななりゆきを利用者の持つ感覚と一致しているとは限らない。例えば、ファイルの中身を参照するコマンドから cat [6] という文字列を連想する利用者は特殊であり、自分の持つファイルのファイル名表が見たい利用者に対して CMS [9] の HELP [9] コマンドではオンラインマニュアルを参照してその用を果たすことはできない。参照手段を考察すると、その機能を表わすいくつかの表現を使った探索キーワードが必要であ

る。

ii) 文型

コマンドやサブルーチンなどの文型が、引教やオプシヨンの種類などを簡単に示したものは、そのシステムにある程度習熟した利用者にとってむしろ大まかな手助けとなる。そのためマニュアルの中で最も頻繁に参照されるので、これらの情報を集めて参照カードとして使うことが多い。

iii) 手短かな説明

オンラインマニュアルは、一般に実行可能なコマンド名の一覧表を参照する機能を持つたものが多く、コマンド名の情報だけでは、その内容を推測できないものも少なくない。この要素は、そのような場合のために最少限の文章でその機能を表現する情報である。例えば、"PL/I Compiler", "Linkage Editor", "Erase files" などの一行以内の短かい説明でその内容を的確に表現している。

iv) 詳しい説明

文型に示した引教やオプシオンについて詳しい説明を施したり、閲教の結果についてその表現の意味を詳細に述べる部分である。複雑なものに対しては使用例を付加したり、さらに詳細に説明を加えたりする。手引書として残された説明は、すべてこの項目に含まれている。

v) 責任者 (著者、管理者)

システムプログラムに残された誤りや実際に適さない動作を修正するようなソフトウェアの管理を行なう上で、実際の利用者からの情報を収集することは、システムを使い易く安定したものにするために重要な仕事である。一般の利用者が作成したプログラムのうち、便利なものをすべての利用者に提供することで、システムの持つ能力を高めようとする場合、特に管理を専業とする人員を省くためには、システムプログラムを作成した各人が責任者となって管理するのが望ましい。あるソフトウェアに関して、マニュアルからは得られなかった情報は誰から得ればいいのか、システムの誤りらしきものを発見したら誰に知らせたらいいのか等、そのソフトウェアに責任を持っている利用者に関する情報は、管理責任者が分散している使用環境では利用者としてソフトウェア管理者と区つなぐためにマニュアルの中に存在していなければならない。

vi) 最新更新日

このマニュアルが書きなおされた日付によって利用者は、ソフトウェア仕様の変更が行なわれたことを知ることが出来る。仕様の変更を知らせる方法は、一般に版の番号によって行なわれるが、日付を使うのもその一種と考えることが出来る。

vii) メッセージ等

ソフトウェアがメッセージを出力する場合、その出力回数が多いと、長い文章になるのを避けて簡単な文章を済ませることがある。その場合、意味が正しく伝わらないこともあるので、ソフトウェアが送るメッセージが一覧となつておくことが望ましい。

viii) 参考となる他のマニュアル

このマニュアルに記載された記事を理解するのに必要な情報の中で、他のデータ構造やソフトウェアに関するマニュアルで説明されているものがある場合は、それらの名前を記入し、間接的に参照したり、それから連想を行つて関連情報を整理

することができる。

ix) ソースファイル

ソフトウェアの最新のソースファイルが存在する場所、およびそれらの修正履歴等の情報が、システム内で統一的に管理されていれば、それらをオンラインマニュアルの中の情報として効率的に利用することができる。

3-2 生成方法

前節で述べたような情報は、システム内のソフトウェアの追加変更にとめない、更新されなくては価値がないばかりか、余計な混乱を起こさせる原因となる。このマニュアルの作成に関しては、次の二つの問題がある。

1) 作成しない。

オンラインマニュアルのテキストをテキストエディタで作成することは、ある程度の労力を必要とするので履行されない場合が多い。また、マニュアルは英文で書かれることが多い。これは、我々日本人にとって作成する労力が大きくなる大きな原因のひとつである。

2) 作成された情報に完全でない。

汎用のテキストエディタを用いて、マニュアルが作成されると、定められた書式が守られない場合や、必要な情報の要素が欠落する場合が存在する。これがマニュアルの参照を困難にし、読み誤りの原因ともなる。

これらの問題を解決するには、作成の手助けをする、すぐれた使い勝手の良いマニュアル生成プログラムが必要となる。このプログラムには、次のような機能が要求される。

- 1) 利用者と対話して必要な情報を利用者から引き出す。
- 2) 入力された情報や、システムから得られる情報(例えば更新日時等)を収集してマニュアルが保持しなければならない内容を構成する。
- 3) マニュアルとして定められた書式に入力情報を編集する。
- 4) 既存のマニュアルに対する必要箇所だけの変更ができる(エディティング機能)。

3-3 管理方法

オンラインマニュアルの情報は、高速アクセスが可能な二次記憶装置上にファイルとして存在すれば良いと考えられる。マニュアルのエントリは、新しいソフトウェアがシステム内に導入される都度追加され、ソフトウェアの仕様が変更されると更新される。仕様の変更は、外部で開発されたものよりも、内部で開発されたソフトウェアにおいて、著しいであろう。このような観点から、マニュアルファイルとの管理には、

- 1) 内部で開発されたソフトウェアと、外部のものとは区別する(例えば、階層を持ったファイルシステムにおいては、異なったディレクトリを使用する等)。
- 2) 更新が容易なように、各エントリ単位での追加、変更が可能な型式にする(例えば、各エントリに関して、1つのファイルを使用する)。

3-4 参照方法

マニュアルファイルの参照は、従来のオンラインマニュアルのコマンドで用いられているコマンドやルーチンの名を用いて参照する形式の他に、次のような方法を用いて参照することができるが望ましい。

1) キーワードによる探索

前述のように、オンラインマニュアルの参照が、そのコマンド名やルーチンの名前だけでは参照できないとすると、利用者が必要とする情報を参照するためには、かなりの訓練が行なわれ、かついろいろな知識を駆使することを要求されることになる。そこで、指定された項目名が存在しない場合には、その名前をキーワードとして所有する他のエントリも参照することができれば、意味による情報の探索が、ある程度可能となる。

2) 項目別の探索

ひとつのマニュアルの中の特定の項目について参照が行なわれるシステムは少くない[4, 5]。利用者にとって必要な項目は、大半は限られているので、この機能が重要となる。

4. 利用環境の作成

4-1 UNIXのオンラインマニュアル

文書の管理が比較的行き届いたシステムのひとつにUNIXがある。UNIXには、全ての関連文書[6, 7, 8]が、システム内にファイルの形で保持されている。MOMSでは、既存のソフトウェアに属しなすこれらのマニュアルの情報を、最大限活用している。そこで、はじめに、UNIXのオンラインマニュアルの情報と形態、並びに参照コマンドmanについて簡単に紹介しておく。

4-1-1 情報の範囲

システムが保持するマニュアル類の情報には、UNIXの主要ソフトウェアの解説書[7, 8]と全使用手引書[6]がすべて含まれている。使用手引書は、利用者コマンド、管理コマンド、サブルーチン、システムコール、特別なファイル(入出力装置等のソフトウェア的な扱い)、マクロファイル構造、ゲーム類に分類される。manコマンドで参照するのは使用手引書のみであるが、解説書の必要に応じて文書処理を行なってみることが出来る。

4-1-2 情報の内容

各エントリは次のような情報内容を持っている。

- ・コマンド名、及び短い説明
- ・文型
- ・詳しい説明
- ・参照
- ・制限事項
- ・メッセージ
- ・関連ファイル名

4-1-3 マニュアルファイルの管理

UNIXの階層的なディレクトリ構造を利用して、図4-1のように管理されている。

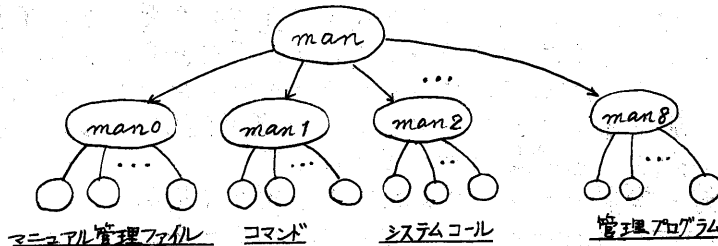


図4-1 マニュアルファイル群の構造

man 1~8のディレクトリ内のファイルは、それぞれのセクションのエントリー一つ一つに対応している。マニュアルの各ファイルは最初、文書清書システムtroff, nroff [8]のテキストの形式で記述されているが、書式化されたnroffの出力と交換しておけば、より迅速に参照することが可能である。

4-1-4 参照コマンド

参照コマンドman [6]は、UNIXのコマンド言語Shell [10]で記述されている。manはエントリの名を引数に与えるとそのマニュアルをA4判に整えて出力をするコマンドで、処理にはnroff, troffを使っている。A4判はCRTにおいては、はなはだ見にくい面もあるが、UCBで開発されたVAX-11版 [11]では、1画面ごとに停止する機能 (more [11]) を利用してこの問題を解決している。

4-2 オンラインマニュアル管理システム-MOMS

MOMSは、マニュアルに関するデータベースとデータベース内の情報の生成探索参照のユーティリティ群とからなるシステムである。ユーティリティ群はデータベースの情報の更新、生成を行なうものと探索、参照を行なうものに大別される。また、実験システムとして、このデータベース内の情報を利用したCAI的な機能を備えたコマンドインタプリタmanshを試作してみた。MOMSの構成と情報の流れを図4-2に示す。

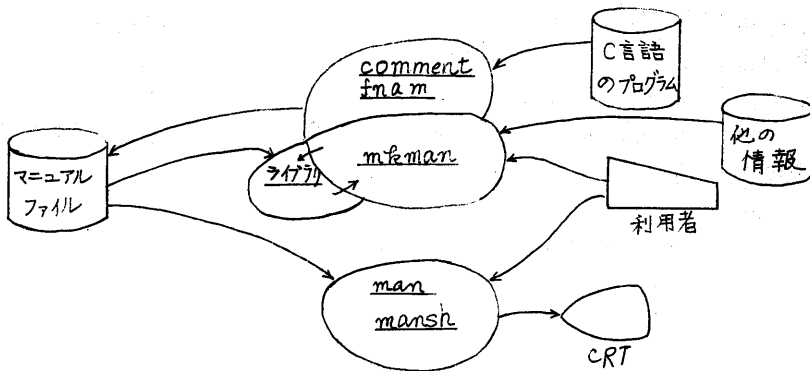


図4-2 MOMSの構成と情報の流れ

4-2-1 データベース

データベースとしては、他のUNIXシステムとの通信を考慮して、本来のUNIXのマニュアルファイルとの互換性を保持している。すなわち、UNIXの各々のセクションのディレクトリに加えて、対応した各々のディレクトリに内部で開発されたソフトウェアに関するマニュアルファイルが保持されている。それぞれのファイルの中に取り込まれている情報は次の様になっている。

1) エントリの名前と探索キーワードを兼ねた手短かな説明

例) mkman-create, modify on line manual files

エントリの名前とそのソフトウェアに関する手短かな説明で構成される一行以

内の文で、説明の部分は、探索の際のキーワードとして利用される。
ii) ~ ix) はそれぞれ 3-1 で述べた項目が採用されている。

名前及び手短かな説明
文型
詳しい説明
関連ファイル 関連文書, コマンド メッセージ 出
ソーステキストの場所と名前
責任者・管理名
最新更新年月日

4-2-2 マニュアル生成のためのユーティリティ

現在MOMSは、ソースプログラムからソフトウェアの情報を自動的に抽出することは行なっていない。したがって、マニュアルファイルは、mkmanによって対話的に作成される。しかし、Cのプログラムに関して、その関数名とファイル名の対応状況を抽出するユーティリティのfnamやコメントを抽出するcomment等のユーティリティを作成し本来のクロスリファレンス的な利用法以外に、そのソフトウェアの内容を表現するデータの一種として利用しようとしている。mkmanは会話形式にマニュアルファイルに必要な情報の入力を要求していくシステムで、既存のファイルに関する編集を行なうことができる。mkmanの実行例を例1に示す。

4-2-3 マニュアル参照ユーティリティ

マニュアルの参照を容易にするために、コマンドのmanとサブルーチンのman(), search() を作成した。

manコマンドは本来のUNIXの同名のものとは異なり、1) マニュアルの参照、2) Keywordによるエントリの探索、3) 項目を指定した参照等の機能を持っている。

search()はパラメータとして与えられた名前から、その名前を持つエントリのファイル名を探すサブルーチンで、man()はパラメータとして与えられたファイル名の中の指定された項目の内容を得るためのサブルーチンである。

4-2-4 マニュアル情報を有効利用するコマンドインタフェース

manshはマニュアルファイル内の情報、特に探索キー、手短かな説明、文型等を利用したCAIとしての機能を持ったコマンドインタフェースである。manshの概念は次のようになっている。

- 1) 入力された行が正規のコマンドであればそれを実行する。
- 2) 入力されたコマンドがコマンドとして認識できない場合は、そのコマンドをキーワードとして発見されるコマンド名と手短かな説明を選択メニューとして表示する。
- 3) 探索用文字(例では '?') で終わる行が入力されるごとに表示されるマニ

アルの情報のレベルが1つ上がる。

マニュアルの情報のレベルは次の様に定めてある。

レベル1: 名前と手短かな説明

レベル2: 文型

レベル3: マニュアル内の全情報

このコマンドインタフェースは、UNIXの入門者には特に有効である。manshの動作例を例示に示す

5. 結論

オンラインマニュアルの実社会での応用においては、ここで指摘した点の他に、日本語によるデータベースの導入を考慮することが、重要な鍵の一つである。MOMISにおいては、データベースとしてのマニュアルファイルに、UNIXのマニュアルファイルとの互換性を考慮してあるので、その面からの制限を受けるが、実際の運用面、特に、マニュアルファイルの生成、参照また、内部検索、変更を行なっているリフトウェアの管理には成果をあげている。現在進行しているUNIXによる学内ローカルネットワークのプロジェクトの中では、特にリフトウェアの分配、およびそれらの運用に関するいろいろな要求に効果が上がるものと期待している。

また、今後の課題としては、ソフトウェアの開発を支援するシステムの研究と、その段階で生成工学データベースの情報から、運用/保守を支援するMOMISのようなシステムのデータベースの情報への情報の利用や伝達についての研究を行なっていく必要がある。

参考文献

- [1] 国井利泰監修, ソフトウェア工学, bit増刊号 1978年8月.
- [2] 「ソフトウェアツール-1」, 情報処理, vol.20, No.6, 1979.
- [3] 「ソフトウェアツール-2」, 情報処理, vol.20, No.8, 1979.
- [4] 国井利泰監修, ソフトウェアプロジェクト工学, bit増刊号, 1981年4月.
- [5] 石田健市, 「ソフトウェアライフサイクル」, ソフトウェアプロジェクト工学, 国井利泰監修, bit増刊号, 1981年4月.
- [6] "UNIX programmer's manual Seventh Edition, vol. 1" Bell Laboratories, Jan. 1979.
- [7] "UNIX programmer's manual Seventh Edition, vol. 2A" Bell Laboratories, Jan. 1979.
- [8] "UNIX programmer's manual Seventh Edition, vol. 2B" Bell Laboratories, Jan. 1979.
- [9] "VAX-11/VMS User's guide", Digital Equipment Corporation, 1980.
- [10] "IBM Virtual Machine Facility: CMS User's guide", IBM 1977.
- [11] J.F. Ossanna, "NROFF/TROFF User's Manual", Bell Laboratories, 1976.
- [12] S.R. Bourne, "An Introduction to the UNIX Shell", Bell Laboratories, 1978.

[13] "UNIX (4BSD) Manual," University of California, Berkeley,
1980.

```
$ mkman
Welcome to the Conversational Manual-Making System
Command name ? mkman
Section ? 1
Brief note of function. ? make a manual file
Usage ? mkman
Description
Just answer some questions, then you can make manual file.

+
Files
/usr/man/manX/xxx.X

+
See also
man(1),mansh(1)

+
Diagnostics

+
Bugs

+
Source Files
mkman.c on "UNIX source-1" Rk-05 disk pack.

+
Responsibilities? Jun Murai
Thank you!!
$ man mkman
```

MKMAN(1) UNIX Programmer's Manual MKMAN(1)

```
NAME
    mkman - make a manual file

SYNOPSIS
    mkman

DESCRIPTION
    Just answer some questions, then you can make manual file.

FILES
    /usr/man/manX/xxx.X

SEE ALSO
    man(1),mansh(1)

SOURCE FILES
    mkman.c on "UNIX source-1" Rk-05 disk pack.

RESPONSIBILITY
    Jun Murai

DATE
    07/03/81
```

例 1 mkman の実行例と結果 (下線が入力)

```

$ date
Fri Jul 3 19:45:38 JST 1981
$ list
1: look - find lines in a sorted list
2: ls - list contents of directory
3: nm - print name list
4: lc - list contents of directory
> 4
$ lc ?
lc [-cn] [dir ...]
$ lc /usr/games

files:                [26]
a.out                 chess                lib                ttt
arithmetic            chins.d              maze              ttt.k
backgammon            cubic                 moo               words
banner                fish                  eet               words1
bcd                   fortune              quiz              wump
bj                    handman              quiz.k
checkers              handman.old          reversi

$ compile
1: cc - C compiler
2: gcc - portable C compiler
3: f77 - Fortran 77 compiler
4: wacc - set another compiler-compiler
>

$ calendar?
1: cal - print calendar
2: calendar - reminder service
> 1
$ cal ?
cal [ month ] year
$ cal ?
NAME
cal - print calendar
SYNOPSIS
cal [ month ] year
DESCRIPTION
Cal prints a calendar for the specified year. If a month is
also specified, a calendar just for that month is printed.
Year can be between 1 and 9999. The month is a number
between 1 and 12. The calendar produced is that for England
and her colonies.
Try September 1752.
BUGS
The year is always considered to start in January even
though this is historically naive.
Beware that 'cal 78' refers to the early Christian era, not
the 20th century.
$ cal 6 1981
June 1981
S M Tu W Th F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
$

```

例2 mansh の実行例 (下線が入力)