

データベース機能を持つ マニュアル作成支援システム

佐藤 誠一郎
(株)リコー ソフトウェア事業部

本稿では、マニュアルのようにある決まった論理構造を持つ文書をデータベース化するためのモデルについて述べ、実際にインプリメントしたシステム CAMMS について報告する。マニュアルのような文書は再版性が高いにもかかわらず実際にマニュアルを作成する現場では過去に作られた資源を生かしきっておらず、それらのデータをデータベース化したいというニーズは大きい。本システムでは、マニュアルをいくつかの階層に分け、それぞれを関連付けることにより文書の構造をそのままの形でデータベース化することができる。また、マニュアルだけでなく、構造化できる文書であればデータベース化し、部品化、再利用をはかることが可能になる。

A Manual Maintenance System with Database Functions

Seiichiro Sato
Ricoh Software Division
1-1-17, Koishikawa, Bukyo-ku Tokyo 112, Japan

This paper describes a database model of a document, such as a user's manual, which has a fixed structure, and a design concept of a manual maintenance system based on a database model. A document can often be reprinted, but also can be rewritten without using past passages found in the previous manuals.

In our system the data model was divided into 4 classes related to each other by physical links. Using this method, we were able to create a fixed structured database system, and to reprint easily manual passages.

1 はじめに

近年、高性能のワークステーションの普及、CD-ROM、光磁気ディスク(MO)などの低価格大容量ディスクの出現により文書の電子化が盛んに行なわれている。そのような状況のもと、実際の文書作成者と印刷会社の間でフロッピー、MO、ネットワークなどを用いた「紙」を介さない文書のやりとりが頻繁に行なわれるようになってきた。また、消費の多様化に合わせ、各メーカーで作られる製品が最少多品種になり、それらの製品に付属するマニュアル(取扱説明書)も当然多種類のものが出版されている。しかし、それらのマニュアルも7~8割は過去のデータを利用することができるため、過去の資産を再利用できる形で取っておきたいという要求が大きくなってきている[1]。

長い間「鉛の活字」によって組版と印刷を行っていた印刷業界も電子組版が普及し、データをフロッピーなどのメディアに保存して、再利用をはかるようになってきている。しかし、フロッピーのような物理的なメディアにデータをただ保存しているだけでは再利用を効率良く行うことができない。そこで、過去に作られた文書を部品化し、それをデータベースで管理することによって再利用したいという要求がおこってきた。

本稿ではまず現状の問題点をあげ、それらの問題点を解決するために考えたマニュアルの構造を表現するための文書モデルについて述べる。また、複数の文書から参照される共通データを処理のしやすいようにモデル化する方法についてもここで述べる。最後に、このデータベースを用い実際に新しいマニュアルの作成支援を行うシステムを、UNIXのXウィンドー上で実現したので、それについて簡単に報告する。

2 現状の問題点

マニュアルなどの再版性の高い文書を作成する現場では以下のような点が大きな問題となっている。

1. 過去のデータの再利用が困難

フロッピーにはファイルの形でデータが入っている。そこから目的のものを探し出す手間と、取り出して利用する手間がかかる。しかも複数のフロッピーに入っているデータを組み合わせるため、結局、出力して切り貼り

することになってしまう。

2. 共通なデータが複数存在する

フロッピーを介したデータ管理ではフロッピーの所有者のみしかデータを利用できないため、同じ内容のデータが複数のフロッピーに存在し、ムダが多い。

これらの問題点を解決する試みとして、国際規格[2]であるSGML(Standard Generalized Markup Language)[3][4][5]というマークアップ言語をもとにした全文データベースが提案されている[6]しかし、マークアップされた文書のタグをもとに全文検索を行うため、データの部品化がはかられているとはいえ、目的のデータを取り出して再利用しにくく、共通なデータが複数存在する。

3 構造を持つデータベース

マニュアルなどのように、ある決まった論理構造をもつ文書を効率良く管理するためには、データベースにもそれに合った構造を持たせる必要がある。また、構造が決まっている文書であれば柔軟に対応できるような拡張性の高いものでなくてはならない。我々はこれらの問題を、文書構造を4つの階層(クラス)に分割し、それぞれのクラスのインスタンス間の関連をデータベース中に保持することによって解決した。

この章ではまず、文書の構造を製品のユーザーズマニュアルを例にして説明し、構造を持つ文書を表現するための文書モデルについて述べる。

3.1 マニュアルの論理構造

例えば、ある製品のユーザーズマニュアルは図1のような構成で書かれている。

このような分け方をしたときに「目次」、「見出」、「索引」などの項目を区分と呼ぶ。例えばこのマニュアルは9つの区分から構成されているとみなすことができる。区分の集合はジョブと呼ばれ、マニュアルは1つのジョブである。また、それぞれの区分は、「タイトル」やいくつかの「文」、「写真」、「線画」などから構成されており、これら、「タイトル」、「文」、「写真」などの項目を種別と呼ぶ。「タイトル」、「文」はテキストデータであり、「写真」、「線画」はイメージデータである。このテキスト、イメージをここでは実体と呼ぶことにする。

このようにマニュアル(ジョブ)は複数の区分から構成されており、区分は複数の種別から構成されている。また、種別はテキストかイメージのどちらかになる。つまり、マニュアルは見方によって4つの階層に分類することができる。これは、おのおののレベルに含まれる内容は異なるが、構造を持つ他の文書にも適用することが可能である。



図 1: マニュアル全体の構成

3.2 文書モデル

前節で構造化された文書は4つの階層に分類できると述べた。その点を考慮にいれて、データベース化するための文書モデルを考える。

各階層をクラスと呼び、4つのクラスをそれぞれジョブクラス、区分クラス、種別クラス、実体クラスと呼ぶ。文書を、ジョブクラス、区分クラス、種別クラス、実体クラスのインスタンスオブジェクトから生成されると考えるのである。ジョブクラスのインスタンスオブジェクトにはそのジョブの名前(マニュアルのタイトル)、作成日、使用言語などが記述され、そのインスタンスオブ

ジェクトは他のクラスのインスタンスオブジェクトと作られる木構造のルートになっている。区分クラスのインスタンスオブジェクトにはその区分の種類が記述される。種別クラスのインスタンスオブジェクトにはその種別の種類の他、その種別が見出である場合はその見出の深さが記述される。最後の実体クラスのインスタンスオブジェクトにはその文書の中に書かれている実際のテキストデータとイメージデータが記述される。

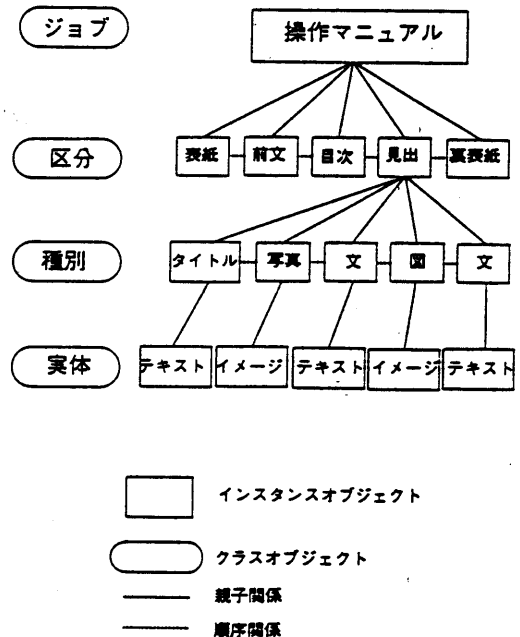


図 2: インスタンスオブジェクト間の関係

木構造をなす各インスタンスオブジェクト間の関係は図2のようになる。あるクラスのインスタンスオブジェクトとサブクラスのインスタンスオブジェクトの間には親子関係が存在する。また、同じジョブクラスのインスタンスオブジェクトを親に持つ区分クラスのインスタンスオブジェクトの間、同じ区分クラスのインスタンスオブジェクトを親に持つ種別クラスのインスタンスオブジェクトの間にはそれぞれ順序関係が存在する。ここでいう順序関係とは、文書の構成上の順序、という意味である。例えば、図1のマニュアルの場合、こ

のジョブの区分クラスに属するインスタンスオブジェクトは、表紙→前文→目次→見出(第1章)→見出(第2章)→見出(第3章)→索引→後文→裏表紙という順序関係を持つ。

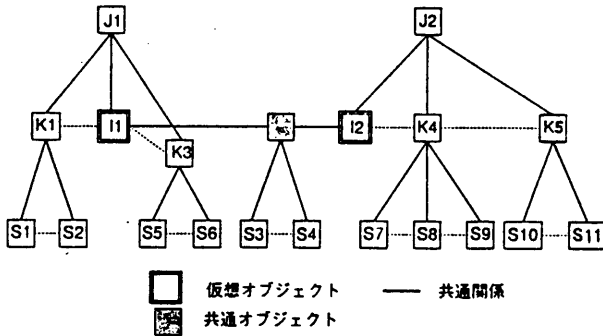


図3: 共通クラス

3.3 共通クラス

マニュアルは7~8割は過去のデータを利用することができる。つまり、データ量が増えれば増えるほど、複数のジョブの間で共通であるデータの量も増えることになる。

ここで、「共通なデータは実体が複数存在しない」という前提を考えると、1つの共通のインスタンスオブジェクトと親子関係、順序関係を持つインスタンスオブジェクトが多数存在することになり、構造が複雑になる。また、あるインスタンスオブジェクトがどのデータと共通であるかを探るとき、最悪の場合同じクラスの全インスタンスオブジェクトを探さなくてはならなくなるので、データ量が増えれば増えるほど処理に時間がかかることになる。

そこで、前の節で述べた、4つのクラスの他に、共通データのためのクラスを考えることにする。このクラスをここでは共通クラスと呼ぶ。そして、この共通クラスのインスタンスオブジェクトをここでは共通オブジェクトと呼ぶ。また、共通オブジェクトと共通関係を持つインスタンスオブジェクトをここでは特別に仮想オブジェクトと呼ぶ。仮想オブジェクトは区分クラス、あるいは種別クラスのオブジェクトであるが他と違い、親と

順序に関する情報のみを保持するオブジェクトである。(図3)

共通クラスの導入により前述のように共通データを探す場合でも、全オブジェクトを見に行く必要がなく、共通クラスに属するデータのみを探せば良いことになるので操作の効率が非常に良くなる。そして、仮想オブジェクトは、各ジョブごとの処理を見かけ上共通データがない時と同様に扱うことを可能にする。また、共通クラスは他のクラスと独立して存在しているので、文書がもっと複雑になり階層構造が深くなっても、柔軟に対応することができる。

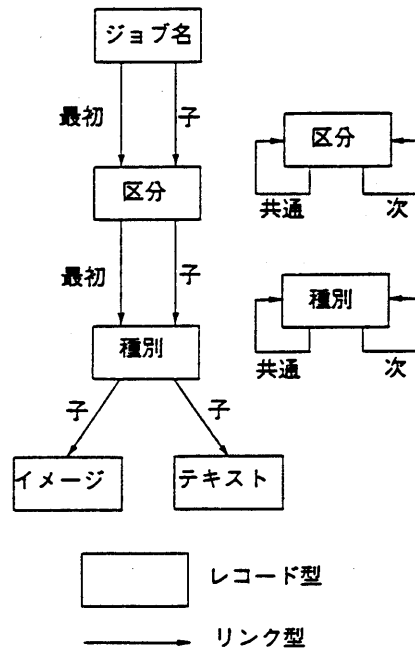


図4: データベースの構造

4 マニュアル作成支援システム

前章で構造を持つ文書を表示するための文書モデルと、共通データの扱いについて述べた。今回我々は、その文書モデルをもとに設計したデータベースを使い、粗原

稿入力から組版用データの作成までをサポートするシステム CAMMS(Computer Applied Manual Maintenance System)を作成した。この章ではこのマニュアル作成支援システム CAMMS について述べる。

4.1 データベース

前の章で述べた文書モデルにもとづいて、データベースを設計する。今回我々は(株)リコーの G-BASE[7]を DBMS として採用した。G-BASE は一般のリレーショナルデータベース管理システムに加え、情報の関連を『リンク型』表現することを可能にした拡張リレーショナルデータベースである[8]。各クラスを、おのおの1つのレコード型に対応させ、そのクラスのインスタンスオブジェクトに記述されている内容を各レコード型のフィールドにする。そして、各インスタンスオブジェクトの関係(親子関係、兄弟関係)G-BASE のリンク型に対応させる。データベースの構造を図4に示す。

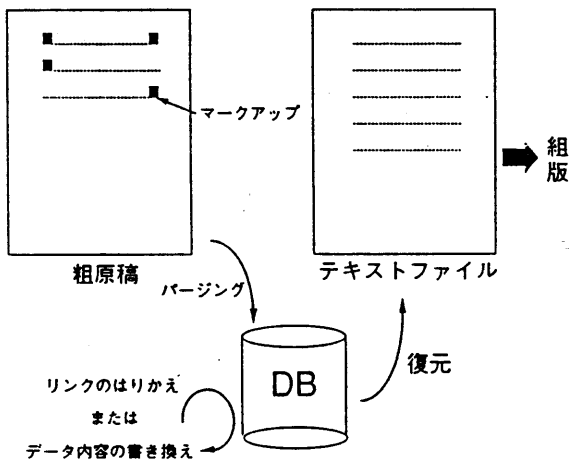


図 5: 処理の流れ

4.2 システムの処理の流れ

CAMMS システムは、マークアップされた粗原稿から原稿データの構文解析を行い、データベースに格納する。そして、そのデータベースから取り出したデータをリンク付けや修正を行って外部の組版システムで扱

るようなデータ構造に変換し、出力する。また、過去に部品化されたデータを組み合わせて新しいマニュアルを作成することも可能である。

CAMMS の処理の流れを図5に示す。

4.3 システムの特徴

CAMMS システムの特徴として以下の3点があげられる。

1. 高い操作性

データなどの入力以外はすべてマウス操作で行えるため、コンピューターに不慣れなエンドユーザーでも容易に使いこなせることができる。

2. 高度な組版、編集

文書はその論理的な構造の他に、意味的な構造を持っている。例えば、ある写真データとその説明は「写真に対する説明をしている文章」という意味的な関係を持っている。そこで、データベース中でテキストデータとイメージデータを関連付けることによりこの意味的な関係を表現する。これにより、組版時にテキストデータからそれと関連したイメージデータを容易に取り出すことができ、組版の効率が上がる。また、印刷機械貿易(株)の組版、編集システムである Computex3000[9] のデータ形式にファイルを変換できるので、ユーザーは CAMMS のデータを使って組版、編集を行うことができる。

3. 独自のマークアップ

データベースが構造を持っているので、文書作成者は新しく文書を作成する時点で、その部分がデータベースのどこに挿入されるのかを記述(マークアップ)する必要がある。本システムでは前に述べた文書の論理構造に合わせ、SGML の考え方を取り入れ、独自のマークアップ規則を定めた。原稿作成者は粗原稿にマークアップすることによって、文書の論理構造を記述できる。

4.4 処理例

原稿作成から、組版処理用ファイルの作成までの処理の流れを簡単に説明する。新しいマニュアルを作成する際、方法としては、(1)マークアップされた粗原稿から作成する、(2)過去の資産を利用する、の2通りが考えられる。ここでは過去のデータを活用して新しいマニュアルを作成する方法について例をあげて説明する。

CAMMS システムはいくつかの画面から構成されており、ここではその中のジョブ表示画面と、マージ画面を使って複数のジョブから新しい1つのジョブを生成する例を示す。ユーザーはまずジョブ表示画面で組み合わせた複数のジョブを選択する(図6)。そこで、原稿作成支援メニューの目次編集を選択するとマージ画面(図7)が表示される。そこでユーザーは右側に表示されている複数のジョブの見出から希望の部分を選択し、右側の画面にコピーする。それによって新しいジョブの基本部分(仮ジョブ)が作成され(図8)、この仮ジョブを他の画面で編集することによって新しいジョブが出来上がる。

このコピー操作によって共通データが定義される。つまり、画面上は複数に見えるデータもデータベース中には一つしか存在せず、しかも、それが共通であるという情報をデータベースが持っている。

そのように作られたジョブデータを組版処理用のデータに変更し、外部組版システムで組版処理を行う。

5 おわりに

マニュアルのように決まった論理構造を持つ文書をデータベース化するために4つのクラスに分かれた文書モデルを提案し、そのモデルにもとづいたデータベースを設計した。そして、そのデータベースを持つマニュアル作成支援システム CAMMS を紹介した。この文書モデルはマニュアルなどの構造を分かりやすくモデル化し、仮想オブジェクトの導入によって複数のジョブの間の共通データも容易に表現することができる。また、X上に作成されたマニュアル作成支援システム CAMMS はユーザーにとって使いやすいインタフェースを備え、マニュアル作成時に過去の資産を有効に、かつ、効率良く利用することを可能にした。

現在我々は、組版システムをも含んだ統合的マニュアル作成システムを開発中である。これにより原稿作成者と印刷担当者が共通のデータベースを利用するため、最新のデータを管理し、利用することが可能になる。そのためにはレイアウトに関する情報をいかにモデル化し、データベース化するかが問題になる。

今後、これらの問題を解決し、将来的にはユーザーによる構造定義を可能にし、通常の文書も扱えるように汎

用化するつもりである。

謝辞

本論文で紹介したマニュアル作成支援システム CAMMS は印刷機械貿易株式会社(以下 PMT と略す)の委託により(株)リコーで開発されたシステムで、PMT の組版・編集システムである Compotex3000 のデータ管理に G-BASE を用いたものである。

開発にあたり、ご指導していただいた PMT の小沢氏、ならびに討論を通して有益なアドバイスをいただいたリコーソフトウェア事業部の鳥谷、宮本の両氏に感謝いたします。

参考文献

- [1] 松野：“マニュアル作成の現場”，ザ・テクニカルライター，19，1989.7
- [2] ISO 8878-1986, Information Processing - text and Office System - Standard Generalized Markup Language(SGML), 1986.10.15
- [3] M. Bryan：“SGML An Author's Guide”，1988
- [4] 田中：“文書の再利用を即すマークアップ言語 SGML”，日経コンピュータグラフィックス，1988.8
- [5] 山崎：“ドキュメント構造記述言語 SGML と電子出版”，日経コンピュータ別冊，1988.7
- [6] 芝野：“SGML と全文データベース”，情報処理学会情報学基礎研究会資料，1989.7
- [7] (株)リコー編：“G-BASE システムマニュアル”
- [8] 長：“グラフデータモデルを実現した G-BASE”，情報処理学会第 35 回全国大会 7Cc-4, 1987
- [9] 印刷機械貿易(株)編：“Compotex 組版マニュアル”

Camm 1			
JOB 操作 履歴作成支援 データ操作 プリント ヘルプ			
ID	JOB名	仕向け地	日付
1	G-BASEレファレンスマニュアル	日本	1991/02/14
2	Compaex3000用マニュアル	イギリス	1991/03/14
3	Compaex3000エキスパートマニュアル	ドイツ	1991/04/12
4	Cammライブラリマニュアル	日本	1991/07/13
5	マニュアルサンプル1	日本	1990/05/28
6	マニュアルサンプル2	アメリカ	1990/05/28
7	マニュアルサンプル3	フランス	1990/05/28

JOB 7

JOBの表示

図 6: ジョブ表示画面

Camm 1	
JOB 操作 履歴作成支援 データ操作 プリント ヘルプ	
G-BASEレファレンスマニュアル	
日本	1991/02/14
よく見る章を作る (H)	
基本はまっすぐ作ることです。(H)	
カーブの量がり方はどうですか。(H)	
《ヒント》 (H)	
スピードアップのために (H)	
日常のメンテナンス (H)	
実行後の手入れ (H)	
コントロールメカの手入れ (H)	
◎ジョーンの上ごれ (H)	
◎ボテのいたみ (H)	

マージの履歴

図 7: マージ画面

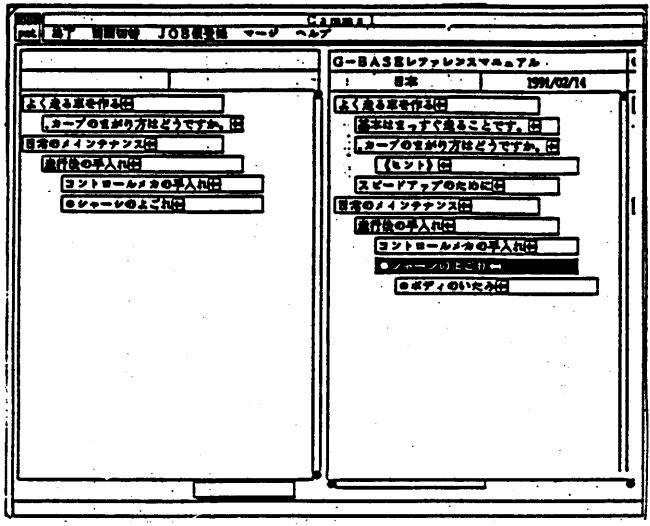


図 8: 新規ジョブの作成