

易しくない データベースのお話し

科学書刊株式会社:電子版の原稿

「橋梁&都市 PROJECT: 2011」(ISSN 1344 - 7084)

島田 静雄

この冊子は、雑誌「橋梁&都市 PROJECT」に連載することを予定して作成した MS-Word 版の原稿から、PDF (Portable Document Format 形式に変換したもののコピーです。2010 年度から、出版関係は電子出版を模索した努力をする時代に入りました。科学書刊株式会社は、この動きに対応するため、ハードコピーとしての「橋梁&都市 PROJECT」の発行を休刊とし、電子化にどのように対応するかの研究を始めました。実を言うと、この傾向は 10 年前から予測されていました。筆者は、この先取りとして、三種類の発表形式を試してきました

一つ目は、雑誌の記事としての形式です。「橋梁&都市 PROJECT」のスタイルは B5 版二段組みです。こちらの方は、しばらく休刊になりました。この原稿は、MS-Word を使い A4 版一段組みで作成してきましたが、そのまま体裁のよいレポート形式になるように注意して編集してあります。この形式そのままにしたのが、この PDF 版です。

二つ目、この PDF 版をインターネットで公開することです。すこしページ数が多くなりますが、ユーザは、これをダウンロードして印刷して見ることができます。プリンタをお持ちでなければ、原稿ファイルを USB (universal serial bus) 接続のメモリに記録して持ち込めば、簡易製本までサービスしてくれる街中の印刷屋さんが見つかりようになりました。PDF 版の WEB サイトは、差し当たり下記にしてあります。

http://www.nakanihon.co.jp/gi_jyutsu/Shimada/shimadatop.html

三つ目は、パソコンの画面でランダムに項目がアクセスするようにリンクを張った WEB (World WideWeb) 版です。この利用方法を考えて、筆者の原稿は、約 600 字程度のパラグラフ (段落) 単位に分けてあって、インターネットでのアクセス速度が速くなるように、一つのパラグラフがパソコンの一画面に入るようにした HTML (Hyper Text Markup Language) 版です。目次と索引とを参照すれば、かなり便利な検索が使えます。WEB サイトは上の PDF 版と同じ個所です。この冊子は、表紙と目次を含めて全 43 ページあります。電子出版を考えると、ページ番号で項目位置を探すことが実用的ではありません。したがって、目次と索引は、章・節・項のパラグラフ番号で検索するように使って下さい。

(この冊子は、全 **43** ページあります)

目 次

0. はじめに

1. データベース構築までの背景

1.1 軍事研究に始まったコンピュータの開発

弾道計算の高速化が主目的であったこと：コンピュータを事務処理に使うこと：資料と情報の用語の意義：データベースは軍事用語であったこと：データベース利用の大衆化へ

1.2 整理と保存の現実

整理法を教える組織的教育がない：整理法は技術であること：公私の区別をすること：開架式と閉架式の管理方式：二次資料の作成作業が大変：分類番号の代わりにキーワードを使う

1.3 古典的な整理方法

古典的な書物保存の背景：雑誌はまとめて製本する：受動的にカードを使う私的な整理法：積極的にカードを使う私的な整理法：パンチカードを使ったデータ管理：カードを使う管理も捨てたものではないこと：保存用のハードウェア

2. 文字処理と英語用語の理解

2.1 JIS 漢字コード系制定までの経緯

文字処理は基礎技術であること：オフィスオートメーション：難産であった JIS 漢字コード系の制定

2.2 新聞業界での苦闘

鉛の活字を使う印刷術の機械化から：写真製版の利用：文字処理の変革期の話

2.3 英語用語の使われ方

用語の意味と使い方に注意が必要であること

2.4 システムズエンジニアリングの理解

システムとは何か：オペレーションズリサーチとセットで使う：オペレーションズリサーチ：システム工学と訳したことによる誤解

2.5 オブジェクト指向の言葉の意味

オブジェクトは理解し難い用語：先ずオブジェクト指向プログラミングから：オブジェクト指向データベースとは

2.6 Structure の意味

Structured programming：構造化プログラミング：スクリプト言語

2.7 データベース作成の前段階

目録作成は書籍だけに限らないこと：文献調査とガイダンス：実体を見る手段にマイクロフィルムを使う：メモ管理もデータベースと言う誤解

3. 目録の作成からデータベースへ

3.1 文献調査のまとめが最初の目的

欧米文献を有り難がること：本州四国連絡橋の計画時の文献調査：委員会方式の研究活動が犯し易い欠点

3.2 知的好奇心を満たす目的に使う

本体にアクセスするのが基本的な態度：本体に代わる情報で妥協する：データベースの作成前にファイル管理から入る：実物に当たる態度を育てる

3.3 資料の静的管理と動的管理

静的なデータベースとその管理：ユーザの利用で内容が変わるデータベース：個人が閉鎖的に使うのはデータファイル：教育利用を意識する共用データベース

3.4 計算機械の原理から

整数しか計算できないこと：レジスタの寸法で最大整数の大きさが決まる：データに番号を割り当てるときの数の制限

3.5 ファイル装置の基礎知識

コンピュータ立ち上げの儀式：磁気テープと磁気ディスク：固定長データと可変長データ：高速でデータを移しかえる命令がある：表形式のデータを順編成ファイルで利用すること：文書の目次や索引はデータベースの性格がある

4. データベース入門

4.1 英語用語の理解から

DBMS：データベースエンジン：DDL, DML, DCL：スキーマ：マスターファイルとトランザクションファイル：マクロ：DLL：中身の読めないデータファイルの問題

4.2 ハードウェアとソフトウェアの要求事項

専用のコンピュータを使うのがよいこと：表計算ソフトの利用から：カード型データベースが使い易い

4.3 ワープロ用辞書への要望

シソーラスの作成が準備作業として必要：正引きと逆引きの辞書：文字校正用のツールに辞書を選択する：辞書には種々の種類があること：自動翻訳を考えるととき：プリントサービス

5. オブジェクト指向データベース

5.1 何をしたいかの問題意識

出来合いのソフトは無いのが普通：ベンチャーソフトを使わない：橋梁資料のデータベース化の試み

5.2 図面データの電子化技術

画像データベースが重要な課題：最初はビデオテープの利用を考えた：デジタルカメラの普及が情勢を変えた：図面保存のファイル形式：実用的な寸法の画像に作り直す：画像の縮小には賢いソフトが必要

5.3 写真データの目録作成と利用技法

例題として橋の切手を目録化したこと：分類番号の設計：目録サービスに使うデータの整理：二種類の画像データを編集する：ユーザが利用する方法：データの管理者側の資料整理

終わりに

用語索引

英字

.chm	5.3.5
.gif	5.2.4
.jpg	5.2.4
.mdb	4.2.1
.tif	5.2.4
ABC 順	4.2.3
ACCESS	4.2.2
Active X	4.3.6
ADSL	5.3.4
API	2.6.3
ASP	4.3.6
CAD	5.2.5
CCD 素子	5.2.3
CD-ROM	5.2.5
COBOL	2.6.1
CODASYL	2.6.1
CPU	3.5.4
CSV	5.3.6
CUI	2.3.1
C 言語	2.6.1
DBMS	4.1.1
DCL	4.1.3
DDL	4.1.3
DEMOS	2.1.2
DLL	4.1.7
DML	4.1.3
DPI	5.2.3
EDP	2.1.1
EXCEL	4.2.2
FORTRAN	2.6.1
GUI	2.3.1
HTML-Help	5.3.5
JavaScript	2.6.3
JIS 漢字コード	2.1.1
JPS	5.3.2
JSDA	5.3.2
KJ 法、	1.3.5
KVS	4.2.3
MICHEL カタログ	5.3.2
MO	5.2.5
NDC	1.2.5
NTSC	5.2.2
OA	2.1.2
OR	2.1.3

OS	2.4.1
PDF	3.5.6
RAM	3.5.2
RDB	4.1.4
RDBMS	4.1.4
Scott カタログ	5.3.2
SE	2.4.1
SQL	4.1.1
Structured programming	2.6.1
UDC	1.2.6
University Microfilms International	2.7.3
VBS	2.6.3
Visual Basic	4.1.6
WYSIWYG	2.2.3

あ行

アーカイブ	1.3.1
あいうえお順	3.5.6
アドレス	3.4.3
安全工学	2.4.3
いろは順	4.3.2
インターネット エキスプローラ	4.3.6
一次資料	1.2.5
梅棹忠夫	1.3.5
閲覧ソフト	5.3.5
オーバーフロー	3.4.1
オーバーレイ	4.1.7
オフィスオートメーション 1.1.4	
オブジェクト 2.5.1	
オブジェクト指向 2.5.1	
オブジェクト指向グラフィッ クス 2.5.3	
オブジェクト指向データベー ス 2.5.3	
オブジェクト指向 プログラミング 2.5.2	
オペレーションズリサーチ	2.1.3
オペレーティングシステム	2.4.1
応用プログラム	2.6.3

か行

カード型データベース	4.2.3
カメラレディ	2.2.2
下位語	1.2.6
可変長データ	3.5.3
夏炉冬扇	1.2.1
画素数	5.2.3
画像データベース	5.2.1
解像度	5.2.3
開架式	1.2.4
階層構造	4.3.1
漢字テレタイプ	2.2.1
関連語	1.2.6
川喜田二郎	1.3.5
キーワード	1.1.4
キャッシュメモリ	3.4.3
キャビネ判	5.3.4
危機管理	2.4.3
技術の三要素	1.2.2
逆引き	4.3.2
クラウドサービス	4.2.3
クラス	2.5.2
グラフィックス言語	5.2.4
桁溢れ	3.4.1
桁落ち	3.4.1
検索	0.
コンテンツ	1.3.2
コンピュータ	1.1.1
コンピュータウイルス	4.1.6
固定長データ	3.5.3
公文書館	1.3.1
構造化プログラミング	2.6.2
構文	2.6.1
国際十進分類法	1.2.6

さ行

サーバー	4.3.6
サムネイル	5.2.1
差別用語	4.3.2
索引	3.5.6
システム	2.4.1
システムエンジニア	2.4.1
システムズエンジニアリング	2.4.1

システム工学	2.4.3
シソーラス	1.2.6
司書	0.
資料	1.1.3
事実と意見	1.1.3
事典	2.3.1
磁気ディスク	3.5.2
磁気テープ	3.5.2
自然語	1.2.6
自動活字鑄造機	2.2.1
自動翻訳	4.3.5
辞書	2.3.1
写植	2.2.2
写真植字	2.2.2
写真製版	2.2.2
順編成ファイル	3.5.2
抄録	2.7.3
上位語	1.2.6
情報	1.1.3
情報科学	0.
情報学	0.
情報検索	1.1.2
情報処理	0.
新聞	2.2.1
スーパーコンピュータ	1.1.2
スキーマ	4.1.4
スクリプト言語	2.6.3
スパゲッティコード	2.6.2
スプレッドシート	4.2.2
図面	5.2.1
セル	3.5.3
整頓	1.3.8
整理	1.3.8
整理法	1.2.1
正引き	4.3.2
専門用語辞典	1.2.6
線図	5.2.4
ソース	1.2.5
ソフトコピー	4.3.6
そろばん	3.4.2
倉庫	1.2.1

た行

対象	2.5.1
弾道計算	1.1.1
段組み	3.5.6
立ち上げ	3.5.1
中間言語	4.3.5
著作物	5.3.1
諜報	1.1.3
超整理法	1.3.5

ディスクキャッシュ	3.5.2
データ	1.2.5
データファイル	3.3.3
データベース	1.1.4
データベース (静的な)	3.3.1
データベース (動的な)	3.3.2
データベースエンジン	4.1.2
データベース管理システム	1.2.4
デジタルカメラ	5.2.3
デバイスドライバ	4.1.1
デリミタ	3.5.5
手札版	5.3.4
定義と宣言	2.3.1
電子データ処理システム	2.1.1
電子化書籍、	4.2.3
電卓	3.4.1
ドットマトリックスプリンタ	2.1.2
トランザクションファイル	4.1.5
図書カード	1.3.2
図書館	0.
図書館学	0.
図書館情報学	0.
頭字語	2.3.1
同位語	4.3.1

な行

二次資料	1.2.5
日本十進分類法	1.2.5
日本郵趣協会	5.3.2
日本郵便切手商協同組合	5.3.2
濃淡図	5.2.4
野口悠紀雄	1.3.5

は行

ハードカバー	1.3.1
ハードコピー	4.3.6
ハッカー	5.1.1
パンチカード	1.3.6
原口忠次郎	3.1.2
博物館	1.2.1
反義語	2.6.2
汎用コンピュータ	1.1.2
ピクセル	5.2.4
ビジネスモデル	1.1.5

ビットマップ	5.2.4
ビットマップフォント	5.2.4
ビデオテープ	5.2.2
光ファイバー	5.3.4
光磁気ディスク	5.2.5
秘書	1.2.2
美術館	1.2.1
百科事典	4.3.4
標準モジュール	2.5.2
表計算ソフトウェア	4.2.2
フォームモジュール	2.5.2
フォント	2.2.2
ブックカード	1.3.7
プライマー	2.3.1
フラグ	3.5.4
フリガナ	3.5.6
プリントサービス	4.3.6
プルダウンメニュー	1.2.6
プロパティ	2.5.3
藤井郁夫	5.1.3
浮動小数点数	3.4.1
物	1.2.5
分掌	1.1.5
分類語辞典	1.2.6
文献調査	2.7.2
ページ	3.4.3
ベクターフォント	5.2.4
ベンチャーソフト	5.1.2
閉架式	1.2.4
保存図書館	1.3.1
放送大学	5.2.2
砲術	1.1.1

ま行

マイクロフィッシュ	2.7.3
マイクロフィルム	2.7.3
マイクロフィルム	5.2.5
マクロ	4.1.6
マスターファイル	4.1.5
乱れ籠	1.3.4
メソッド	2.5.3
メモリ	3.4.2
メモ帳	3.5.6
森鷗外	1.1.3
文字処理	2.2.3
目次	3.5.6
目次と索引	4.3.2
目的語	2.5.1
目録	2.7.1

	やゆよ				
有価証券	5.3.1	ライブラリアン	0.	レジスタ	3.4.2
郵便切手	5.3.1	ランダムアクセスファイル	3.5.2	レポジトリ	1.1.3
予約語	2.3.1	ランダムアクセスメモリ	3.5.2	連想	4.3.1
用語解説	2.3.1	リレーショナルデータベース	2.6.3		
用語辞典	4.3.4	レーザーディスク	5.2.2	わ	
		レーザープリンタ	2.1.3	ワード	3.4.3
				ワードプロセッサ	2.1.3
	ら行				

0. はじめに

何かの書物資料を探したいとき、**図書館**(library)を利用することを思い付くでしょうか？ ここでは、**書物**(=書籍、図書、本)を財産扱いますので、分類番号などを付けて整理・保存をします。目的の書物を探す**検索**(retrieval)の手助けに**図書カード**(library card)などが整備され、種々の**目録**(bibliography)も備えます。この全体についての専門家が**図書館司書**(ライブラリアン: librarian)です。もともと、**情報学**または**情報科学**(information science)は、大枠として文科系の専門に分類されていて、**図書館学**(library science)は、その一分野です。一方、コンピュータに関する学問の方は、大枠を computer science に分類します。情報科学の性格もありますが、理工系の応用技術の性格を持ちます。しかし、コンピュータのようなカタカナ語を学問名にすることを避けるため、これを**情報工学**と言う大枠に含めたことことから、混乱が始まりました。コンピュータを利用するあらゆることを、**情報処理**と総称して、情報をコンピュータと読み替えるような解釈になってしまいました。図書館学にコンピュータの利用も必須になってきましたので、**図書館情報学**(library and information science)と言うようになりました。コンピュータは、所詮、道具です。目的に合わせて使い方の研究をします。**データベース**(database: DB)の開発と利用の研究は、文科系専門の性格もあります。コンピュータが開発されると共に直ぐに始まっていたのですが、理工系専門の数値計算への利用に比べると、地味な扱いを受けていました。データベースの利用を、筆者が、専門とする橋の研究に役立てたいとする要望は自然です。**文献調査**(literature survey)は、必然的にデータベース的な利用につながります。現在(2011)では、個人レベルでパソコン(personal computer)を利用して、私的なデータファイルを作成するための道具(ツール)も多くなりました。それを使いこなすところまで行くには、図書館が長年努力してきた図書館情報学についても、一通りの常識を埋めることから話しを始めます。

1. データベース構築までの背景

1.1 軍事研究に始まったコンピュータの開発

1.1.1 弾道計算の高速化が主目的であったこと

悲しいことに、多くの科学技術は、殺戮と破壊を目的とする軍事研究に応用することで、豊富な予算が使われてきました。**コンピュータ**(電子計算機: electronic computer)技術も例外ではありません。その主目的は、高速で移動する敵の軍用機を撃墜する高射砲の制御をするとき、軍用機の位置と速度などの測定データを利用して、狙いとタイミングを素早く計算して弾丸を発射することにあります。日露戦争(1904-1905)の日本海海戦では、軍艦に搭載した巨大な大砲の制御が必要でした。軍艦の動きは、航空機に比べれば低速ですので、そろばんなどを含めた単純な計算機械を使った手計算で行われました。この技術を**砲術**と言い、数値計算の専門家が養成されました。原理は、力学計算を踏まえた**弾道計算**です。現在では、人工衛星の軌道計算がその延長です。開発当初のコンピュータは、巨大な設備でした。これは、軍艦に搭載することも考えにあったようです。世界最初のコンピュータは、アメリカの ENIAC (1946)とされています。しかし、軍事機密として隠されていた装置が、既にイギリスで Colossus (コロサス)として開発されていた、と言うのも定説です。

1.1.2 コンピュータを事務処理に使うこと

コンピュータは、科学技術計算の道具としての使い方が主題でした。その目的以外の利用法、つまり会計計算、統計処理などの事務処理、印刷に関係する文字処理などは、あまり注目されませんでした。事務処理に利用することを、軍用に対して商用、さらに、大衆化することを民生利用と言います。後の二つの利用法が、実は、コンピュータ処理のマーケットでは大きいのです。コンピュータのメーカーは、会計計算に便利な装置の開発にも焦点を移し、名称として**汎用コンピュータ**と言いました。科学技術計算用には幾らか不便になりましたので、数値計算を主目的とする**スーパーコンピュータ**を別に開発する方向に向かいました。筆者のデータベースとの関わりは、1970年代の始めからです。当時、筆者が所属していた名古屋大学での汎用コンピュータ FACOM230-60(1968)は、研究目的の利用に科学技術計算を掲げていましたので、**情報検索**(information retrieval)を始め、実務的な情報処理に大学の大型計算機センターを利用することは、規則で許されていませんでした。筆者が、コンピュータ言語の FORTRAN だけでなく、COBOL も使っていたことがバレて、研究目的の利用とは認めないと言って、一か月程の利用差し止めを食らったことがありました。

1.1.3 資料と情報の用語の意義

データベースは、軍事作戦遂行に必要なデータを集めて、分析に利用するための施設を作ったことが出発です。古くは、**諜報**活動と言ひ、スパイ活動に関連した意義で使いました。これに和製漢語の**情報**を当てたのは、軍医でもあった**森鷗外**(1862-1922)である、とするのが定説です。これは、「敵情報告」の中央の二字を取ったものと解釈されています。情報に英語の information の訳を当てたのは正解ですが、コンピュータを利用することと同義に解釈されることが多いのが混乱を伴うようになりました。情報は抽象的な概念です。より具体的な意味を持たせるときは、データ(data)と使い、漢字では**資料**を当てます。英語の data は、datum の複数形です。information と同じような集合名詞の意義で使うようになりました。英語辞書の説明には collection of facts の説明を見ます。ここで、facts と複数形で使っていることに注意します。英語の名詞は、単複を神経質に区別します。したがって、情報またはデータの、さらに複数の集合を表す表現が必要になるとき、物質名詞なみに、例えば、two pieces of information の言い方を見ます。Collection も集合名詞の性格を持ちます。加算名詞として、集合の、そのまた集合の意義を持たせて複数形でも使います。データベース(database)は加算名詞ですが、集合名詞の意義を持ちます。そこで、データベースの集合に対して二つの言い方が必要になりました。一つは施設またはシステム単位で勘定する databases の複数形で表すとき。二つには、中身の data 集合の、そのまた集合を意味したいときの言い方です。後者に対しては、repository (**レポジトリ**) の別用語を見ます。**リレーショナルデータベース**も一つの形態です。英語の fact も、訳し難い抽象概念です。**事実**と当て、客観的に**記録**したもの、とします。記録の段階では、恣意的な選択もありますが、感情的な**主観**または**意見**が入っていないとします。ちなみに、論理的な実用文書をまとめるときの作文は、**事実**と**意見**とを峻別することの教育が必要です。

1.1.4 データベースは軍事用語であったこと

科学技術計算以外のコンピュータ利用は、大卒としての事務処理です。**オフィスオートメーション**(office automation: OA)の言い方があります。"office, file, folder"などの用語は、事務処理の方から転用して使われるようになったものです。事務処理の中の重要な課題の一つが、資料本体の管理です。データベースは、アメリカ軍の軍事用語に起源のある用語です。軍用機専用の飛行場を air base と言います。軍事作戦は、データを収集して分析に利用しなければなりません。このデータ収集と利用の基地と言う意義が data base です。初期には、2語を並べましたが、ハイフンで繋ぐ data-base を経て、1語の **database** と使うようになりました。データベースの施設は、図書館と同じ性格を持つ作業環境です。図書館は、製本された書物の形にしたものを主な管理の対象とします。データベースの方は、論文や報文の中身の要点を手掛かりとして所在情報、さらに記事本文に当たることができるように管理します。手掛かりに使う分類語を**キーワード**(keyword)と言います。これも2語の key word を詰めて一語で使うようになりました。

1.1.5 データベース利用の大衆化へ

軍用データベースの管理運営の技術を、一般の図書館にも応用する計画は自然です。しかし、図書館の歴史が古いこともあって、データベースの管理技術を導入することには多くの障害がありました。一つには、コンピュータが理工系の数値計算の道具であるとする先入観が強かったこと。二つは、従来の管理手法とコンピュータを使う管理手法とを共存させながら、段階的にシステムを改善して行かなければならない現実的な課題です。文科系の立場でコンピュータを利用するにも、何がしかの研究体制が必要です。既存の図書館には、職務上の**分掌**がありますので、データベース研究の環境がありませんでした。しかし、個別の研究者は、データベースがあれば研究上の資料の管理が便利であるとする認識もあって、個人単位での情報検索の開発も行われるようになりました。しかし、データベースを実用的なシステムとして構築することは、個人的な作業の手に余ることも判ってきました。そこで、共同でデータベースを作成し、共同で利用する、その方法の模索が幾つかありました。研究開発には費用も必要です。それを生み出す民間の**ビジネスモデル**が幾つか試みられましたが、長続きした例は多くありませんでした。コンピュータ技術が進歩してきたことと、インターネットの利用が大衆化してきましたので、プロバイダが提供するデータベースの検索手法が、個人の環境でも経験できるようになりました。その検索先に、共同で利用する実用的なデータセンタを構築することの提案が、ようやくできる時代になりました。

1.2. 整理と保存の現実

1.2.1 整理法を教える組織的教育がない

公私に関わらず、我々は、日常的に、何かを探す作業と、保存と再利用を目的とした整理作業に多くの時間を取られます。一般に、整理が良いと言うのは、行儀作法的な素養が関係します。しかし、合理的な整理方法を教育することは、従来 殆んどありませんでした。合理的な整理をするためには、保存の空間を別に必要とします。欧米をはじめ、冬の寒さが厳しい所では、冬籠りを考えて、物を溜め込む倉庫の空間を持つ習慣があります。日本の住建築は、従来は、開放的で多目的に応用します。押入れを個別の部屋に設けます。それでは足りないとき、収納用の家具を持ちこみますので、その分だけ部屋が狭くなります。夏炉冬扇という言葉があります。収納場所が別に無いので、夏冬で別に使う道具が一部屋に同居した状態の意味に理解することもできます。公共的な建築物でも、倉庫の空間を最初から準備しておく設計習慣がありません。また、その空間があったとしても、そこを機能的に活用する技法も、経験的に覚えます。廊下や通路を物置のスペースにするのは、よく見かける光景です。大学の研究者は個室を欲しがりますが、公的な空間を私的な物置のように使っているのが常態です。原則として、公的な場所に私物を持ちこむべきではありません。また、公的な資料を自宅に持ち帰って作業することもしません。このことは、一般企業の作業環境ではごく普通の規則です。図書館は、書物を公的に整理保存する機関です。図書以外のものを保存する施設が博物館(museum)、美術館などです。これらは一般向けに展示の空間を持ちますが、それを支える大きな、かつ、安全な倉庫と作業用空間が別に必要です。

1.2.2 整理法は技術であること

整理には上手・下手がありますので、一種の**技術**です。一般論で言います。技術の中身は、三つに分けて考えることができます。これを筆者は、**技術の三要素**「道具・技法・技能」として言っています。現代風に言えば、「ハードウェア・ソフトウェア・インタフェース」です。専門家が居ても不思議ではありませんし、そうあるべき時代になりました。書類の整理で言えば、例えば、書庫・書架・ファイリングキャビネットが道具、分類法が技法、そして、書類の扱い方が技能に当たるでしょう。公的な環境では、前の二つは規則を決めることができます。技能を持つ専門家としての資格が、図書館司書です。欧米の管理職につく女性の**秘書**(secretary)は、職業としての位置が社会的に認められています。この秘書がコンピュータユーザの一大勢力です。マイクロソフト社が Office の名称を付けたソフトウェア製品は、秘書の希望を反映するように発展してきました。秘書の仕事は、かなりの部分が整理です。また、女性の方が整理を面倒がらずにやってくれます。このとき、上にあげた三つの区分を納得していないと、整理については何も解決しません。三つの区分を理解できれば、整理法の教育を具体的に立てられます。道具の中に、パソコンが加わったので、この取り扱いも秘書の素養として必須になりました。

1.2.3 公私の区別をすること

個人が書物資料を保存して利用する場合は、所蔵点数としては知れています。趣味で骨董的に私蔵するのではなく、実質的な利用を目的とする場合の仕分けを、三種類に分けて考えることができます。

- ・ 一度読めば済む一過性のもの（小説・新聞・雑誌など）；
- ・ 他人も含めて、何度も利用するもの（辞書・参考書など）；そして
- ・ 消耗品的に使うもの（教科書・演習目的のテキスト類・コピーした資料、ノートなど）です。

最初のもは、多くの部数で発行されますので、時期を見て廃棄します。全体保存を考えるのは、本来、発行元の責任です。公的には、特に国立国会図書館が保存図書館として大きな貢献をしています。2番目の扱いには、資料を利用する人の価値観に左右されます。最後のものが私物です。書き込みなどをして汚します。物を大切に扱うことは美德です。汚して使うのは消耗品扱いです。こちらは本人の責任で、保存か廃棄かの判断をします。大学の研究者が書物資料を保存する場合、公私を区別する意識が薄いようです。退職などで、資料や蔵書を処分しなければならなくなったとき、私物を図書館などに寄贈する例もあるのですが、受け入れ側が迷惑することが少なくありません。その理由は、余分な保存空間と、目録を作るなどの作業が必要になるからです。保存利用を考える場合は、図書館を介して公的に分類・登録した上で私的に借り出す形にしてあれば、この問題の大半は解決されます。共通に利用できる小規模の共通図書室でもあれば、一過性の閲覧利用ならば実質的には便利ですが、しかし利用者（ユーザ）の利用方法（マナー）で問題が多く発生します。個人が書物を排他的に保存したがる理由の一つが、このマナーにもあります。コピーが便利でなかった頃、盗難や紛失もありますが、かみそりでページが切り取られる被害に悩まされたものです。

1.2.4 開架式と閉架式の管理方式

共通に利用する図書の整理と保存をハードウェア的に見れば、書庫を設けてそこに書架や書棚を置きます。分類法を決めて、それを使って保存スペースを割り振ります。利用者が書庫に自由に入れて、そこで見るだけでも済む管理方式を**開架式**(open shelf)と言います。これに対して、書庫への立ち入りを制限し、ユーザが目録などから検索して、司書を介して書物を借り出す方式や、閲覧室を設けて閉鎖的に利用する方式が**閉架式**です。開架式で管理するときは、利用者の良いマナーをいつも期待できませんので、安全管理の問題が発生します。庶民レベルの書物の利用は、日本では、書店での立ち読み、古書店、貸し本屋スタイルに特徴があります。後の二つは、考えてみると、私設図書館としての**ビジネスモデル**です。日本の書店は、立ち読みができる開架式図書館的な利用がされています。欧米の書店は閉架式に近いので、立ち読みは追い出されますし、買わなければ読めないような包装もあります。図書を情報源と見るとき、情報を公開するか秘匿するかの区別は、物理的には図書の保存方式を開架式か閉架式かにすることに対応します。コンピュータの普及によって、情報は記憶媒体に保存され、直接見ることができなくなっていますので、言わば閉架式の管理方法になっています。したがって、丁寧な案内手段がなければ、情報を取り出して利用することができません。これが、ソフトウェアとしてのデータベース管理システムの考え方に繋がります。

1.2.5 二次資料の作成作業が大変

データベースでは、データ(data)を**資料**と訳し、情報元は、書物本体、または広く解釈して、**実体**の有る**物**(object)です。**一次資料**とも言い、英語の source (**ソース**) が当たります。風聞または伝聞は、ソースが無い情報です。一旦広まると見掛けの実体を持つようになる恐さがあります。データは、測定装置で得られる、眼に見えない電気信号なども含みます。情報は、データとほぼ同義です。ただし、何かの言語を介して文書化し、眼に見える形態に変えたものを指します。これが**ソフトウェア**に当たります。科学技術の測定装置で得られる電気信号は、デジタル化すればビット情報になり、(1,0)の符号文字で文書化したバイナリーファイルを作成できます。これは一次資料扱いをします。一次資料の要点をまとめた資料を**二次資料**と言います。図書文献の場合には、表題・著者名・発行年・発行機関、そして内容紹介の**抄録**(abstract)を含めることがあります。日本の書籍は、これらの情報を**奥付**(imprint)にまとめます。これは良い制度です。欧米の書籍は、この習慣が一般化していないからです。図書文献のデータベースに使う材料は、この二次資料です。従来から図書館では二次資料をカードや目録などの形で作成するのですが、奥付の項目の他に、分類番号を付けます。日本では**日本十進分類法**(NDC: Nippon Decimal Classification)を使います。図書館での蔵書目録をデータベース化するときは、古くから保存してある書籍の全目録も電子化資料にしなければなりません。これには、膨大な手間と費用とが掛かります。

1.2.6 分類番号の代わりにキーワードを使う

図書は種々の言語で出版されますので、表題に書いてある文字並びの意味が分からなくても、適切な分類のところに一応保存しておく必要があります。番号を使う分類であれば、司書の作業で迷うことはありません。この場合、言語が異なっても共通の番号数字を使います。ヨーロッパ圏では、**国際十進分類法**(UDC: Universal Decimal Classification)が使われています。ただし、日本とアメリカの分類法は、これと整合していません。この番号と分類語との対照表は、言語ごと、または国ごとに決めています。一つの言語環境では、分類番号を使うよりも、普通の用語(**自然語**と言うことがあります)を使う方が便利です。これを、**キーワード**の提案に繋がります。分類語の種類が増えないように、用語を制限します。その用語を集めたものを**シソーラス**(thesaurus: **分類語辞典**)と言います。通常、専門ごとの学術用語にしますので、**専門用語辞典**とも言います。日本語の環境ならば、日本語をキーワードにすることができます。キーワードとして使うときは、同じ意味で、用語の揺れがなく、コンピュータの文字処理に向くように決める必要があります。例えば、英字綴りでは、data base, data-base, data.base, database, databasesなどをすべて認めるのではなく、database一つにします。そうすると、それ以外の綴りを使うと、検索漏れが起きます。これを防ぐソフトウェアの技法の一つが、パソコンの使い方にある**プルダウンメニュー**です。これは、大分類を**上位語**(BT: broader term)にして、サブメニューがツリー構造の**下位語**(NT: narrower term)の種類を表示してくれます。ユーザがキーワードを思い付くことと、キー入力の手間を省くことができます。これが**インタフェース**に当たります。なお、上位語、下位語、さらに**関連語**(RT: relational term)は、シソーラス関係の用語です。

1.3 古典的な整理方法

1.3.1 古典的な書物保存の背景

もともと、図書館は、耐久性のある厚手の表紙を付けて、上製本された書物（ハードカバー）を縦に並べて保存します。製本は費用が嵩みますので、商業出版物では簡易製本で済ますペーパーバックや、雑誌形式も多くなりました。こちらは縦に並べて保存する整理方法に向きません。和書もそうです。これらの保存整理が面倒です。公共図書館では、利用者の便も考えて、新書版のようなペーパーバックも置きます。雑誌類は、適当な期間の展示が済めば、廃棄します。したがって、製本形式にとらわれずに保存を目的とする図書館を保存図書館と言います。代表的には国立国会図書館がそうです。英語では archives（アーカイブ）と言い、通常は複数形で使います。こちらの用語は、主な対象を官公庁の重要文書を扱う施設に使い、日本語では公文書館と訳しています。これを広く解釈させて、変更が無い歴史記録として残したい物を保存する施設にも言います。コンピュータのファイル管理でも眼にする用語になりました。クラシック音楽の愛好家には、ドイツ語読みのアルヒーブと説明すると納得してくれます。

1.3.2 雑誌はまとめて製本する

保存図書館は、雑誌なども含め、発行された書籍類を原形で保存します。一般図書館は、必要と認めれば、製本した上で登録・保存の手続きをします。雑誌の発行元は、それを助けるため、年単位の区切りで総目次を付けることがあります。学術雑誌は、この方式を採用しています。雑誌の状態で、通年のページ番号を付けておくこともします。通常は、奥付情報などを図書カード（library card）に作成して利用者の便を図ります。一方、雑誌を製本した書籍は、複数の著者が種々の表題で論文を寄稿していますので、個別の中身（コンテンツ：contents）の要点（二次資料）をカード化することまではしません。したがって、この情報を電子化して検索に使うことが、データベースの第一義的な利用方法です。所在情報が分かれば、論文の中身を見るために保存図書館に行きます。見るだけで済むことができなければ手書きでノートに取るか、複写を依頼していました。このような利用方法は、利用者にとっては不便ですので、通信回線で結んだ端末で、抄録だけでなく、論文本文も読める全文検索が要望されるようになり、その延長が電子書籍の利用へと進んでいます。なお、製本について付け加えると、写真撮影でコピーをするには、ページを開いても左右ページが連続平面を保つような、上製本でなければなりません。そうでないと、書籍本体を傷めますし、コピー画像も歪みます。

1.3.3 カードによる管理の得失

狭い意義でのデータベースは、書物の要点をまとめた情報、個別の目録内容や文献情報、とかく問題になる個人情報など、つまり二次資料を単に集めたものを言います。コンピュータ化が進む以前は、データベース的に検索する作業に、図書カードを使いました。本体（一次資料）の所在情報が分かればよいとします。どのような蔵書があるかを探す手立てに、図書館ごとに図書カードが準備されてきました。カードは、差し替えが簡単にできますので、蔵書が追加されても、アルファベット順または 50 音順のカード並びの変更が簡単にできます。印刷した蔵書目録は、特別な、追加の無い書物の集合を紹介するときに作成します。登録された蔵書は、廃棄を考えません。新聞、雑誌、パンフレットなどは、保存に向く設備があるか、書物なみの体裁にしなければ、ある期間の公開展示が済めば廃棄します。これ以外、レコードやテープなどの新しい形式の資料は、取り扱い規則がなければ、業務に組み込むことができません。定型的な運営をする場合には、弾力的な対応ができません。こちらは、研究者、個人、民間の機関などが、いわば、私的なボランティア活動で支えることでバランスが取られてきました。

1.3.4 受動的にカードを使う私的な整理法

整理の実践的な技術のうち、メモ用紙、カード、付箋などを補助的な作業媒体として使う技法が種々工夫されています。葉書、特に年賀状は、住所と氏名の情報が記録されたカードです。名刺は企業の顔を持つ個人情報のカードです。これらは、他人からもらって集まるものであって、積極的に収集しません。寸法が規格化されていますので扱い易くなっています。これらを私的に整理する方法には、種々の工夫があります。最も単純には、整理の手間を省いて、適当な箱に放り込んでおきます。筆者は、これを乱れ籠方式と言っています。整理が必要になったときに、乱れ籠の中を調べ、処分してもよいかどうかの判断ができるようにします。この方法は、パソコンのゴミ箱を管理するときに応用されています。少し工夫した整理を考えると、複数の乱れ籠を使います。日常的には、ゴミの分別がそうです。経験的に理解できると思いますが、分別、つまり分類は面倒なものです。

1.3.5 積極的にカードを使う私的な整理法

メモ用紙やカードに書き込んで知的な作業に使うアイディアは、古くから試みられていました。ノートに書くと、並べ替えが不便だからです。ルーズリーフ式のノートや、綴じ込みファイルは、差し替えが自由にできることで愛用されます。しかし、穴を空けて綴じると、穴の個所の文字が飛ぶことや、見開きでの利用に不便です。そのため、バラのまま保存する形式の文房具も売られています。用紙寸法としてA4版またはB5版では寸法が大き過ぎますので、作業用には、葉書き大(A6版)以下のカードを使います。川喜田二郎(1920-2009)の名前のついたKJ法、梅棹忠夫(1920-2010)の京大式カード利用法は、それなりの評価がありました。しかし、中身を自分で作成して使う技法は、情報管理には基本的に不便です。整理というのは、当面の仕事の後始末をして、次の仕事に備える作業ですので、言わば、後向きです。コンピュータを利用する場面では、カードで整理することを意識下においた整理モデルを考えると、理解し易くなります。超整理法の言葉は、野口悠紀雄(1940-)の造語に始まりました。内容は、書類情報の私的な整理法について、パソコンを利用することを軸に、個人的な経験をまとめたものです。一般的に言えば、メモを別に作ることは、それを積極的に利用する場面が、いつになるかが決まっていないと、制作意欲が長続きしません。

1.3.6 パンチカードを使ったデータ管理

初期のコンピュータは、文書データの作成にパンチカード(punched card)やパンチして使う紙テープを使っていました。それなりの便利さもありましたが、大量の紙資源を費消することと、広い保存場所が必要ですので、これがフロッピーディスクの利用などに代わり、現在では使われません。情報を書き込んだ、やや寸法の大きなカードの集合から、或る複数の検索項目に合うカードを取り出す方式に、ホールソート方式が使われたことがあります。カードの縁に複数のパンチ穴を開けたカードを作っておき、穴の位置に検索語を決めておきます。複数の細い針金を検索語の個所に差し込んで上に引き上げるようにしてカードを取り出します。検索語以外の項目は、穴から縁までを切り取っておくと、検索語にヒットするカードだけが取り出されます。アイディアは良いのですが、カード穴が傷むと検索もれが起きますし、また大量のカードを扱うには不向きです。カード化した資料は、直接 目で見る利用法が便利ですが、便利さを欲張ると、使い勝手が悪くなります。したがって、この方式もすたれました。



図 1.1 文房具として売っていたホールソート（ケルンパンチカード）の用紙と道具
インターネットから検索して採図

1.3.7 カードを使う管理も捨てたものではないこと

パソコンの性能が向上してきましたので、カードを媒介とする情報処理は時代遅れであると考えられると間違えます。葉書きや名刺の利用は、すたれることはないでしょう。欠点と指摘されることの一つは、保存場所が必要であることです。利点の方を言うと、何も特別な装置を使わなくても読めることです。小学校にあるような小規模の図書管理では、書物に**ブックカード**を付けておいて、借り出す人が署名してこのカードを図書室に残しておきます。書物を探すとき、最初から無いのか、誰かが借り出しているかが簡単に分かります。コンピュータを使うことを過信すると、装置が無ければ、また故障でもすれば、確かめることができません。コンピュータの外部記憶装置は年々改良されてきました。しかし、磁気テープ、フロッピーディスク、レーザーディスクの規格が目まぐるしく変わりましたので、新しい規格に合わせるバックアップを怠って、読めなくなって死滅した資料も増えています。時代遅れのように見えても、紙に記録した形で保存することは意義があります。

1.3.8 保存用のハードウェア

一次資料も二次資料も、保存場所と保存のための入れ物の用意が重要です。倉庫の必要性については、1.2 節の始めに触れました。そこには、さらに入れ物を使います。日本の気候は湿気が高いので、湿気を遮断できる入れ物の工夫があります。伝統的な日本の保存具は、桐のタンス、桐の箱を使います。一歩譲って木製の長持があります。これらは、嫁入り道具の定番です。筆者は、内側にブリキを張った長持風の茶箱を衣類の保存用にも使っています。欧米人は、日本の古い家具の愛好家が少なくありません。木製の入れ物は、入れる方の物に対して傷を付けない、などの優しさがあるからです。最近では、実用的なプラスチックの入れ物が豊富になりました。しかし、鉄製の入れ物は保存用具にはあまり見かけません。これらは、製作が特殊になります。汎用性がないことも理由ですが、日本では腐食が問題になります。この点では欧米諸国では湿気の問題をあまり考えませんので、事務用品としても各種の鉄製のキャビネットが使われています。家具として見ると、単純な塗装は装飾性や温かみがありません。日本では木目を描いた壁紙などで表面を飾る製品を良く見ます。資料の点数が大量になると、保存用の入れ物単位は、なるべく単純なデザインで、追加ができる機能が重要です。見てくれはよくないのですが、昔は木製のリンゴ箱、最近では段ボール箱が、規格の揃った入れ物として実用されています。保存場所は、空間利用を考えた書架のような頑丈な棚を備えます。見えるように保存し、ほこりや湿気を防ぎ、さらに盗難などを防ぐため、オプションとして外側にガラス戸を付けることもあります。高さ方向の利用ができないときは、平面的に並べるしかありません。この状態を、小間物屋を開くと言うことができます。これが整理の乱雑さの元凶です。整頓は、高さ方向に積み上げて、自由平面を増やすのですが、これは整理ではありません。空間を有効に使うために入れ物が必要になります。図 1.1 に示したホールソートも、その入れ物が特殊な構造になることもあって、結果的に普及しなかったのです。

1.3.9 事故や災害で失われる資料も多いこと

筆者の縁者には、関東大震災のとき、戸籍情報が焼失し、さらに空襲でも被害を受けました。お寺には過去帳がありましたが、そのお寺も焼けてしまいました。分かる範囲で戸籍の復元はしましたが、御先祖さまの記録が分からなくなりました。重要な資料の保存は、歴史に堪えることが必要です。東北大地震とそれによる津波で、役場の資料が全滅した所がありました。この地震の規模は 1000 年に一回起こる確率だそうです。しかし、地球の何億年の歴史を考えれば、瞬き程度の短い期間です。この 10 年くらいの間に、比較的気候が安定していると考えられていたヨーロッパでも、大きな河川の氾濫が何度か起きました。地下室に書庫を設けていた図書館が、大きな被害を受けました。戦争のような人為的な破壊による資料の損失もあります。日本では、大学紛争で占拠された施設や研究室で、多くの重要な書類が被害に遭いました。資料のバックアップの計画は、やや高い見識を必要とします。公的な費用を期待するときは、法律による縛りが必要ですが、よいビジネスモデルの有る方が長続きします。多数決で計画を決める場面では、価値観が庶民レベルで判断されますので、否定的な扱いを受けやすい面があり、計画が中止されることもあるからです。戦争では、美術品の略奪が行われるのが普通です。この場合、略奪されても、実体またはレプリカが別の個所で保存されていれば、長い眼で見れば救いがあると考えられます。

2. 文字処理と英語用語の理解

2.1 JIS 漢字コード系制定までの経緯

2.1.1 文字処理は基礎技術であること

アメリカで開発されたコンピュータを日本に輸入して利用していた 1960 年代、最大の障害は、日本語の扱いが不便であったことです。表音文字のアルファベットを使う欧米語は、文字種(character set)として 128 種類あれば済みます。日本の環境では、それに追加する形で、少なくとも 3000 の表意文字の漢字が扱えなければならないからです。商用活動にコンピュータを利用したいとすることの全体を、EDP (electronic data processing) と言います。企業における事務処理の効率化を図るために導入されたコンピュータシステムを言います。日本語では電子データ処理システムと当て、主に給与計算、会計記帳、売上集計などを目的としました。この時代、大型コンピュータは高価な施設でしたので、資本の大きな、例えば銀行業などでないと、排他的な利用ができませんでした。銀行業務は、顧客名の管理に漢字を使う必要があります。アメリカ製のコンピュータの利用では、差し当たって、英字の部分にカナ文字を使うことから事務処理の合理化が出發しました。数値計算を主な利用目的としていた理工系の分野では、特に漢字を使わなくても済みます。データベースは文字処理を基礎技術とします。数値計算にコンピュータを利用する場合は日本語を使わなくても済みますので、漢字を含めた文字処理技術まで、手を伸ばすことをしませんでした。

2.1.2 オフィスオートメーション

一般企業は、事務の合理化が現実的な要求です。電気機械装置を含め、種々の事務機械が使われてきました。1970 年頃から、広義の合理化をオフィスオートメーションと言いました。日本の事業所必須の設備は、邦文タイプライタと、通信手段として TELEX が普及しました。これに続いて、電子化機器として、ファクシミリ、コピー機械、そして小型コンピュータの利用へと発展してきました。電話が一過性の通信手段であるのに対して、TELEX は、通信記録が残ることが重要です。しかし、電報文のような、英数字とカナ文字しか利用できませんでした。大型コンピュータよりも低価格の、中型・小型のコンピュータの開発も進められていましたが、一般の事業所にはまだ高価すぎる設備でした。国産の大型計算機をオンラインで共同利用する一つの形態として、TELEX の利用を発展させた電電公社の DEMOS が 1971 年に運用を開始し、パソコンが使い易くなるまでの一時代を画しました。16 ビットパソコン NEC-PC98 は 1982 年の発表です。これは中型計算機と同じ使い方ができますので、科学技術計算だけでなく、一般事業所の OA 化を促進させる原動力になりました。このパソコンが愛用された大きな理由の一つは、ドットマトリックスプリンタ (dot-matrix printer) で漢字の印刷ができることにありました。

2.1.3 難産であった JIS 漢字コード系の制定

コンピュータ処理に向くような統一した JIS 漢字コードの最初の制定は 1978 年です。その前、情報処理学会は 1969 年から始めて、足かけ 10 年の長い期間、各方面の意見調整を行いました。最初の日本語 ワードプロセッサ (ワープロ : word processor) 東芝の JW-10 の発表が、同じく 1978 年です。この印刷機械は、ドットマトリックスプリンタが使われました。なお、この機械は、新聞社のような大量高速印刷をする装置には向きません。レーザプリンタもそうです。漢字の利用については、ワープロメーカー毎に、異なるコード系が使われていました。このことは、文字データの互換性が取れなくて、いわゆる文字化けと言う障害となっていました。また、JIS のコード系を利用することにも根強い反感があります。その理由の一つは、特に、人名・地名などの固有名詞に使う漢字が、微妙に異なる書体や読みを持たせて使っていることに、標準化文字が対応できないからです。しかし、文字を日常的に扱う印刷・出版業界、とりわけ、一日 24 時間と言う時間制限の中で、組み版原稿を作成し、大量で高速の印刷をしなければならない新聞業界では、歴史のある伝統的な鉛活字を使う印刷方式を、コンピュータを利用して近代化することへの取り組みが必要でした。この課題は、1950 年代のアメリカの大手新聞社でも同じでした。この解決方法の研究は、コンピュータを科学技術の計算だけに応用するのではなく、多くの分野の研究を総合すると言う、広い視点が必要です。大学の研究者の協力も必要です。ここに、軍事研究であった オペレーションズリサーチ (OR : 米語 Operations Research、英語 Operational research) そして、実践技術としての システムズエンジニアリング (systems engineering : SE) が注目されるようになりました。データベースは文字処理を踏まえます。その説明に入る前に、解説しておきたい項目が複雑に絡んでいます。OR と SE の説明に先立って、まず、印刷技術について解説します。

2.2 新聞業界での苦闘

2.2.1 鉛の活字を使う印刷術の機械化から

印刷技術は、歴史の古いものです。1960年代までは、アメリカでも日本でも、伝統的に鉛の活字を一字単位で選んで原版を組み上げていました。新聞社を含む印刷出版関係の業界では、この組み版作業を効率化することに努力してきました。新聞の印刷は、一日単位の短い時間制限の中で、大量の活字を組み上げ、高速で大量の印刷をしなければなりません。一字単位で活字を拾うことを自動化して、文章単位の文字列の活字を出力する自動活字鑄造機が使われました。その文字情報の入力は、英語の環境では、電報の送受信に使う、紙テープ制御のテレタイプでした。日本では1950年代末に紙テープ制御の漢字テレタイプが開発され、新聞原稿を電信回線で送受信すると共に、自動活字鑄造機への入力に使いました。このテープの漢字コード系が、JISの漢字コード系の母体です。鉛の活字を組んだ原版をそのまま印刷作業に使うのでは、高速大量印刷ができません。原版から紙型を取り、それから輪転機用の版を鑄造して印刷しました。したがって、印刷作業の現場には、鉛活字の鑄造工場がありました。元の原版を使う印刷は、校正刷りに使いました。一般書籍の編集・印刷・出版を扱う企業は、改訂版の出版に備えるときは、この鉛原版を保存しますので、別に、倉庫のような広い保存場所を必要としました。新聞社では、その日限りの原版を利用し、記事の訂正や差し替えなどに使います。その日の最終版を公式記録としてマイクロフィルムに撮影して保存し、新聞の縮刷版としても出版しています。ここまでの機械化では、新聞記事を電子化文書の形では保存できませんでした。

2.2.2 写真製版の利用

鉛の活字を使う版の作成は、見出しなどに使う寸法違いの活字の一揃い（フォント：font）も用意しなければなりません。文字は図形です。組み版と言うのは、或る用紙面積を活字図形で埋める作業です。一般的な図形は、別作業で版を作成して挿入します。そうであるならば、文字も写真処理で版を作成することができます。これを写真植字（写植：phototypesetting）と言います。世界に先駆けて1925年に日本で実用が始まった方法です。寸法違いの字形であっても、一つのネガ図形を拡大・縮小操作で済ませることができます。この場合の校正作業は、文字並びを焼き付けた印画紙の、切り貼り処理が必要でした。一ページ単位で完成した原稿をカメラレディと言い、これから印刷作業用の版を写真処理で作ります。写真製版は、利点もあるのですが、短時間に校正まで済ます製版速度が重要な新聞紙面では、ワープロの開発とDTP技術が実用化に入るまでは、作業に組み込むことができませんでした。

2.2.3 文字処理の変革期の話

アメリカでも日本でも、新聞業界は、鉛の活字を並べる組み版を柱とした人海戦術で作業を進めていたシステムでした。1946年をピークとして年々経営が悪化していき、1950年代の半ばには、大手の新聞社である高級紙のニューヨーク・タイムズさえも利益の減少に悩まされるようになりました。この危機を乗り切るために、ORの研究が始まりました。設備などの合理化も図られてきましたが、ORの研究対象は、軍隊の制御と同じように、人のシステムの作業効率を改善することにあつたからです。日本の新聞社でも事情は同じですが、漢字を効率的に扱うことに大きなハンディキャップがありました。ワードプロセッサの開発研究は、新聞紙面1ページをモニタできることを目的としました。現行のパソコンのワープロは、高い解像度のグラフィックスモニタを使うことで、編集と割り付けにWYSIWYG（What You See is What You Get：画面見たままの印刷ができる）が利用できるようになりました。そうであっても、モニタの画面は、A4寸法の書類1ページ全体を原寸で見ることにはできません。新聞の場合、1ページはA2の寸法ですので、全体の割り付けを観察できるように、寸法の大きなモニタを利用し、スクロール機能と拡大・縮小表示をしながらページ単位で原稿を編集します。このコンピュータ化が実用の段階になって、ようやく、新聞原稿の電子化保存と利用に眼が向くようになりました。データベースの利用研究は、アメリカ発の新聞・雑誌の情報検索サービスの先行に、日本の多くの新聞社が飛びついたので始まりでした。しかし、情報検索は、あらかじめ、過去のデータの後始末的な作業が必要であつて、眼に見えるような業績に繋がらないので、各社とも急速に熱が冷めてしまいました。それを尻目に、地道な実力を蓄えていたのが日本経済新聞社でした。気が付いてみると、圧倒的なデータ処理能力で他社と大きく水を空けて独走体制に成長したのです。その実力は、各種の「日経…」の表題を持つ多様な出版活動に反映しています。

2.3 英語用語の使われ方

2.3.1 用語の意味と使い方に注意が必要であること

コンピュータ技術はアメリカ主導型で開発されてきましたので、用語の大部分は、元がアメリカ英語です。市販のコンピュータ関係の参考書を見ると、英単語を日本語訳にしないで、そのままカタカナ読みにしたものが多く見られます。そのカタカナ語をさらに詰めた使い方もします。例えば、パーソナルコンピュータ (personal computer) は、パソコンと使うのが普通になりました。英単語を併記すれば、未だ手掛かりがありますが、PC などのように頭文字だけで構成した頭字語(acronym)が厄介です。ありきたりの日常使う単語も、専門によっては特別な定義を持たせて使うことがあります。PC などの頭字語は、英米人にも説明が必要です。これらは専門用語ですので、辞書(dictionary)または事典にまとめます。専門用語は学問的な意義で理解されますが、普通に使う日常用語ではなく、言わば業界用語、職人言葉、方言(dialect)、悪く言えば隠語(slang)と同列です。したがって、その専門ごとに意味や使い方を限定したものを、用語解説(glossary)にまとめます。専門用語には、標準の語彙とは違うスペルもあります。アメリカ英語から見れば、イギリス英語は方言(dialect)扱いです。同じ意義であっても、専門違いで別の用語を使うことがあります。コンピュータ関係で使う用語は、3種類を使い分けます。

(1) 人に説明するときに使う専門用語：

コンピュータのハードウェア、ソフトウェアの説明書(マニュアル)では、始めに、そこで使われている専門用語の意味を約束します。ただし、既知と考える用語が抜けることがあります。また、日常会話で普通に使われている単語であっても、意味を限定していることがあります。これらは専門用語ではないのですが、日本語の環境では特別な用語になることがあるのが厄介です。これらの用語は、元の英語と共に、自国の言語に翻訳した用語を決めます。適当な日本語訳ができない時、英語読みをカタカナ語にします。これらは、自国の言語ごとに、解説辞書(glossary)、用語辞書が作られます。

(2) コンピュータを人に見立て(擬人化し)、コンピュータ本人が理解する言語：

この代表がプログラミング言語です。こちらは、正確に元のスペルで使わなければなりません。これを予約語(reserved word)と言います。これも英語の方言の性質がありますので、日本人にとっては厄介な代物です。データベースの方では、キーワード(keyword)と言います。定義が決まったキーワードは、分類を決めて索引などに利用します。キーワードの辞書がシソーラス(thesaurus：専門用語辞典)です。分類法それ自体も、一つの学問分野を構成しています。アイコンとマウスを使うインタフェースは、言葉が分からなくてもコンピュータが使える方法です。これがGUI(Graphical User Interface)です。文字を入力することでコンピュータに知らせる方式は、GUIと区別するため、後からCUI(Character User Interface)の用語ができました。これらを、Man-Machine-Interfaceとも言います。事務処理では女性が多く働きますので、MANを使わない User Interfaceの方が、抵抗がないようです。

(3) ユーザが、自分で決めて使う用語：

代表的なものが、変数名です。これも、コンピュータに、それと理解させなければなりません。プログラミング言語では、変数名を使うとき、プログラミング文の最初の方で型名と共に宣言しなければなりません。これは面倒な儀式です。初期のFortranやBasicでは、暗黙(デフォルト)の型定義の変数名を宣言無しに使うことができました。これは、代数式を書くときには普通の方法です。プログラミング単位が小さい時には、この約束がユーザにとっては便利です。しかし、コンピュータ側から言えば、判断に困ることが起こり、計算エラーの原因になります。最近のプログラミング言語では、これを避けるため、変数や定数の、定義と宣言の儀式が複雑になりました。

この他に、コンピュータを使う側の専門ごとに、固有の用語があります。なるべく統一した用語にしたいとする努力は、専門ごとの団体や学術的な学会で試みられていて、JISとして標準化も提案されます。マニュアルの作成は、上のことを踏まえますので、一種の規則書、法律文書の形になります。そのため、初心者教育には、入門書(introduction)、ガイドブックなどを別に工夫しなければなりません。プライマー(primer)の用語も見ることがあります。JIS本体は、目次も索引も作成しませんので、教育用の参考書としては使い難い文書です。

2.4 システムズエンジニアリングの理解

2.4.1 システムとは何か

データベースの話から脱線するように見えますが、コンピュータを利用することと密接に関連のあるシステムズエンジニアリング (systems engineering: **SE**) の解説をしておきます。英語の system は、カタカナ語のシステムとして国語辞典にも載るようになりました。和語には抽象的な概念を表す言葉が少ないので、辞書には漢字の熟語で組織・制度・体系などと当てています。システムとは、幾つかの要素を集めて、或る機能を達成するように構成したものを言い、個別の要素の単純集合ではない、一段階上の概念を持って使います。英語では複数形を持つ普通名詞ですが、単数形でも集合名詞の意義があります。単数形は、工学的には、電気・機械装置などの単体を言うときに使います。人の集合単位にも言うことに注意します。軍隊がその一例です。何とか組合もそうです。そこには階級制度があって、個人の勝手な行動が制限されます。コンピュータに関連する技術にもシステムの用語が眼に付きます。代表的なものが **オペレーティングシステム** (Operating System: OS) です。実体はコンピュータプログラムの一つです。システムの用語が多様に使われますので、複数のシステムの集合で構成される総合には、何かの固有の名称をつけて、単数形の何とかシステムの言い方を見ます。ここで、複数のシステムの集合を合理的に機能させるための、計画・設計・管理の技法が必要です。これを **システムズエンジニアリング** と言い、システムを複数形で使うことに注意します。コンピュータ全体システムの管理をする人を、日本語では **システムエンジニア** と言いますが、元の英語は systems engineer と複数形で使います。ここでのシステムは、ハードウェア、ソフトウェア、それに人の組織を含む、複数のシステムを扱うことを意味します。

2.4.2 オペレーションズリサーチとセットで使う

軍隊は、戦争遂行するためのシステム (組織) です。負ければおしまいですので、**過去**の戦争経緯を調べる研究をし、相手の作戦計画を調べて対応を研究します。これが **オペレーションズリサーチ** です。作戦研究と訳しています。こちらも、オペレーションズ (アメリカ英語) と複数形です。また、エンジニアリングではなく、リサーチと使うことに注意します。軍隊の戦争遂行は、システム全体を勝つ方向に進める意思決定が必要です。これは、何が起こるか分からないことを含みにした **未来** 志向の、危険を伴う設計です。過去の事例を参考にはしますが、前例にとらわれない生臭い決断が必要です。これがシステムズエンジニアリングであり、research や study ではなく engineering を使う真意です。学問としての研究態度は、過去の事例を扱うことが普通ですので、優雅に構えていることができます。OR も SE も、1950 年代半ばからコンピュータを経営戦略に応用することを目的として一般的に使われ始めた用語です。これらの研究は、主にビジネスの分野で注目されました。科学技術への応用は、1961 年に始まったアメリカのアポロ計画を支える実践的な技術として有名になりました。

2.4.3 システム工学と訳したことによる誤解

日本でも、システムズエンジニアリングを、アメリカに学び始めました。筆者の持っている古い参考書は、1961 年、日本能率協会 EDP 委員会が訳して発行した「OR とシステムズエンジニアリング」です。**EDP** は、汎用コンピュータを経営の合理化に応用する意義を持った実践技術を意味します。実務と距離を置いた大学などの研究者は、取り上げ難い内容を持っています。その中の個別の話題として、例えば、ゲームの理論など、理論と銘打った話題については、理工系の研究者が取り上げていました。システムズエンジニアリングを「システム工学」と言い換えて研究対象に取り上げたのですが、単数形のシステムと使い、「学」の字を使ったことが誤解の始まりです。「学」というのは、「**真似ぶ**」の転化した語と言われ、**過去**の事例を研究する意義が強いのです。これから実行する何かの **未来** 計画は、過去の経験を参考にはしますが、意思決定はシステムのリーダーが下さなければなりません。失敗する危険もありますので、システムズエンジニアリングは **危機管理** (risk management) ともセットです。危機管理も生臭い決断を伴います。過去の失敗例を取り上げる研究を「**安全工学**」と言い替え、学者の格好の研究テーマにしています。複数の部局を個別のシステムと見れば、その総合としての企業全体はシステムズです。その最高責任者は、全体システムを、一つの進路に導くため、孤独な帝王学的な素養が要求されます。インテリは「歴史に学べ」と言いますが、視点が過去にありますので、現実的な提言をしません。実務者が採用する最も無難な提案方法は、合議を経た前例に倣うことです。設計作業は、**未来志向** です。未知の事象に挑戦するにしても、過去の事例を研究することも重要です。ここにデータベースの合理的な利用が必要になる、という筋書きに繋がります。

2.5 オブジェクト指向の言葉の意味

2.5.1 オブジェクトは理解し難い用語

コンピュータの参考書を見ると、オブジェクト(object)の用語が多く見られます。英語を母語とする環境では、特に抵抗なしに理解されていますが、その分、使う場面が広いので、日本語に訳したいとき、適切な言葉を当てるのが難しく、意識することが多くなります。直接には、普通名詞の「もの：物」と訳します。英語の環境では人や生物も含まれます。実体のない事柄も含まれます。漢字の物も、抽象的な意義で使い、これも原義は人も含まれます。人物、動物、生物と使うことを見ます。和語では、物と事とを使い分け、人の方に「者」を当てます。英語の文法書では、object を目的語と訳しますが、よい意識です。この目的語には人も物も事も含めます。文法的には、関係代名詞の先行詞で使います。抽象的に、対象と言うことがあります。そうすると、日本語の環境では、具体的に何を指しているかが分からないことが起こります。その代表的な使い方が、形容詞的に使われる“object-oriented…”です。ハイフンで繋いだ表現であることに注意します。よく分からない用語ですので、orientedの部分を指向と当て、残りをカタカナ語のままにした「オブジェクト指向…」の使い方が氾濫するようになりました。

2.5.2 先ずオブジェクト指向プログラミングから

パソコンは、計算機械の概念を越えて、一種の擬似的な装置(シミュレータ：simulator)の使い方が普通になりました。モニタの画面上に、擬似的な、ボタンなどの図柄を表示して、それをマウスでクリックするなどして、擬似的に操作します。この図柄をオブジェクトと言います。プログラミングの説明ではコンポーネントなどと言い、さらに具体的には、フォーム、ボタン、ラベルなどの普通名詞が当てられています。プログラミングの中では、個別に固有名詞的な名前を付けて区別します。これらのオブジェクトに機能を持たせて、ユーザとの対話に使うプログラミングがオブジェクト指向プログラミングです。教育用の例題には、電卓を作成するプログラミングが、よく紹介されます。プログラミング言語の Visual Basic (VB) は、何度もバージョンが改訂されてきました。VB6 までの版では、従来の Basic 流のコーディング記述をしたソースファイルを標準モジュールと言います。昔に作成した Basic のプログラムコードや、Fortran や C などのソースコードも、多少の書き換えをすれば標準モジュールとして使うことができます。これとは別に、オブジェクトを扱うプログラミングコードを記述するフォームモジュールを作成します。この区別をしておく方が分かり易いでしょう。オブジェクトに具体的な部品の図形を扱うので、その部品単位でプログラミングを作成しなければなりません。部品の種類が多くなり、部品ごとのプログラミングを集めて追加していくとページ数が増えます。相対的に、普通の計算やリストの作成などが主体の標準モジュールの説明が簡単になるか、省かれることが多くなり、初心者の勉強には不便になっています。両方を含めたモジュール単位をクラスと言うのです。

2.5.3 オブジェクト指向データベースとは

「オブジェクト指向…」の…部分にデータベースと使った“object-oriented database”と、単純に“object database”と表現した言葉の区別が分かるでしょうか？ 後者の言い方は、幾つかあるデータベースがあって、いま話題にしているデータベースを指します。同じ使い方は、プログラミング言語 COBOL の予約語(reserved word)に object computer があります。そのプログラムを使うコンピュータの機種を特定したいときに使います。また object code の用語もあります。人が読んで分かるプログラミングコードを source code と言うのに対して、コンピュータの CPU (演算処理装置) が理解できるバイナリーコードを言います。CPU が違えば読めません。“object-oriented database”は、object に具体的な対象を指定する必要があります。一般的には、テキスト形式のデータに加えて、音響データ、ビデオ、グラフィックス、さらにはプログラミングコードそのもの、なども考えます。より具体的な例は、画像データベースがあります。これは、デジタルカメラの普及によって、一般の人でも、大量の図形情報を扱うことが増えたことで、話題として多く現れるようになりました。この解説は後の第5章で取り上げます。グラフィックス関係では、“object-oriented graphics” (オブジェクト指向グラフィックス) の用語があります。これは、工業製図の作成の時に必要となる技術です。工業製図は、基本的に線図 (line drawings) で描きます。線の種類としては、実線、破線、点線などがあり、線の太さも何種類かを使い分けます。この線分をオブジェクトとします。線種の区別を表す性質を属性 (プロパティ: property)、線を描かせる操作は動詞の命令形を意味するメソッド (method) と言い換え、これらの用語を使うプログラミング単位がフォームモジュールです。複数のオブジェクトを組み合わせ、何かの図形を描くとしたのが、オブジェクト指向グラフィックスです。このようにしておくとも図形情報をファイルに保存し、図形の変更などができます。この方法が CAD のソフトウェアに応用されています。

2.6 Structure の意味

2.6.1 Structured programming

コンピュータのプログラミング言語は、1950年代には、数値計算用に FORTRAN、事務処理用に COBOL、少し遅れて、コンピュータ寄りの C 言語の三つが主流でした。FORTRAN は、数値計算を指向（目的と）していて、John Backus が 1954-58 年に掛けて開発しました。一方 COBOL は、委員会方式の CODASYL (Conference on Data Systems Languages) が 1959-61 年にまとめた、ビジネス利用を指向した言語です。COBOL は、Common Business-Oriented Language の頭字語です。ビジネス指向を標榜していることに注意します。また、COBOL の用語説明の辞書には、English-like Structured Language とあります。日本語で言えば、英語風の構文のプログラミング言語です。科学技術の計算利用とは少し距離のある、一般ユーザに理解できることを意識したものです。ここで、構文と言う日本語を使いましたが、これに当たる英語が language structure であって、structure を使うことに注意します。この英単語は、構造物や建造物の意味に理解されますので、構文として「構」の漢字が使われることには、直ぐには理解が及ばないのです。動詞としての使い方も、日本語の感覚からは距離があります。過去分詞が形容詞として使われたのが structured programming です。この例は、データベース関係で SQL (Structured Query Language) があります。なお、C 言語は少し遅れて、1972 年 Dennis Ritchie の提案です。マシン語で動くコンピュータの機能を生かすことが目的ですので、コンピュータの専門家向きの言語です。

2.6.2 構造化プログラミング

英語の structured programming は、構造化プログラミングの訳を当てるのですが、上で挙げた「英語風文書に構成する」の言葉を補うと納得し易いでしょう。普通、文章は、頭から順に読んで理解できるようにまとめます。論文をまとめるとき、章・節・項、さらに段落 (paragraph) 構成を意識します。「後で説明します」「前に説明したように」のような作文は、一旦読みを中断して、そちらのページを探すこととなります。COBOL の、DIVISION, SECTION に分ける構文は、論文作文作法が意識されています。科学技術を扱う説明文書では、説明が前後することは或る程度、避けられない面があります。これを補うために、目次と索引を丁寧に作り、さらには参考文献名を添えます。筆者の経験から言うと、学校教育でプログラミングを教えるとき、お料理のレシピを作文することを例に使うと納得し易いところがあります。鍋・釜・食器類がハードウェア、材料がデータ、そして料理の手順がソフトウェア、つまりプログラムです。時間的に前後する手順で説明するのはよくありません。お料理は一人で取り仕切るので、複数の人が協力する、または鍋などを並べて同時処理をするのが並列プログラミングに当たることを理解してくれます。ただし、後の場合は、時間的に処理が重なると失敗の危険があります。具体的に構造化プログラミングに構成する一つ方法として、GO TO 文を使わない方法が提案されます。GO TO 文は、処理が飛んでいく先をラベル（見出し語）で指定する必要があります。教育用に工夫された初期のプログラミング言語の BASIC は、FORTRAN 流のプログラムの性格を持っています。マイコン (micro computer) に搭載された BASIC のインタプリタは、コード全体を一単位のモジュールに書きます。プログラム単位の寸法が大きくなり、GO TO 文を多用すると、処理の流れをプログラム文から追いかけることが難しくなります。これをスパゲッティコード (spaghetti code) と言い、反義語 (antonym) です。

2.6.3 スクリプト言語

COBOL は事務処理を目的にしますので、ファイルに記述されたデータの読み書きが重要です。このとき、或る特定のデータをファイルから検索する処理が必要です。これを別の言語体系として規格化しておいて、親言語としての COBOL を始め FORTRAN などの本体に埋め込んで使う方法で提案されました。これが SQL (structured query language) です。Structured の英語が使われている理由は、COBOL が英語風の構文であったのに合わせるスタイルを取ったからです。親言語ごとに、固有の構文スタイルがありますが、大きな書き換え無しに、追加する文として扱う言語をスクリプト言語 (scripting language) と言います。これは種々の応用プログラム (API: application programming interface) に見られます。VBScript (Visual Basic Scripting Edition), JavaScript などがそうです。親言語は、スクリプト言語を解釈するコンパイラを追加しますが、その言語のオブジェクトコードは、ライブラリ形式で参照します。SQL の最初の規格提案は 1986 年です。SQL は、データベースのファイル構造の仕様と関係します。最も単純なデータベースは、データ全体を一つの表でまとめたものです。検索効率・データの追加・変更に対応するため、互いに関連を持たせた複数の表を扱うようになりました。これが リレーショナルデータベース です。SQL に対する要望が多いので、規格の改訂が頻繁に行われてきました。

2.7 データベース作成の前段階

2.7.1 目録作成は書籍だけに限らないこと

目録とは、何かの複数の実体が別にあって、その大まかな構成内訳を書き出したものを言い、種々の場面で使われます。英語では、list, catalogue, inventory, index を使います。箇条書きの書式に作成することが多く、番号や分類など付け加えて表の形にし、見易くする工夫もします。書物単体の作成では、本文以外に目次と索引を作ることがその書物単位内の目録作成に当たります。複数の書籍や文献の目録は、データベースの原稿の性格があります。書籍以外の実体についても、目録作成からデータベース化をする方へも発展しています。具体的には何かの商品カタログの作成がそうです。カタログは、説明文章だけでは実体の情報が分かりませんので、写真や図を使い、実体本体、または実用にも使える程度のコピーが得られるように、紹介する手段が必要です。これはパソコンのグラフィックス機能が便利になったことで可能になってきました。これがオブジェクト指向データベースに繋がります。この場合でも、内容説明の文章は必要ですので、標準的なデータベース作成に、グラフィックス情報を追加する形になります。

2.7.2 文献調査とガイダンス

何かの問題を解決したいと計画するとき、過去にどのような研究があったかを調べることも、一つの実践的な研究分野を構成します。これが文献調査です。この成果は、目録の形にまとめます。そうすると、これがデータベースの原稿になる場合と、逆にデータベースから目録原稿を作成する場合とがあります。学術論文をまとめるときの定型は、introduction (序論) から始め、そこに研究の位置づけと、既知と未知の問題の説明をします。そのときに参考文献を挙げるのが丁寧です。これは、扱っている問題に関連を持つように選択をします。どの範囲まで紹介するかを絞ることの判断は、難しいところがあります。この部分は、多分に教育的な面があります。論文が扱っている専門に詳しい人にとっては、冗長に感じられても、若い研究者にも理解してもらうためには、説明が丁寧であるのが望ましいのです。一般に、学術論文を受け付けるときは、審査委員の査読があります。研究課題の解説や紹介的な報文は受け付けません。しかも、参考文献を多く書いてあると、論文の中で参照していなくても、審査委員が「よく勉強している」と好意的に捉えることもあります。逆に、非常に独創的な研究では、参考文献が無いか少ないこともあります。そうすると、参考文献が無いと言う理由だけで不採択になることがあります。したがって、定評のある教育目的の参考書が有って、それが良く知られていれば、かなりの記述を省くことができます。専門の学術機関では、教育目的に使う参考書を編集するか、別に出版された定評のある良い図書を推薦するガイダンス活動が必要です。

2.7.3 実体を見る手段にマイクロフィルムを使う

学術論文に引用する論文は、著者名・論文名・発行元の情報だけです。中身の簡単な紹介(抄録)があれば、一つ一つ原典を見る手間が省けます。論文発行機関では、抄録を集めた抄録誌を発行していました。古い論文では抄録を付ける習慣がなかったからです。最近の学術誌は、著者の方で抄録をまとめ、さらに分類に使うキーワードを付けるように投稿規定が決まっています。この抄録をデータベースの材料に使います。しかし、論文本文を見たいとしても、身近の図書館に原本がないとき、別の図書館に依頼してコピーしてもらいます。コピーを依頼される側では余分な仕事ですので、それを引き受けるビジネスがあると便利です。アメリカでは、**University Microfilms International (UMI)**がマイクロフィルムサービスを引き受けています。この手段の一つに、1960年代、フランスで開発されたマイクロフィッシュ(図 2.1)が利用されました。葉書き大のマイクロフィルムに最大で60ページ分を収めることができます。マイクロフィッシュからマイクロフィッシュの複写コピーを作ってユーザに郵送します。ユーザ側では、簡単な光学的ビューアを使って読むか、自分の費用でハードコピーを作ります。カードのような保存と利用ができます。参考文献に学位論文が引用されていて、図書館で現物が見られない場合にも対応できます。マイクロフィッシュの利用法として、何かの特定した研究課題に関係した論文やデータだけを選択して、マイクロフィッシュの形で研究者に配布する使い方もします。1960年代には、かなり普及したのですが、後始末的なデータ管理の道具ですので、日本ではこれも熱が冷めてしまいました。UMIは、電子化文書でのサービスをするように進化しています。電子化文書の利用は時代の流れでしょうが、印刷形式、つまりハードコピー化した実体を直接目で見て調べる手段が失われると、資料管理で大きな危険をはらむことが危惧されるようになっていきます。

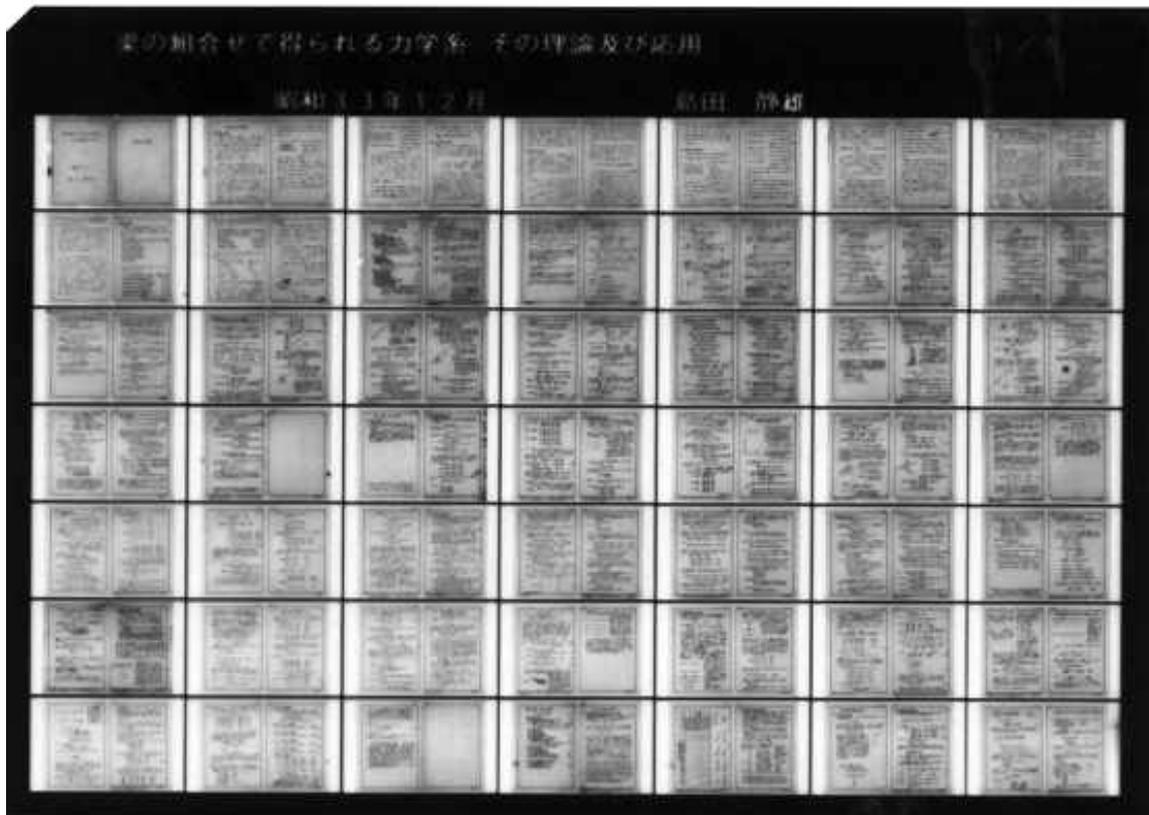


図 2.1 葉書き大のマイクロフィッシュ。スキャナでもページ単位を読むことができます。

2.7.4 メモ管理もデータベースと言う誤解

データベースに真面目に取り組むと、コンピュータシステム全体について、相当に深い知識と経験とが必要です。個人がデータベースを構築して利用すると言うのは、例えば、名簿管理をして、年賀状の宛名書きに利用する、その程度が限界です。この目的に使うパソコンのソフトウェアは、EXCEL が便利になりました。ACCESS は、本格的なリレーショナルデータベースが Windows 系のパソコンで利用できるツールです。しかし、一般の人が扱いこなすのは難しいでしょう。このこともあって、名簿管理などは、ベンチャーソフトが開発したソフトウェアで済ます人が多いのです。本格的なデータベースは、多くの人が、互いに独立していても、通信回線で繋がった環境で共同利用できるシステムです。専門的な話題について個人が自前のデータベースを作成するとしても、そのデータ量は知れています。しかし、他の人にも利用してもらい、また、他の人のデータも利用したいとなると、共同利用できるデータベースセンターを育てる方向に進むのが望ましい解決です。データベースの維持管理は、ハードウェアとソフトウェアとを含めたシステムが必要ですし、専門的な知識を持つシステムズエンジニアが専任しなければなりません。これには相当の費用が掛かりますので、現在のところ、ビジネスモデルとして成功した例は多くないようです。インターネットが便利に利用できるようになりましたので、この解決の動きも見られるようになりました。それには、個人が持つデータをセンターに集めて、データ量全体を増やします。このとき、各個人が、データベースの知識を共有する事前の勉強を踏まえる必要があります。

3. 目録の作成からデータベースへ

3.1 文献調査のまとめが最初の目的

3.1.1 欧米文献を有り難がること

明治以降、日本は海外、それも欧米の文化や科学技術を学んで、近代化を急ぎました。欧米の文物を理解するには、欧米言語で書かれた図書や文献が読めなければなりません。欧米人との直接交流の機会は少なかったので、会話は重要視されず、もっぱら、和訳の能力を高めることで、知識の吸収をしました。外国語を理解できることが、上に立つ人の教養として必須になりました。実践的な科学技術を効率よく育てるには、既に欧米で実績にある事物をそのまま輸入し、次いで、それを真似て（学んで）国産化する方法を取ります。海外文献に載る内容は、多くの実践技術の経験や実験などを踏まえた上で得られた知見です。実践技術の現場から離れた象牙の塔の研究者は、海外文献を翻訳して前例を紹介するだけで尊敬を受けました。明治から昭和の半ばの敗戦まで、日本の科学技術は、表面的な発展にうぬぼれも見られました。それが太平洋戦争の敗戦で打ち砕かれ、再び海外文献に学ぶ研究態度が復活しました。前例のない計画に、自前で研究開発を育てることに眼が向くようになったのは、1960年代、新幹線の建設技術の頃からと言えるでしょう。これを支えたのは、戦時中の旧海軍の技術者集団を、旧国鉄が引き取ったことが大きな戦力になりました。

3.1.2 本州四国連絡橋の計画時の文献調査

神戸市の市長であった原口忠次郎(1989-1976)は、神戸市と淡路島とを結ぶ吊橋の計画を、1959年(昭和34)年に政治主導で発表し、その目的のための、文献調査の部署を発足させました。実際に架設が完成したのは、1998年です。足かけ30年も掛かった理由は、本州と四国とを結ぶ橋と言う、国としての計画に、様々な政治的な駆け引きが行われたこともあります。純粋に架橋技術の面から見て、多くの調査が必要でもありました。吊橋の設計で最も恐れられるのが、タコマ吊橋を落橋させた原因となった風の作用です。日本では、それに加えて、地震対策も重要な要素です。この二つに絡んで、多方面の調査が行われました。調査研究は、土木学会で多くの研究者を集めた委員会方式で討議されました。海外の長大橋の建設は、設計者の主導でおこなわれたことと、大きな違いがあります。このとき、重要と考えられた論文を撰んで、委員にコピーが配られ、また、論文の抄録集を別に作りました。この研究の進め方は、同じ学術的な問題に複数の研究者が情報を共有する、一見 民主的な方式です。1960年代は、まだワープロも、またデータベースの技術もありませんでした。情報処理は手探りで研究されました。委員会の資料は、現在の感覚ならば、情報公開の対象として保存するアーカイブです。当時は、委員を中心とした、限られた人数の中での公開でした。論文コピーを共有する方法として、マイクロフィッシュの利用も試されました。抄録集は、手書きの原稿を青写真でコピーを取って委員に配布しましたが、幾らかの費用をかけて軽印刷でまとめることもしました。この委員会方式で試行した方法は、基本的に、その後のデータベースの開発と利用方法と同じでした。ただし、後者の場合、利用者は互いに独立し、閉鎖的であって、委員会でのような討議の場がないことです。したがって、データベースの作成は、教育と研究に利用して役立てる、公共的な方法を紹介する広報活動が望まれます。これを、民間のビジネスモデルの構築に繋げたいところです。

3.1.3 委員会方式の研究活動が犯し易い欠点

本州四国連絡橋は、政治主導で計画されましたので、結果的に3ルートが提案され、建設されました。純粋に技術的な立場に立つと、いきなり世界最長の明石海峡の建設に挑戦することの前に、段階的に経験を積むことが望まれます。その意味で、3ルートに種々の形式の長大橋が建設されたことは価値がありました。実を言うと、政治主導の建設計画は、欠点として、不都合なデータを公表しないか、無視することがあることです。実は、明石海峡の架設ルートには断層地帯があることが知られていましたので、研究者や技術者の本音としては、建設に乗り気ではありませんでした。したがって、本四架橋の殆どが完成した時点で、最後の長大橋として工事が始まりました。その建設工事の最盛期の1995年に阪神・淡路大地震が襲ったのです。吊橋の主塔間隔が約1mずれたのですが、幸いなことに全体建設に大きな障害にはなりません。原子力発電所の建設計画においても、政治主導の委員会形式の技術委員会は、素人委員の主観的な判断も入る多数決ですので、透明で公平な結論を曲げることがあったことが、明るみになりました。このことを考えると、データベースの利用が広く一般化し、データの客観的な評価を可能にすることは意義があります。

3.2 知的好奇心を満たす目的に使う

3.2.1 本体にアクセスするのが基本的な態度

データベースを作成する目的は、図書館においては、保存してある図書の実物を探して書庫から取り出すこと（アクセス）にあります。閉架式図書館での古典的な方法は、個別の利用者が、カードなどの目録を見て、司書に依頼して書籍本体を書庫から取り出してもらいます。カード探索に代わって、コンピュータで検索できるようにしたのが、図書館で利用するデータベースです。探したい書籍を絞り込むための情報に使うのがキーワードです。このとき、内容についての簡単な紹介（抄録など）も見られると無駄が省けます。データベース化は、ページ数の少ない論文の検索もできて、さらに通信回線で繋がった研究者の端末で利用できるように進化してきました。論文本文をモニターで見られるように電子化する努力もされてきました。著作権のある書物の形式であっても、電子書籍化が始まりました。このようにすると、書籍本体を持ちだすことによって、傷める、などの事故に対処することができます。しかし、図書館での対象は、複製数で印刷された書籍を主に考えています。複製の少ない貴重な書籍は、博物館や美術館並みの厳重な保存と管理が必要ですが、目録は公開します。こちらを、データベースに登録すると、**知的好奇心**で所在情報を知りたいときに使うことができます。

3.2.2 本体に代わる情報で妥協する

人も動物です。動物は本能的に知的好奇心を持っています。動物は、実物に当たることで経験を増やします。人の場合には、情報を入手して知識を増やすことにも積極的です。典型的な例が美術品です。例えば、絵画の本物は一点しかなく、それを保存してある美術館が遠くにあれば、実物を見る機会が殆どないのです。しかし、その情報は、美術書の中の複製を見て広く知られています。複製は、厳密に考えれば著作権を侵害しているのですが、多くの複製が広まって有名になれば、元の作品の評判が上がり、そこにビジネスが発生します。実物を直接見ることができる展示会があれば、出かけて行きます。平和な時代になると旅行がブームになって、有名観光地、つまり 情報として知っていた場所、に出かけて、直接 眼で見ると満足感が得られます。しかし、それが簡単にはできないことの方が普通です。一般大衆がインターネットをデータベース的に利用する目的の大部分は、情報を知って、知的好奇心を満たすための妥協です。しかし、下世話な噂話を扱うことも少なくありません。

3.2.3 データベースの作成前にファイル管理から入る

手を掛けて集めた情報を、趣味として個人が楽しむのは罪がありません。仕事の上で資料を探すときは、第三者が再利用できる形に整理しなければなりません。私的な著作や論文の作成の資料として情報を集めるときは、誉められた話ではないのですが、その情報元を隠すこともあります。学者が雑誌などに寄稿を依頼されるときは、ページ数などの制限がありますので、悪意はないのですが、論拠の出典を紹介する余裕がなくて、結果として美味しそうな部分だけのつまみ食いの説明に終わることがあります。研究者としての態度は、引用した論文や、資料のコピーを手許に保存しておくのですが、これが面倒な整理作業を必要とします。この作業を個人の私的な利用目的だけに使うだけでなく、後継の研究者も利用できるように、公的な配慮を考える視点が望まれます。成果が認められる研究活動は、個人の能力が大きな要素であっても、複数の研究研究者の協力が必要になっています。そうであると、データベースを作成する一段階前に、データの共同利用ができるファイルの管理法と、それを後継者に引き継ぎができるような習慣も育てなければなりません。これを広い意味で教育問題と捉えます。

3.2.4 実物に当たる態度を育てる

筆者の専門は橋梁工学です。橋は地域のシンボルマークになりますので、一般の人を含めて、知的好奇心の対象です。製作・架設の現地に出かけることは、いつも可能ではありません。写真、紙に書かれた資料、などの情報を、インターネットを介した検索サービスで収集することが便利になりました。橋の専門家ならば、専門のフィルタを通して利用する視点が必要ですが、専門家向けの資料は多くありません。パソコンの画面を見るだけの勉強法に加えて、専門教育の立場からは、実物を眼で見、手で触る、関連する実験、などが欠かせません。橋の調査に行っても、どうして良いかが分からなくて、野次馬と同じようにボーと見ているか、せいぜい写真を撮る程度です。それも、どこを撮影して良いかの判断もできない例が多くなりました。専門家が利用する写真の管理は、図面の管理と共に非常に重要ですので、このデータベース化の動きが実用化の時代を迎えています。こちらは、橋の観光案内的な資料のデータベース化とは違って当然です。

3.3 資料の静的管理と動的管理

3.3.1 静的なデータベースとその管理

一般の人は、データベースを作成するという意識なしに、私的に使う住所録などのデータを整理しています。これをパソコンで利用できるようにすると、例えば、年賀状などの宛名書きの手間が少なくなります。いわゆる名簿屋は、住所・氏名などの個人情報の集積である名簿のデータファイルを売買する裏商売を言います。これが、ダイレクトメール、迷惑電話、などに利用されています。考えようによっては、データの中から或る条件に適合する個人データを引き出して利用していますので、データベースの最も標準的な使い方をしています。しかし、容易に理解できることは、名簿内容は、追加・削除・訂正など、更新の管理が必要です。したがって、古い名簿は商品価値が下がります。一方、図書館の蔵書管理に使うデータベースは、蔵書の追加だけを管理します。一旦登録すれば、廃棄をしませんし、データ項目の変更も削除も考えません。このデータベースのファイル処理は、権限を持った管理者だけに許すように厳重な安全管理で保護され、一般のユーザがアクセスしても書き換えができないようにします。筆者はこれを**静的なデータベース**とすることにしています。

3.3.2 ユーザの利用で内容が変わるデータベース

互いに独立した複数のユーザがアクセスできて、データを共通に利用したいとします。そのデータ内容の変更を伴う形式のデータベースが、ビジネスで最も多く利用されるようになりました。商品の注文と在庫管理などに利用するデータベースがそうです。複数のユーザが見掛け上同時にアクセスしても、混乱しないように管理することが必要です。比較的単純なデータを扱う例として、新幹線や航空機の指定席の予約(ブッキング)を考えると理解し易いでしょう。このデータベースにアクセスする利用者は、個別のサービスセンターの端末を扱う窓口の掛です。互いに全く独立ですし、データベース本体がどこに在るかには頓着しません。運行単位ごとに、予約状況を保存するファイルが作成されますので、お客さんの希望を聞いて、日時、便番号、などをキーにして、空いている席の現状をまず確認します。希望者が殺到するときは、直ぐに予約をしないと、他の人に持って行かれますので、差し当たり、唾を付けるように保護をかけます。お客さんの確認を待って予約が確定します。確定の前後で取り消しが必要になることがあります。この手続きは厳重なセキュリティの手順があります。そうしないと、他の人の勝手な書き換えや、ダブルブッキングのエラーが起こります。この予約のデータは、空席か、そうでないかです。最も単純な(0,1)情報で区別ができます。データ内容は刻々と変化しますので、筆者は、これを**動的なデータベース**と区別することにしました。

3.3.3 個人が閉鎖的に使うのはデータファイル

データベースの定義の中で最も基本的なことは、独立した複数の利用者が共通ファイルにアクセスできることにあります。データを集めたファイルの複製を作って配布または販売し、それを個人が自分専用として使うものも、便宜的にデータベースと言います。見掛け上、一単位のデータファイルを複数の人が共有して使っています。しかし、このファイルは、データベースではなく、或る目的でデータベースの検索結果を利用した成果品ファイルです。元の内容が更新されても、自分用のファイルの更新は、自動的には行われません。自分で作業をして更新することは、実践的には不可能です。利用者の立場から言えば、自分の作業目的に必要なデータがあれば良いのです。殆ど使うことのないデータに、多くの保存媒体を個人的に準備したいのは人情ですが、無駄に繋がります。個人単位で作成するデータファイルにデータベースの名前を付けたい理由は、他の人にも利用してもらいたい、または研究成果としても評価してもらいたい、と言う世俗的な下心もあります。研究者が作成する、この種のデータファイルは、作成者の専門観によるフィルタを通した、項目の選択があります。

3.3.4 教育利用を意識する共用データベース

商業利用を目的とするデータベース化は、ビジネスモデルを提案することができます。しかし研究資料に使うデータファイルは、ビジネスモデルを考えた運営が難しい面があります。この種のデータファイルは、次世代の研究者に研究資産を引き継いでもらうとする教育利用を意識すれば、或る共用できる環境、例えば学術団体、にデータベースを置くことに、複数の研究者の協力が得られるはずです。個人が貢献できる資料管理は、たかが知れた量だからです。これらを総合する管理運営には費用が掛かります。公的な補助を仰ぐことを直ぐに考えたいのですが、自助努力を育てるようなビジネスモデルを模索する時代になりました。

3.4 計算機械の原理から

3.4.1 整数しか計算できないこと

パソコンは、道具として必需品になりました、大容量の内部・外部メモリが使えることが常識です。データベースへの利用を理解するには、計算機械開発の歴史について、一通り復習しておく必要があります。数値計算に使う道具は、基本的に整数しか扱いません。小数点を含む実数であっても、小数点位置を便宜的に決めていて、実際は整数計算をします。そろばん（算盤）を使うとき、桁並びのどこかを小数点位置に仮定するのがそうです。数値並びで、計算桁数の範囲の外に小数点を決めなければならないときは、元の数値の単位系を変えます。**電卓**（電子式卓上計算機）は、数の表示画面の桁数が、10桁または12桁が普通です。この桁数並びの範囲に小数点位置が無い数は、表示できません。二つの整数の掛け算をするとき、二数の有効桁数の和が、表示できる桁数を越えるときは、エラーになります。小数点を含む実数では、見掛け上、大きな桁数になる計算ができます。このときは、小数点位置が画面上に残るときはエラーになりません。整数部分の桁数が表示桁数を越えるときは**オーバーフロー（桁溢れ）**のエラーと言います。また、小数点数以下に0が並ぶ小さな数が計算されると、表示できる有効数字の数が減っていき（**桁落ち**）、それを越えると0しか表示しません。これを**アンダーフロー**と言います。少し高級な関数電卓は、**浮動小数点数**で表示してくれます。小数点位置が画面から外れる数は、小数点位置を左端に移し、補助的に10の何乗になるかの指数表示を添える表し方をします。

3.4.2 レジスタの寸法で最大整数の大きさが決まる。

標準的なそろばんは、計算場所である一続きの長い**レジスタ**（普通は23桁）を、論理的に幾つかの小単位のレジスタに分割し、部分的に数を保存する**メモリ**にも使います。電卓は、10桁または12桁の整数を扱う4つのレジスタがあって、その3つを使って掛け算（ $A \times B \rightarrow C$ ）の計算をします。4つ目は、メモリとして使います。コンピュータは、最小単位のビット数のレジスタが幾つかあって、それを繋いで桁数の大きな数を扱います。コンピュータの場合、標準寸法のレジスタのビット数で、8、16、32、64ビットのコンピュータ、のように言います。そうすると、一つのレジスタが扱うことができる正の整数の最大値があって、8ビットでは255まで、16ビットでは65535（通称で65K）までです。実用計算に桁数が足りないときは、レジスタを繋いで、論理的に一続きの長いレジスタに構成します。数値計算のときは、ハードウェアとして演算プロセッサを別に追加する方法を取ります。精度を上げる計算、例えば倍長計算などをさせるときは、ソフトウェア的に計算手順を組み立てることもします。

3.4.3 データに番号を割り当てるときの数の制限

個数の多いデータを扱うとき、コンピュータは、外部メモリからデータを読み出し、演算用のレジスタの他に、追加の内部メモリを使います。これを**RAM**（Random Access Memory）と言います。データ位置（**アドレス**）に番号を割り当てる、この番号の最大値が、レジスタが扱う数の最大値で制限を受けます。データベースは、データ数が非常に大きくなりますので、その対応をどう設計するかが、データベース管理システムの課題の一つです。データ数が多くなければ、普通のプログラミング言語を使っても程々の処理ができます。データを演算用レジスタに読み込んで計算し、内容の書き換えや並べ替えをします。扱うデータのビット並びの寸法単位を、紛らわしいのですが**ワード**と言います。8ビットのマイコン（マイクロコンピュータ）は、2バイト（16ビット）を1ワードとしましたので、全体は65Kバイトのメモリ空間を使うことができます。オペレーティングシステムと内臓プログラムの領域に32Kバイトを当て、残りを作業用に使います。したがって8ビットのマイコンを使ったテキストエディタは、最大で32Kバイトの文字数制限がありました。16ビットの計算機は、大型計算機に対して中型計算機と言いましたが、大型計算機に合わせて32ビット長のメモリ寸法を1ワードに割り当てました。16ビットのパソコンは、この中型機的设计を踏襲し、65Kのワード数寸法を1**ページ**長と約束しておいて、ページ単位でメモリ領域の管理をします。プログラミング言語では、大きな配列のデータを扱うとき、この65Kの寸法を巡って、幾つかの制限があります。例えば、表計算ソフトのEXCELは、表の行数の最大値に65Kの上限がありました。列数は、255まで、RAMの許容ページ数です。データベースを扱うとき、この制限を越える寸法も考えなければなりません。したがって、パソコンも32ビットのレジスタを使うようになって、本格的なデータベースの作業ソフトACCESSが可能になったのです。なお、追加のメモリに使うRAMは、多ければ多いほど作業のゆとりがありますが、常時使うメモリを、現金を手許に置くと言う意味を含めて、**キャッシュメモリ**（cache）と言います。32ビットのパソコンでは、1GB寸法のRAMが普通に実装されるようになりました。

3.5 ファイル装置の基礎知識

3.5.1 コンピュータ立ち上げの儀式

コンピュータに電源を入れて、作業に使える状態にするまでの準備を、立ち上げ(boot)と言います。昔々のコンピュータは、手作業で何語かの機械語のプログラムを入力し、実行させると、紙テープまたはカード装置を起動して、オペレーティングシステム (OS) のプログラムをコンピュータのメモリに読み込ませ、常駐させます。これを IPL(initial program load)と言い、一種の儀式のように実行させました。現在のパソコンの IPL は、歴史的な経緯から、自動的にフロッピーディスクを最初に探して OS を読みに行くように設定してあります。フロッピーディスクが無ければハードディスクを読むようになっています。現在のパソコンの OS は、往時に比べて巨大になっていますので、準備が完了するまで、しばらく時間が掛かります。電卓や 8 ビットのマイコンは、小さい寸法の OS が内蔵されていますので、電源を入れれば直ぐに使うことができます。コンピュータを利用する段階では、プログラムもデータも、鑽孔した紙テープかカードから読み込み、計算結果も紙テープかカードに出力しました。印刷は別作業です。コンピュータを使い易くするには、大容量の内部メモリと大容量の外部記憶装置の開発が必要でした。コンピュータの電源を切る時にも、一通りの儀式が必要です。RAM の領域で書き換えられたデータをディスクに保存するかどうかの判断が必要です。RAM の寸法が大きくなかった時代、RAM に書き込んだ全データを磁気テープに保存しておいて、次の立ち上げのときに読み出して作業の継続ができるようにもしました。突然の停電は、RAM の記録が消失しますので、作業に大きな支障が起きます。電池駆動のパソコンは、その心配がないので助かります。

3.5.2 磁気テープと磁気ディスク

コンピュータの実行に必要なデータを保存しておく大容量の外部記憶媒体として、最初に使われたのが パンチカード です。統計処理に主に使いました。鑽孔紙テープは、古くから電報用電信機に使われていましたが、データの追加・変更・並べ替えには不便でした。紙テープに代わったのが 磁気テープ です。これは、オーディオ機器、さらにはビデオ装置に使われ、一つの時代を画したのですが、現在では各種のディスク装置に代わってきました。磁気テープは、大容量の記録ができる利点があります。欠点は、データを頭から順に書き込み、巻き戻して頭から順に読み出す使い方をしなければならないことです。磁気テープの論理的なファイル構造を 順編成ファイル(sequential access file)と言います。テープの終りの方にあるデータを取り出そうとすると、巻き取りと巻き戻しにかなりの無駄時間を取られます。磁気ディスク装置、また、同類の磁気ドラム式は、任意の順番で書き込みと読み出しが、磁気テープに比べて高速です。こちらを ランダムアクセスファイルと言います。データベースを扱うには、先ず大量のデータを準備しなければなりません。それは、論理的には順編成ファイルをディスクファイルに作成することから始めます。そのデータを検索して、並べ替えなどの作業をするとき、データ全体または部分を内部メモリに移して作業ができれば、処理速度が圧倒的に改善されます。これが、前節で説明した RAM (ランダムアクセスメモリ) です。この使い方を ディスクキャッシュ(disk cache)と言います。なお、光ディスクが便利になりましたが、英語表記では disk に代えて disc と書き分けています。

3.5.3 固定長データと可変長データ

縦・横 複数の枠を使って表の形に書いて整理できるデータの、書き込み枠を セル(cell)と言います。文字並びが納まるような寸法にしますが、そこに書かれるデータの、コンピュータ内部の寸法単位は、バイト数です。この表をファイルにするとき、行単位をレコードと言い、その中をバイト数の異なる複数のデータセルに分けます。扱い易いデータ構造は、1レコードのバイト数が一定のものです。これを固定長レコードと言います。数値データの場合には、バイナリー形式で記録すれば、数値ごとに一定のバイト数を当てることができます。文字並びをレコードにすると、普通、一行の文字数が一定になりません。そこで、保存に使うセルの寸法ごとに或る最大バイト数を決めて置く方法(固定長)と、文字並びの数に合わせて変化させる方法(可変長)とがあります。表のデータを並べ替えたいとき、行または列単位のバイト寸法、さらに枠単位もバイト寸法が一定であると処理が効率的にできます。この操作は、行単位の固定長レコードで ランダムアクセスファイルに構築しておく、ファイル操作でもレコードの並べ替えができます。表全体データを RAM (ランダムアクセスメモリ)に読み込むことができれば、高速処理ができます。このことが、データベースシステムのハードウェアに対する基本的な要求事項です。

3.5.4 高速でデータを移しかえる命令がある

コンピュータの主演算装置（CPU: central processing unit）から見れば、ファイル装置も RAM も外部装置の位置づけです。ファイル装置にあるデータと RAM との間で読み書きする、さらに RAM の或る個所の一続きのデータ並びを単に別の場所の RAM にコピーする、などの移動処理に CPU を使うと、データを 1 個単位で処理するので非効率です。そのため、データ転送用の補助的なプロセッサが別にあって、そちらに一括して転送処理を任せる方法を取り、CPU はその間、別の作業をさせることができます。そのデータを CPU で使うときは、転送処理が済んだことを知らせる **フラグ** を参照するように、機械語のプログラムを組み上げます。テキストエディタやワードプロセッサは、この転送機能を有効に活用して文字並びの編集作業を行います。一固まりの文字並びを削除するときは、その文字並びの下にある文字列全体を上に移動させます。また、或る個所に文字列を挿入するときは、そこから下にあるデータ全体を下に移動させなければなりません。文字列をモニター上で確認するときは、表示のための工夫がされます。モニターは、表示専用にするメモリを別に用意して、そこにテキストを転送します。これによって、テキストのスクロールや、行末での文字列の自動折り返し表示ができるのです。眼で見る限りでは、この転送速度が気にならない位、高速になりました。

3.5.5 表形式のデータを順編成ファイルで利用すること

表の形にデータを整理してリストを作ると見易くなります。そのため、表計算を目的にしなくても、EXCEL をテキストエディタやワープロのように使うと、文字位置を揃える書式制御が便利にできます。表の作成に使うデータを、エクセルの作業画面（ワークシート）を開いて直接キー操作で入力することもできますが、別にテキストファイルに作成してあるデータの読み込み処理（インポート）を併用すると、効率がよくなります。EXCEL は、テキストファイルを読み書きする機能があります。そのデータ構造は、データの区切り記号（**デリミタ**: delimiter）にコンマ、またはタブを使い、一行の区切りに改行コードを使います。エクセルの作業画面は、行・列単位でも追加・削除・並べ変えができますので、表全体の編集作業にはテキストエディタを使うよりも機能的な処理ができます。編集が済んだワークシートは、テキストファイルに書き出して（エクスポート）、別の処理に使うことができます。

3.5.6 文書の目次や索引はデータベースの性格がある

文字数の多い著作、また、ページ数の少ないレポートの作成でも、本文の作文が済んだ時点で目次や索引を作ります。これは目録を作ることと同じです。しかし、この作業はかなり面倒です。一つの理由は、後戻り的な整理作業の性格があるからです。目次や索引を作成する目的は、検索にあります。著作物単位を一つの世界と見るとき、探したい項目のキーワードを並べ、その位置（ページ番号など）を教える手段が必要です。これは、小規模ながら、データベースの作成と見ることができます。紙の書物であれば、項目の位置をページ番号で示すのですが、電子化文書ではその方法を使うことができません。筆者は、HTML 形式のファイルの目次や索引を作成することも考えて、ページ番号に代えて、章・節・項の番号を使っています。したがって、マイクロソフト社の Word が、サービスとしているツールを使うのではなく、単純なテキストエディタ（メモ帳: NotePad）と EXCEL を使い分けて、目次と索引の原稿を作成しています。見本は、筆者の HTML 文書をインターネットで見て、目次または索引をパソコンのモニター表示しておいて、マウスの右ボタンをクリックして、ソースの表示をクリックすれば原稿を見ることができます。例えば下の例を見て下さい。例図を図 3.1～図 3.6 に示しました。

「易しくないコンピュータ言語学」 www.nakanihon.co.jp/gijyutsu/Shimada/Complinguistics/top.html

簡単に作成方法を説明しておきます。目次と索引の原稿は、Word で作成した本文原稿をコピーして、キーワードを残し、これを作業用のテキストファイルに改めてコピーします。これを EXCEL に読み込んでから、幾つかの共通文字並びを列セルとして追加した表に作成します。索引の場合には、キーワードを あいうえお順 に並べ変えるのですが、漢字の訓読みを使うこともありますので、フリガナ用の列セルを追加します。並べ変えが済めば、フリガナ用の列セルなど、不要な作業用セルを削除します。この EXCEL のデータを、コンマ区切りのテキストファイルに書き出し、不要になったコンマや制御文字をテキストエディタの置き換え機能を使って消去します。このテキストファイルの拡張子を *.txt から *.html に直し、幾らかの書式情報を追加して完成させます。図 3.1 と図 3.2 は、PDF 形式に変更した電子書籍のスタイルです。ユーザ側でダウンロードして、自分用のハードコピーにして読みます。インターネットでの閲覧用には、本文の項単位で HTML 文書にし、相互にリンクを取れるようにしてあります。このときの、目次と索引の作成見本が図 3.3～図 3.6 です。

目 次

0. はじめに

1. 言語学が関与する環境

1.1 言語学が生まれた経緯

狭い言語環境に居ると言葉数が少なく済む；言語は文化的な侵略の武器になること；外来語の輸入は語彙を増やすこと；言語と民族とは関連を持つこと；声を文字で表すことは難しいこと；音声を直接扱う研究は難しいこと；複数の言語環境の場では公平な言語学が必要になる

1.2 道具としての文字

言葉を音のまま研究対象にすることは難しい；文字の並べ方にも言語による違いがあること；漢文訓読という文体が表れたこと；英訳には翻訳調の文体があること；文字は形を持った道具と考える；部品に分ける単位が品詞であること；実用文書の書き方に応用する研究が必要；言葉は危険な武器にもなること

1.3 標準化と多様化

差別用語の扱いが難しい；官製の標準化があること；人工言語は標準化が意識される

2. 日本語文書の構造

2.1 読みの仮名表記

図 3.1 PDF 版に編集した文書の目次部分の見本

用語索引

アーカイブ	12.4.8	オブジェクト指向データベース	5.2.11
全	9.1.3	ス	9.4.3
アクション(action)	7.3.1	オブジェクト指向プログラミング	5.3.4
アクセント	12.2.3	全	9.4.3
全	2.2.1	オペレーティングシステム	7.2.2
アスキーファイル	6.5.4	オラクル社	10.1.5
アナログ計算機	11.2.1	奥付	6.2.3
アニミズム	8.4.1	音節	2.1.1
頭の良し悪し	12.2.6	音便	2.2.1
網かけ	11.2.3	折り紙技法	6.5.3
曖昧検索	8.2.2	折り畳み方	6.5.3
イラスト	11.1.1	送り仮名	5.2.1
インスタンス	10.3.3	大槻文彦	3.2.4
インタフェース	9.4.3	カーソル	10.3.4
インタブリタ	7.3.4	カード型データベース	
インデックス	7.4.1		
インデント	6.3.2		
		漢文訓読法	1.2.3
		漢文体	6.3.3
		環境	7.3.3
		間接目的	2.4.2
		間接目的語	7.3.2
		関係代名詞	3.6.2
		全	8.3.1
		関連語	10.3.1
		全	9.3.3
		型の定義	7.4.1
		賢さ	12.2.6
		書き言葉	12.2.4
		全	6.3.3
		川喜田二郎	9.3.1
		キーショートカット	7.2.3
		キーワード	9.3.2

図 3.2 PDF 版に編集した文書の索引部分の見本（3 段組み）

PDF 版に変換する前の元文書は、Word で作成します。図 3.1 の目次は、章・節番号までが見出しまでにして行数を抑えてあります。項の表題は、抄録の意義を持たせています。目次と索引の原稿は、EXCEL で整理を済ませて、再度テキストファイルに落とし、これを Word に取りこんで、編集し直して、用語索引のように段組み表示に作成します。

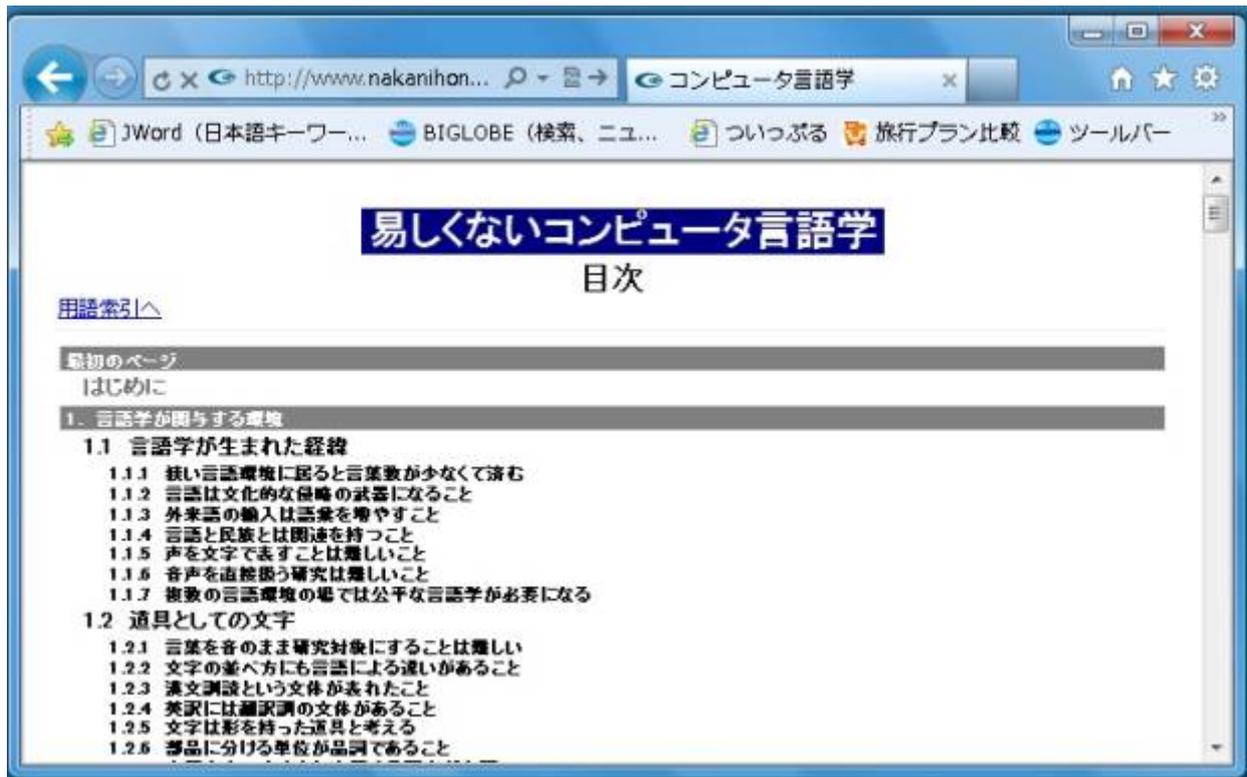


図 3.3 インターネットエクスプローラーで表示した HTML 版の目次



図 3.4 HTML 文書はテキストファイル（部分）です。目次本文は EXCEL の表作業で作ります。

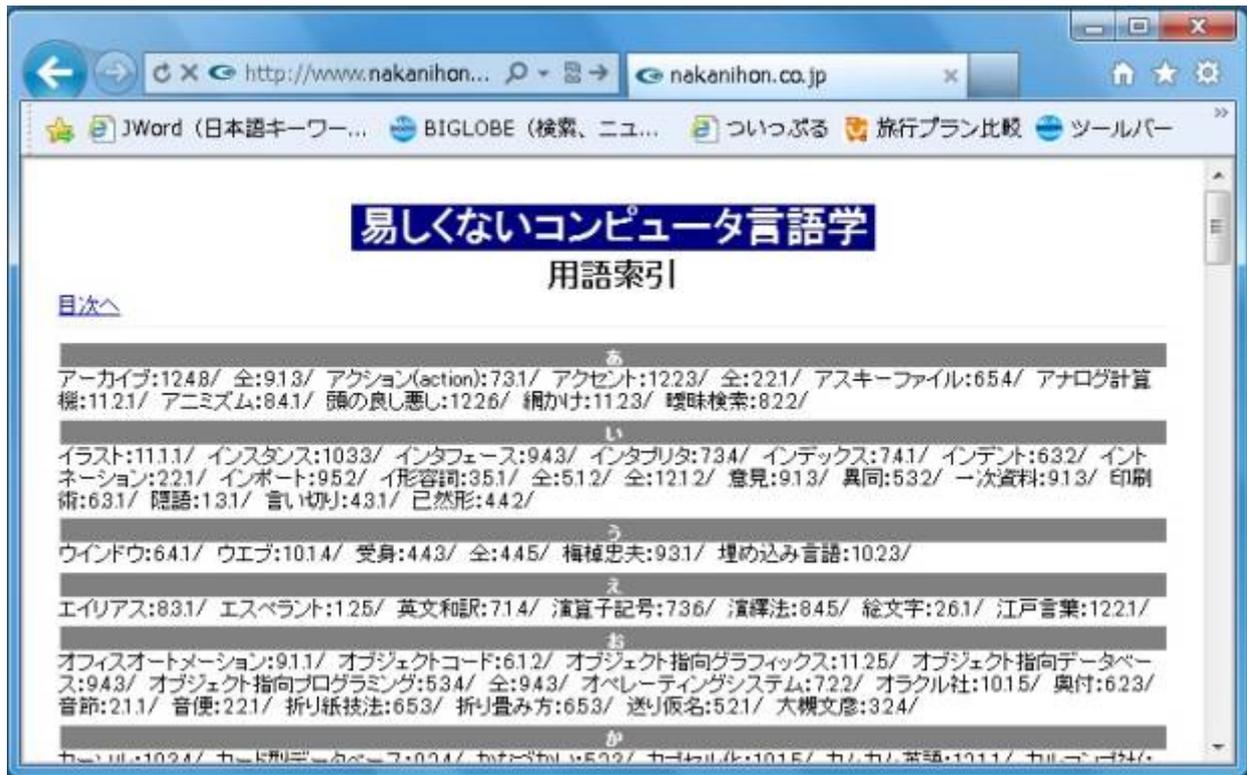


図 3.5 インターネットエクスプローラーで表示した HTML 版の索引 (部分)。



図 3.6 HTML 文書に作成した索引のソースです。索引本文は EXCEL の表作業で作ります。

4. データベース入門

4.1 英語用語の理解から

4.1.1 DBMS

第 2.6 節の最初の段落に、structure を説明する例として、SQL (Structure Query Language) の紹介をしました。SQL は、それを利用する親言語に COBOL (1959) を主に使うことが意識されたので、それに追加する機能として 1986 年に規格化されました。COBOL 本体は、既に、良く設計された言語として利用されていましたが、データベースの処理ができる予約語の追加が必要になったからです。COBOL が提案されてから 25 年も後になりましたが、これには幾つかの理由がありました。SQL は、コンピュータ側から見れば、サブルーチンや関数プログラムです。ファイルの高度な使い方が必要ですので、装置依存のプログラミングをしなければなりません。種々のファイル装置を使うとき、装置が変わる度にプログラムを書き換える面倒さを避けるため、プログラムの中では抽象化した装置 (デバイス) でデータを読み書きし、そこからソフトウェアとしての デバイスドライバ (device driver) を介して、物理的な装置の読み書きをします。デバイスドライバを管理するのが OS (オペレーティングシステム) です。したがって、SQL を追加して利用するには、OS の機能を強化しなければなりません。このソフトウェアが データベース管理システム (**DBMS**: DataBase Management System) です。これは、OS の機能と密接に関係します。特にファイル装置に依存する度合いが大きいことが特徴です。ファイル装置をオブジェクト扱いにした、オブジェクト指向プログラムと理解することができます。パソコンで、例えば、EXCEL や ACCESS を利用しようとするとき、OS のバージョン違いで読めなくなることがあります。このような裏事情がありますので、一般ユーザは、データベースソフトの作成に直接挑戦するのではなく、専門のプログラマと協力する方法を考えるのが良いでしょう。

4.1.2 データベースエンジン

上に挙げた DBMS を扱うユーザインタフェースのことを データベースエンジン (database engine) と言います。俗称で呼んだのでしょうが、感覚的に良い命名です。しかし、公式に言うときは、SQL (Structured Query Language) の方を使います。出来合いのデータベースを利用する一般利用者は、システムの中身についての知識は必要ではありませんが、検索などの機能を代行してくれるソフトウェアをエンジンに例えた言い方は、妙を得ています。エンジンが故障するようであれば、データベースのファイルそのものが使いものにならなくなるからです。

4.1.3 DDL, DML, DCL

SQL は、データベース処理に関係する言語の総称に使う用語です。より具体的には、三つのグループに分けた言語集団で構成されています。DDL (Data Definition Language: データ定義言語)、DML (Data Manipulation Language; データ操作言語)、DCL (Data Control Language) です。これらは、データベースのデータそのもの作成・追加・削除・変更を扱う機能を提供する言語です。したがって、プログラマとデータベースの管理者が理解する言語です。データベースを利用する側のユーザは、自分でプログラムを組んでデータベース本体に生でアクセスすることができませんので、この言語体系を使うことはありません。しかし、データベースについての一般常識として理解しておくことが望まれます。以下の用語も、そのような意義で解説します。

4.1.4 スキーマ

JIS のプログラミング言語の SQL には、スキーマとデータベースの用語の概念を 4.6 項と 4.7 項で説明してあります。この文を読んでも一般の人には理解できないでしょう。分かり難い理由の一つは、元になる英語の規格を訳したからです。スキーマの英語は schema です。scheme の複数形の意義を持つ集合名詞です。意味は、データベースを構成する項目の全体を云い、具体的に言えば、表の形に構成するデータの集合です。スキーマの定義と言うのは、表の行列の構成と、セルの中身のデータ型を決めることを言います。データベースの定義は、スキーマの集合としますので、複数の表で構成することを意味します。複数の表を使い、表相互の関連を付ける使い方が普通になりました。これを他の構造のデータベースと区別したいとき、リレーショナルデータベース (relational database: RDB) と断ります。DBMS も、それを受けて、RDBMS を使うようになりました。

4.1.5 マスターファイルとトランザクションファイル

一般ユーザは、表の形になった文字データの、論理的な構造を理解しておけば充分です。このとき、第 3.3 節で触れた、静的データベースと動的データベースの区分が重要です。表の中身を変更しない、または変更してはならないファイルが**マスターファイル**(master file)です。一方、表枠の中身を変更しながら使うファイルが**トランザクションファイル**(transaction file)です。この英語 transaction は、商取引を意味します。商品データは、価格や在庫の変更が頻繁に行われますので、これを扱うファイルは動的データベースの性格を持ちます。

4.1.6 マクロ

コンピュータ用語で使われる**マクロ**(macro)は、小単位のプログラミングコードを言う名詞です。接頭語的に、大きい・小さいと言う意味の macro/micro とは違います。汎用言語と言われる COBOL、FORTRAN、C 言語は、何から何まですべてをプログラミングできるツールです。それを使って作成した実行形式のプログラムをアプリケーションプログラム、ユーティリティプログラムなどと総称し、ファイル名の識別子に(.exe)が付きます。この実行形式のプログラムが作業用のファイルを使うとき、固有の識別子を付けます。そのファイル名をフォルダのリストからクリックすると、本体の実行形式のプログラムが起動して、そのファイルを開くように設定できます。マイクロソフトの WORD、EXCEL、ACCESS、NotePad などの使い方で馴染みのある方法です。ここで、例えば WORD の作業画面で、数学式を書きたいとするとき、WORD から間接的に呼び出す実行形式の数式エディタを呼び出します。これは、ユーザが単独で呼び出して使うことができません。EXCEL の場合、表計算を助ける関数副プログラムが豊富に準備されています。これは、一行で書ける式の中での関数名として使います。この計算結果を代入するセルの裏情報としてセルに組み込みます。しかし、少し複雑な処理をしたいとき、一行で納まりきれない計算手順が欲しくなります。この目的に使う小単位のプログラムがマクロです。この記述に Visual Basic 風の言語を使うので、**VBA** (Visual Basic for Applications) と言います。応用プログラムの中で、固有の機能を小プログラム単位、つまりマクロで実行させます。これを、例えば EXCEL VBA のように言います。こちらはユーザがプログラミングして作成するのですが、元になった Visual Basic のプログラミングを勉強しておく必要があります。その過程を覚える機会がない初心者には、荷が重いようです。誰かが作成したソフトに組み込まれているマクロがあります。これに悪意を持ったコンピュータウイルスが埋め込まれていることがあるのが問題を起こします。

4.1.7 DLL

コンピュータ処理を効率よく行わせるには、実行コードとデータとを、共に RAM に読み込んでおきます。RAM の寸法が小さければ、実行時にファイルから動的に読み込む使い方を工夫します。ビジネス関係の処理に使う COBOL は、データをファイルから読んで処理し、別のファイルに書き出すことを主に考えていました。科学技術計算では、複雑な計算手順のプログラミングコードが大きくなる場合があります。そこで、実行コードを分割しておいて、実行時にそれを動的に読み込む方法も工夫されました。16 ビットのコンピュータは、RAM のアドレス制御に 65K を単位として分割利用しなければならなかったので、**オーバーレイ**(overlay)と言う実行時制御をして、メモリ領域を効率よく利用する工夫をしました。この技法は、32 ビットプロセッサのパソコンでは、使う必要が無くなりました。しかし、たまにしか使わない実行コードが RAM を占拠しているのは RAM 資源の無駄使いですので、実行要求が発生したときに、ディスクファイルから読み込んで使う小寸法のサブルーチンが工夫されました。これを **DLL** (Dynamic Link Library) と言います。このオブジェクトコードは、拡張子が(.DLL)です。そのプログラムの作成者だけが詳細な仕様を知っているので、それ以外のプログラマは、使う方法が分かりません。つまりブラックボックス化です。VBA が、ソースコードの見える使い方であることとの、大きな違いです。

4.1.8 中身の読めないデータファイルの問題

これは、上の二つの用語についての追加のコメントです。例えば、EXCEL のデータファイルは、拡張子が(.xls)です。主要な中身が文字データであっても、それをテキストエディタで開くと、暗号並びのような記述しか見られません。このファイル構造は、データとバイナリーコードの両方を含むからです。そのため、バージョン違いの EXCEL を使ってデータファイルを開こうとしても、読めない障害を起こすことがあります。これはデータ管理上 困ることです。したがって、データのバックアップをするとき、テキスト形式での保存 (例えば CSV 形式) が望ましいのです。

4.2 ハードウェアとソフトウェアの要求事項

4.2.1 専用のコンピュータを使うのがよいこと

大型コンピュータシステムが高価であった 1960 年代までは、このコンピュータを多目的に使う必要がありましたので、汎用コンピュータの呼び方が使われました。この時代、米国では、図書館でデータベースを利用するためだけに、一台の専用コンピュータを追加するように割り当てて、汎用コンピュータの負荷を軽くするようにしました。こちらを database machine として、汎用コンピュータの周辺装置に位置づけていました。贅沢なように見えますが、効率的なサービスをする方法としては正解です。データベースは、データ量が大量になる可能性がありますので、外部のディスク装置は複数とし、追加も考えます。また、セキュリティのため、バックアップも考えます。データベース専用のディスク装置にしてあると、外部の利用者がファイルを壊すことを防ぐセキュリティの方法に、ハードウェア的な読み書き禁止にすることもできます。ディスクの読み書き速度は、高速になったとは言え、CPU の演算速度に比べれば桁違いに低速です。大きなファイル単位を扱うと、待ち時間も馬鹿になりません。個人のパソコンで本格的なデータベースが扱えるようになったソフトウェアの ACCESS では、作業に使うデータは、拡張子が (.mdb) の作業用ファイル一つのみを使います。しかし、その中身は複数のデータファイル、制御用データ、それに、特別なソフトウェア (マクロ) を含みます。ACCESS でこのデータベースを開くと、内部的に作業用の RAM に個別のファイルに展開しています。また、作業が済めば RAM のデータを編集してからディスクに保存します。始末が悪いことに、ACCESS のバージョン違いで、ファイルが読めない場合も起こります。単純なテキストエディタを使って mdb ファイルの中身を見ることもできません。したがって、一般の人は、データベースの扱いに挑戦することは難しいでしょう。

4.2.2 表計算ソフトの利用から

会計計算は、表の形に作成して作業をすることが一つの定型ですので、パソコンのモニタ上でインタラクティブに処理できる表計算ソフトウェア (スプレッドシート: spreadsheet) の方が、COBOL でプログラミングするよりも便利になりました。代表的なものが EXCEL (エクセル) です。16 ビットのパソコンを利用するときは、表の行数の最大に 65K の制限がありました。一般の事業所で使うには充分の機能がありました。簡単な検索機能がありますし、データの並べ替えや印刷機能も程々です。このままで、小規模のカード型のデータベース的な利用方法ができます。やや本格的な ACCESS のようなデータベースを利用する場合であっても、データ作成を EXCEL で作業しておいて、それを ACCESS に取り込むこと (import) は普通に行われています。したがって、ACCESS などを利用するにしても、基礎的な知識として EXCEL の使い方に慣れておくのがよいでしょう。

4.2.3 カード型データベースが使い易い

図書館で利用できる図書カードは、カード型データベースの原形の意義があります。検索サービスに使うため、二種類の見出しカードを作ります。著者名カードと、書名カードです。どちらも、和書はあいうえお順、洋書は ABC 順で並べます。これをコンピュータ化するときには、蔵書全体を一つのマスターファイルを作成しておいて、それから、種々の検索語を使って図書のレコードを探します。検索を便利にしたいため、ソフトウェア的に種々の工夫が考えられてきました。データ量が少なければ、単純な目録に作成します。そのツールとしてテキストファイルを使うことができます。これも検索機能を使うことができます。EXCEL の表形式に読み込めば、見易いリストにするような並べ替えが簡単にできます。プログラマの心情としては、便利な検索機能を工夫したいのですが、必ずしもユーザに親切になるとは限りません。Google が提供しているインターネットの検索機能の使い方に、一般の人が慣れてきました。その入力も、思いついた言葉 (自然語) を入力するだけであって、著者名・書名・固有名詞・専門用語などの区別をしなくても済む方式です。これは KVS (Key-Value Store) と言われていて、静的なデータベースを利用する方法です。考え方は至って単純です。データの管理も単純になっています。そうでなければ、膨大なデータ量を扱うことができないからです。探したい項目本体の保存個所、インターネットの世界では、ウェブサイトが分かればよく、そこから先の探索はユーザ側の努力です。図書で言えば、カード情報に番号をつけておきます。別に、作業用にキーワードとこの番号の対照表をつくり、これを検索に使います。複数のキーワードがあると、その数だけ対照表の項目数は増えますが、元のデータリストを直接使いませんので、作業用のデータが大きくなりません。さらに、元のデータがどこにあるかの所在情報だけを扱いますので、ファイルがどこにあってもよいのです。この発展的サービスとして、電子化書籍、さらにはクラウドサービスが注目されるようになりました。

4.3 ワープロ用辞書への要望

4.3.1 シソーラスの作成が準備作業として必要

データベースだけに限りませんが、検索をするときはキーワードを入力しなければなりません。この作業は、キーボードの操作を必要とし、GUI の利用ができませんので、案外面倒なものです。キーワードは、主に連想(association)で思い出します。その選定は、検索資料を準備する側(サーバ)と利用者側(クライアント)、夫々に立場が異なります。基本的なことは、あまり特殊な用語をキーワードにしません。利用者側が思い付かないキーワードは、検索洩れを起こします。文字単位で言えば、漢字ならば誤字・当て字、英語ならばスペル違いの語もそうなります。逆に、あまりにも普通の用語も焦点が定まりませんので検索語には不適當です。これを助ける工夫の一つが連想探索です。シソーラスは「検索に使う語を、これにして下さい」と言う語の集合を集めます。データベースのプログラミングの参考書では、キーワードとシソーラスについての解説は殆んど載っていません。これは、そもそも第2章で説明した文字処理と関係します。その語彙は、分類を決めて階層構造にしておきます。上位語(BT)、下位語(NT)、関連語(RT)が分かるように編集します。例を挙げます。例えば、住所表記では、愛知県・名古屋市・名東区の語順は、上位語から下位語に並んだ階層構造です。名古屋市に注目すれば、愛知県が上位語、名東区が下位語です。岡崎市、豊橋市が関連語ですが、同位語と言う方が階層構造的な言い方です。検索をしているユーザは、シソーラスが手許にありませんので、この階層構造を辿って別の用語を発見することができません。この用語検索も、データベース利用技法の一つです。

4.3.2 正引きと逆引きの辞書

漢和辞典は、部首を大枠分類とし、画数順に漢字を見出しに使い、その漢字を頭とする熟語を並べてあります。例えば、正の字を使う二字熟語は、「正位」を始めとして50語ほど取り上げてあります。しかし、二字目に正と使った、例えば、「修正」のような熟語も参照できる漢和辞典は、コンピュータを利用して編集ができるようになってからです。筆者が持っているのは平成8年版(1996)角川必携漢和辞典です。岩波逆引き広辞苑は1992年が初版です。普通の文字並びの辞書は、特に正引きとは断りません。文字の並べ方についても、あいうえお順が常識ですが、昔の習慣は、いろは順が普通でした。何かの文書を作成するときの目次は、項目の並び順は内容の種別によりますので、文字の読み順にはなりません。そこで、別に、何かの約束を使った並べ変えのリスト(例えば時間順)を作ることもします。索引を作るのが丁寧な編集です。こちらは、文字並びをあいうえお順とするのが普通です。表現方法は違いますが、目次と索引の対応は、内容の正引きと逆引きの関係になっています。前の項で説明したシソーラスでも、階層構造を正逆2方向に辿れるようになっていないと、適切なキーワードを探すことができません。或る語の上位語は一つです。下位語は複数ありますが、それらはすべて紹介しなければなりません。関連語は、そこには示しません。こちらは、一旦上位語に戻り、その下位語から探します。このキーワードの探索方法は、キーワードが階層構造に分類できることが条件です。いつもそうとは限りません。ネットワーク(網目構造)は、関連を探す道筋が一意に決まらないので、探索に失敗することがあります。索引は、語の並べ方に規則を設けますが、内容へのアクセスはランダムです。

4.3.3 文字校正用のツールに辞書を選択する

日本語は表意文字を使いますので、ワープロでは、音の並びから漢字の候補を表に示し、眼で見て適切な漢字または漢字熟語を選択します。ユーザは、頭の中で文字を音として発想しています。同音異義語を撰んでも、本人は気が付かないことがあります。例えば「人口衛生→人工衛星」がそうです。文学作品では、本人承知で当て字を使うこともあります。新聞や週刊誌などは、見出しなどに同音同義の熟語を、意図して造語することも見ます。ワープロ側では、変換候補の熟語が適切な場面で使われているかどうかの検査ができません。文字並びの候補が多い方がよい、と考えるのが普通ですが、或る程度の仕分けがあると使い易くなります。例えば、常用漢字以外を使った人名・地名などの固有名詞を撰びたいとき、それを辞書の語彙に含めます。専門用語には、固有の文字並びが使われます。筆者が間違えた一つの例に材料強度の限界を示す「降伏点」があります。これを「降服点」していました。専門用語の辞書が、シソーラスです。専門に関わる文書を作成するときは、それに合わせて変換用辞書を選択・追加・削除できるようにしておくのが良いのです。その場合、誤用の危険がある文字並びの辞書が欲しくなります。これは負の意義で参照する辞書です。上の例で言えば、「人口・衛生・降服」がそうです。差別用語も、そのような辞書にしておきます。現在のワープロ用辞書は、負の機能の辞書は扱っていません。

4.3.4 辞書には種々の種類があること

海外技術に学ぶには、外国語から日本語への翻訳に利用する辞書が要求されます。明治以降、その外国語に当たる和語の語彙が無いことが多かったため、それに当てるため、多くの漢字の熟語が造語されました。漢字は、和語に比べて抽象的な意味を持つ語が多く、熟語を作る機能が高いからです。漢字は、元はと言えば、中国からの輸入語です。日本で作る熟語は和製漢語と言い、本場の中国語の造語習慣と異なるものもあります。和製漢語を提案して学者は、漢学の素養がありました。そのこともあって、専門用語などは、中国でも使う例があります。日本語にない事物には、元の言葉の発音を元にしたカタカナ語の借用語が使われます。これが増えると、意味不明の日本語になります。カタカナ語辞典と言う不思議な辞書があって、その目的は、元の語の正しいスペルを教えます。辞書と言うと、例えば、「dog⇔犬」とすることとありますが、これでは何のことか分からないことが起こります。そこで、dog とはどのような動物か、その定義や概念を説明する辞書が必要です。こちらが、**百科事典** (encyclopedia) や用語辞典(glossary)です。岩波の広辞苑は、百科事典の性格があります。データベースを作成したり利用したりするとき、文科系の学問と考えていた言語学の領域まで勉強しなければならないことは、理科系の研究者の予想に無かったことです。

4.3.5 自動翻訳を考えると

技術の国際化が進んで、外国語の文献から日本語に翻訳するだけでなく、日本語を外国語に翻訳する需要が増えました。そうであると、専門用語についても逆引き辞書が必要です。翻訳は難しい問題があります。人に頼る作業を軽減するため、自動翻訳が研究されるようになりました。英文和訳は何とか日本語として読める文章にできますが、和文英訳はみじめな現状です。これは、そもそも、翻訳に当てる文章が、論理的に作文されていないことの方に罪があります。文学的な表現は含みを持たせた省略、話題を間接的に表すレトリック、冗談を加える、などの技法を評価するのですが、これが間違いの元になります。コンピュータは、裏の意味を読み取ることはできません。率直な話し方は失礼になる、と思われているからです。話が脱線するようですが、論理的な、実用を目的とする作文指導をする場が、日本の教育環境では抜けていたからです。現実的には、旅行などの日常会話での相互翻訳があります。これにコンピュータを介在させる試みが研究されています。多くの言語種類に対応するには、言語ごとに翻訳機能を開発したいところです。その種類と複雑さを避けるため、一つの**中間言語**を介して、二段階の翻訳をすると効率的です。その中間言語には英語が便利です。この英語は、必ずしも正確にイギリス英語またはアメリカ英語にはなりません。標準言語と言わず、中間言語を言うことに注意します。

4.3.6 プリントサービス

データベースの利用者は、検索要求の検索語を入力し、何かの結果(出力)を返してもらいます。端末、通常はパソコン、の画面で見ますが、その情報は一過性です。この情報を、メモ的に紙に写しておく手段も必要です。この表示区別を、前者を**ソフトコピー**(soft copy)、後者を**ハードコピー**(hard copy)と言い分けます。多くのソフトウェアは、プリンタに転送するツールを持っています。しかし、プリンタは、ハードウェアであるプリンタと、その制御に OS の機能が関係します。プリントの作成はユーザ側のプリンタを使います。自宅でパソコンを使う個人ユーザは、プリンタを持っていないこともあります。企業では、部局単位でプリンタを共用します。しかし、テキスト形式のデータで出力する機能は、必ずしも持ってはいません。例えば**インターネットエクスプローラ**は、主に、モニタ画面上で見ただけのツールです。プリント機能はありますが、テキスト形式でデータをファイルに書き出すことは考えにありません。何かのデータベースのサービスをしているサービス元(サーバー)は、検索結果を HTML 形式にして送信する方法を採ります。マイクロソフト社の **Active X**、またソフト名として **ASP**(Active Server Page)を見るようになりました。これは、ウェブページを動的に作成するソフトウェア技術です。これらを利用する環境は、データベースのサーバー側ですので、一般ユーザからは距離のあるソフトウェアです。保存に使える、書式の整った、また送信先で内容を変更しては困るセキュリティを持たせるデータには、PDF ファイル形式を使い、メールの添付資料として別に通信します。個人の場合には、そのデータファイルを USB メモリなどに入れて、街中の印刷屋さんを持ち込めば、製本サービスまでしてくれるようになりました。

5. オブジェクト指向データベース

5.1 何をしたいかの問題意識

5.1.1 出来合いのソフトは無いのが普通

コンピュータは、高速の数値計算をする道具を開発するという明確な目的意識から始まりました。コンピュータを実務に利用するには、目的に合わせてプログラミングを工夫します。これは、知的な創造的作業です。意図した通りにコンピュータが動作してくれることは喜びです。この延長として、実務以外の趣味的な利用、例えば、コンピュータゲームの開発は、若い人の興味を引き、これが直接間接にコンピュータ全体技術の発展を刺激しました。コンピュータをゲームに使うことや、当面の目的以外に利用する場合であっても、上に立つ実務の責任者は、管理上、白い眼で見ます。ハッカー(hacker または cracker)は、非常に優秀なプログラマとしての能力を持つ人が、いたずら心で悪意のあるプログラムの作成に走ったのです。これらの人は、実務上の明確な目標を見つけることができれば、その能力を良い方向に発揮できるでしょう。そもそも、文化と言うのは、真面目な宗教心も原動力になりますが、遊び心も、昇華して花が開きます。日本の漫画やアニメも、真面目な教育者は白い眼で見る傾向があります。良い作品は、いまや日本を代表する文化として世界的に認められるようになりました。データベースの話に戻すと、データベースに構築したい真面目な要望は、専門ごとの事情を考えますので、公式的な方法は無いと言えるでしょう。手っ取り早く解決したいとしてインターネットで探しても、ぴったりしたソフトが見つかる確率は高くありません。市販のデータベースの参考書が扱う例題の多くは、ビジネス指向です。これは比較的汎用性が高いからです。専門ごとのソフトの開発問題は多様ですので、問題点を説明して、若い人たちが興味を持って挑戦してもらいたいものです。

5.1.2 ベンチャーソフトを使わない

コンピュータを、利用者の方で見た歴史は、ハードウェアの変更と、ソフトのバージョン改訂とに対応する苦闘の連続です。ハードウェアの方で言えば、データの記録媒体である、例えばディスクの仕様が目まぐるしく変わりましたので、新しい媒体に対応できなくなって死滅した資料が増えました。データは残っていても、ソフトウェアが異なったコンピュータシステムに対応できなければ、これもデータの死滅に繋がります。一時的な処理では、ベンチャーが作成したソフトは気の利いた便利さが受けます。しかし、バージョンの改訂に対応できなくなる危険と隣合わせです。したがって、プログラムもデータも、テキスト形式ソースコードで残し、さらに眼でみて読めるハードコピーで保存しておく安全策も必要です。紙に印刷したものは嵩張りますので、マイクロフィルムで残すこともします。そうすると、内容を公開することに繋がりますので、秘密にしたくなるのですが、読めない資料は、結果として全滅する危険があります。そのことを意識して、安全対策が必要です。このことは、第 1.3 節に説明したアーカイブと第 2.4 節で説明したシステムズエンジニアリングに関係しています。

5.1.3 橋梁資料のデータベース化の試み

筆者が専門の一つとしている橋梁工学分野の専門家は、一つの技術集団を構成しています。その理由は、設計から架設まで、安全と危険とに向き合うからです。橋梁は図体が大きく、また費用も嵩みます。試しに作ってみて、具合が悪ければ作り直すという、試行錯誤ができません。設計は、意匠的なデザインも重要ですが、力学計算で安全性を確かめる作業が不可欠ですので、数学の学力も必要です。橋梁が完成した後でも、荷重の通行の段階で支障が起きることがあります。そのため、既に完成して安全に利用されている橋梁のデータを参考にする調査が重要です。この情報をデータベース化したいとする要望も自然です。また、一般の人にも向けて、そのデータベースを利用して知的好奇心を満たすことにも、また、教育用資料として利用することもできます。多くの研究者は、その作業を個人ベースで行ってきました。一般的な興味は、どうしても有名な橋梁を恣意的に選択する方に偏ります。網羅的な作業は、藤井郁夫氏が、個人的に努力して EXCEL にまとめた大部の資料が、現状では最も充実したソースです。その CD-ROM(1995)は、土木学会で閲覧できます。この作業の時代は、まだデジタルカメラが一般化していなかったこともあって、写真や図面などを除き、主に文献資料をソースとして、そこから得られた橋の情報、つまり、橋梁名・架設位置・完成年・橋長・支間・形式 などを集めた資料です。データベースの一次資料は、書籍や文献を対象とするのですが、橋梁実体のデータを構築したいときは、オブジェクト指向データベースの、オブジェクト(対象)を、橋梁と書き換えることを意味します。これは、個人情報を集めてデータベースに作成する方法と似た構造になる、と説明すれば納得できると思います。

5.2 図面データの電子化技術

5.2.1 画像データベースが重要な課題

橋梁の資料は、写真や図面があると具体的な知見が得られます。写真の方は、一般の人も興味を持ちますので、多くの資料があります。写真が実用されなかった古い時代から、多くの画家が橋を題材としました。これには歴史的な資料としての意義があります。絵画が作成された年代が分かっていると、その年代の記録になっているからです。現在では存在していないか、条件が変わったことが、記録として残ります。ただし、専門家から見ると、詳細構造まで分かる例は多くありません。写真を使う記録は、絵葉書の題材に多く見られます。これも、郵便の日付を見ることで、年代が分かります。アマチュアが撮影した写真は、インターネットで多く見られるようになりました。しかし、撮影年などのデータの扱いには無神経です。図面や専門的な視点で撮影した写真は重要です。ただし、図面は、マイクロフィルム化して保存することもしますので、風景写真とは扱いが異なります。写真と図面とを含めて、これらを一次資料の扱いをするデータベースを**画像データベース**と括することにします。図書とは、図と書とを合わせた意義の言葉ですが、図をデータベースに構築するときは、別の視点が必要です。それは、中身が分かる程度の小さい寸法の図をまとめて目録を作成しないと、検索の能率が悪くなるからです。画像の場合は、親指の爪大の縮小画像(**サムネイル**：thumbnail)の利用が便利になりました。デジタルカメラのデータを写真屋さんを持ち込むと、CD-ROMに保存してくれます。ネガに代える、一覧用のサムネイルをサービスしてくれますので、広く知られるようになりました。

5.2.2 最初はビデオテープの利用を考えた

筆者は、画像データベースの研究を1990年から始めていました。この目的は、一貫して教育利用が焦点でした。その目的には、三つの課題がありました。①：画像を保存し、それを表示する電子化媒体、②：大量の画像データを素早く探す方法、③：検索方法です。第一の問題は、スライド写真をコマ撮りしてビデオテープに保存する方法です。ビデオテープはランダムアクセスには不便でしたので、当時は12インチの大版レーザーディスクを使うことにしました。この基礎研究は放送大学で行われていましたので、そこの施設の協力を仰ぎました。当時のテレビの画像はアナログデータで送信されていて、アメリカ規格のNTSC(National Television System Committee)方式でした。画面の解像度は、現在のデジタル画像と比較すればみじめなものですが、教育用表示には充分と考えていました。レーザーディスクの再現装置には、やや特殊仕様の、パイオニア製レーザーディスクプレーヤを使いました。普通のビデオプレーヤでも静止画として見る機能を持つ装置がありましたが、ランダムアクセスには向きません。保存されている画像一覧を見る方法は、これらのデバイスでは無理でした。希望する画像を見たい時の検索は、別に作成した目録からコマ撮りの画像番号を元に行いました。画像ならば、画像の特徴を生かす検索方法もあるのではないかと考えました。単純な線画のスケッチをパソコンの画面を使って入力し、それと似た画像を取り出す方法も実験してみました。これと似た方法は、ワープロに使う、読みの分からない漢字を探すとき、手書き文字を入力して漢字を検索する方法があります。橋の場合は、あらかじめ、元の画像の外郭線を強調するような図のデータを作成しておいて、それとのマッチングが良いものを候補として取り出します。これは文書情報で言えば、抄録作成に当たります。研究の結論から言えば、実用には向かないことが分かりました。

5.2.3 デジタルカメラの普及が情勢を変えた

デジタルカメラの歴史は1975年まで遡るのですが、これが庶民にも手が届くように普及してきたのは2000年以降です。それを支えたのは、高密度のデジタルメモリが安価に利用できるようになったことが大きな理由です。デジタルカメラ自体の解像度も上がってきました。その表現方法として、CCD素子の画素数で何万画素のように言うのですが、直観的な理解に向きません。パソコンモニタの**解像度**(resolution)を言うときは、例えば640×480ピクセルのように言いますが、このピクセル数の積、ここでは約300Kになります。これが30万画素に相当します。1ピクセルに濃淡情報として1バイトを当てると、モニター画面当たり300KBのメモリが必要です。画素数の大きなデジタルカメラで撮影したデータをレーザプリンタで印刷するとき、プリンタの解像度は単位長さ1インチ当たりのドット数(DPI; dot per inch)と言う習慣です。通常約0.1mmよりやや狭い間隔の、2点が識別できる精細さが使われますので、約300DPIが多く使われる数値です。これらの数値を踏まえて、画像データベースの骨格を設計します。

5.2.4 図面保存のファイル形式

設計業務では図面の管理が大問題です。写真と図面とは、印刷段階ではどちらも同じ扱いです。図形の物理的な構造は二種類です。**線図**(line drawing)と**濃淡図**です。後者はカラーも含め、コンピュータ処理では点情報 (**ピクセル**: pixel) の集合であって、**ビットマップ** (bitmap) と言います。文字も図形ですので、文字データは**ベクターフォント**と**ビットマップフォント**が当たります。工業製図は、線図が主体です。このデータは、**グラフィックス言語**を使ってテキスト形式で表現できます。ファイルに保存するときの構造は、テキスト形式と、バイナリー形式とがあります。後者は記憶領域を小さくできて、作業段階では処理速度も速くなります。しかし、専用のソフトを使って読み書きしますので、別のソフトでは利用できなくなる危険があります。工業製図に応用する CAD のソフトウェアは、作業画面の裏で作図手順をグラフィックス言語に変換してファイル保存します。モニターで再現できて、また修正もできます。このファイル寸法は図面の大小ではなく、線の多少に関係します。一方、写真に代表されるビットマップデータは、二次元のピクセル並びですので、画像の寸法(面積)に比例してファイル寸法が大きくなります。眼でみて粗さが気にならないピクセル密度は、パソコンの画面では約 30 ピクセル/cm です。プリンタの方の解像度にするると、約 75 DPI になります。工業製図は A1 版の用紙寸法を使うことが普通です。これをビットマップ形式に変換すると、ピクセル数は、約 9000×6300 になり、1 ピクセルに 1 バイトを当てて正直に保存すると、約 60 MB もメモリを必要とします。ピクセルデータを生で扱うファイル形式が(*. bmp)です。したがって、データを圧縮する画像ファイルの形式が研究されてきました。このデータ形式は、圧縮技術と関係して、画像ファイルの識別子に幾つかの種類があります。一般的に使われるものが(*. jpg)、(*. gif)です。工業製図では(*. tif)を良く使います。これらのファイルの寸法は、画像データをメモリに保存するときだけでなく、インターネットを介してデータを転送するときの効率に直接関係します。パソコンで利用できるメモリ寸法の上限にかなり余裕が得られるようにはなりませんが、大量で大寸法の図面データを扱うことを考えると、メモリの増加を抑える対策が不可欠です。

5.2.5 実用的な寸法の画像に作り直す

CAD のソフトウェアで作成した図は、作業時に使ったグラフィックス言語のファイル形式ではなく、変更しない完成品として、レーザプリンタでハードコピー化するか、ビットマップ形式のファイルにします。手書きで図面を作成していた時代は、計算書などの書類も含め、マイクロフィルムにして保存しました。これらは、改めてスキャナーでデジタル化できます。元の資料は、アーカイブとして保存する施設が無い場合、廃棄します。これらのデジタル画像は、元のデータが読める程度の解像度が必要ですので、ファイル寸法が大きくなります。このファイルの保存媒体に、大容量の光磁気ディスク (MO が使われましたが、このデバイスの時代は終わり、現在(2011)では CD-ROM の利用に移行しています。筆者が持っている MO は、80 本ほどありますが、CD-ROM へのバックアップを考えなければならなくなりました。目録を頼りに、これらの保存媒体を見る機会は少ないのですが、その理由の一つは、複数の枚数の図面全体を一覧する方法が不便であることです。図面をデータベース化して検索するときには、美術書目録のような、小寸法の図を添えた一覧をまず表示しておいて、それを見て選択する方法が必要です。これには、写真のときのサムネイル作成方法を応用できるようになりました。

5.2.6 画像の縮小には賢いソフトが必要

サムネイルの寸法は、ほぼ郵便切手大です。ビットマップ形式の画像を縮小するときは、例えば一つおきにピクセル情報を抜けば、寸法は 1/2 になります。しかし、単純にこの操作をすると、細い線は切れ切れになり、線としての図形が消えていきます。最近の画像処理ソフトは、細い線図で描いた図面をサムネイル化しても、程々の線図で観察できるように進歩してきました。元々、工業製図では濃淡図やカラーを使うことを制限し、ハッチングなどの作図技法を規格化しています。しかし、A1 版の手書きの図面を、切手寸法程度に縮小してしまうと(図 5.1C)、何が描いてあるかも分からなくなりますので、35 mm フィルムのコマ寸法程度(図 5.1B)が限界です。図 5.1A は、その約 2 倍、名詞寸法に作成した例です。白黒画像をマイクロフィルムで保存する、またはスキャナでビットマップ化するには、コントラスト(明暗度)が高くなるように、元の図面を墨入れして清書するのが最善です。鉛筆で製図して青写真にした元データからスキャンすると、再生画質が落ちるのが一つの悩みです。図 5.1 は、コントラストを強調するように修正したものです。元の画像は線が薄いので、縮小が過ぎると、線図形の大部分が分からなくなってしまうことを見て下さい。

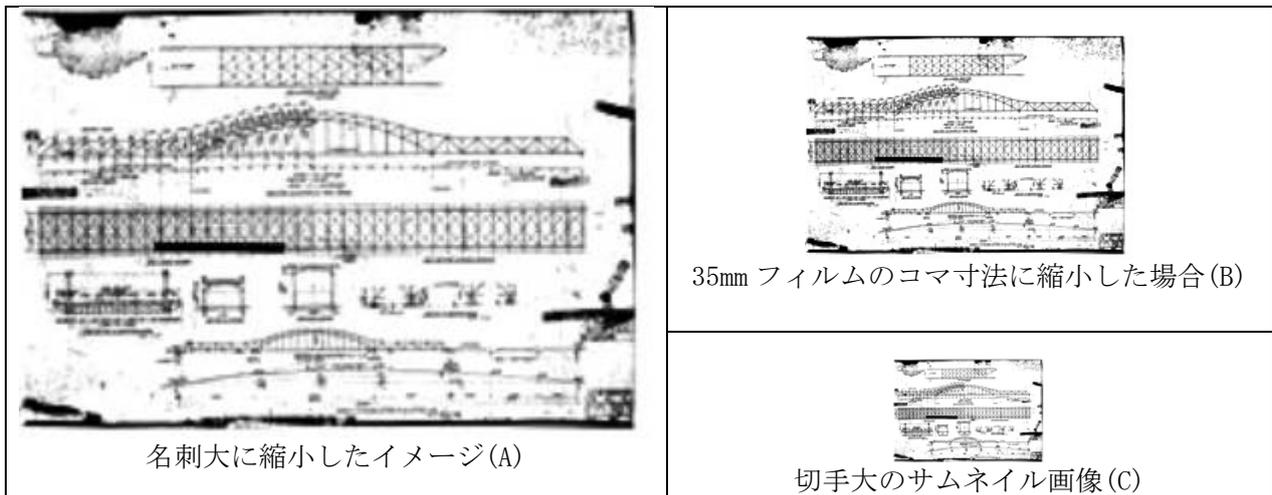


図 5.1 忠節橋（岐阜県）の一般図

5.3 写真データの目録作成と利用技法

5.3.1 例題として橋の切手を目録化したこと

筆者は、個人的な趣味として、橋を図柄とした、日本と世界の郵便切手を集めてきました。特別な説明なしに、その図柄は、インターネットで公開しています。10年ほど前からドイツ人の土木技師とペンフレンドになり、情報交換をしています。そのドイツ人はコンクリート橋の技師として日本に来たことがあって、日本の橋の切手も集めています。日本の切手の図柄を説明してきましたが、それを機会に、切手をデータとした、画像データベースの構築の例題にするように再整理をすることにしました。その作業をして得られた、幾つかの知見を説明することにします。郵便切手 (postage stamp) には二つの顔があります。一つは郵便料金を前納した証紙として、未使用のものは有価証券としての価値があること。二つ目は、切手の図案が著作物であることです。収集家の間では、美術品的な価値、さらには骨董的な価値が付いて売り買いされる価格があって、そのためのカタログも発行されています。筆者は、橋梁工学を専門としていますので、教育利用を目的として、橋に関係のある絵柄の切手を集めてきました。切手の紹介をするときは、切手の図柄を複製しなければなりませんので、公的な許可が必要です。しかし、消印があるものや、黒い線を入れるなどがしてあれば、許可なしに複製することが認められています。原寸大の切手は図が小さくて細部が分かり難いのですが、そのまま、サムネイルになっています。画像処理をして2ないし3倍に拡大し、斜めの線を加えてWEB上で公開しました。図柄と共に添付した参考写真などの著作権については、切手ごとの説明の個所で紹介することにしました。



図 5.2 英領西インド諸島ニービスの切手。ロイヤルアルバート橋 (1859) と設計者ブルネルの肖像

5.3.2 分類番号の設計

橋を図柄デザインに持つ切手は、毎年、何点かの割合で発行されていて、日本の切手では 2009 年末で、約 210 点あります。2006 年から、フレーム切手、一般には写真切手が発行されるようになり、これが切手の収集家の対象になってきました。これは、郵便局のサービス事業です。個人または企業も、自前の写真やイラストを入れたオリジナル切手の発行が認められようになりました。全国的な規模で販売される切手ではありませんし、網羅的なカタログもまだ無い状態ですので、偶然みつけられれば、購入する切手です。データベース的な見方をすると、切手の情報は静的な性質があり、本体データの変更はありませんが、管理上は追加を考えます。日本の切手カタログは、日本郵便切手商協同組合 (**JSDA**) 編と、財団法人日本郵趣協会 **JPS** 編との二つが主要なものです。ただし、整理番号の付け方が違います。海外切手では、米国の **Scott** カタログを切手商が主に使います。日本では、JPS が日本語訳にした国別のカタログを出しています。ドイツで発行されている **MICHEL** (ミッシュェル) も、日本を含めた全世界の網羅的なカタログを出しています。筆者は、ドイツ人のペンフレンドから中古版を譲りうけて参考にしています。新しい年代の切手の情報が間に合いませんが、急ぐこともありませんので、追い追い資料を追加することにしています。この三種類の分類は、相互参照ができるようにしました。橋と言う視点で切手を扱うときは、別に固有の番号を付けます。どのようにするか工夫が必要です。筆者は、これを下のような半角の英数字 6 文字で設計しました。例えば ”**BJP123**” です。これは、その切手に関するファイル名などに共通して使うことにしました。この命名規則は下のように決めました。

表 5.1 整理番号の付け方

B	切手の関係であることの識別英字です。
JP	日本の切手であることの国名英字 2 字コードです。JIS (ISO) に基づきます。
123	切手発行順に付けた番号です。3 桁の整数を考えれば充分と考えました。

5.3.3 目録サービスに使うデータの整理

インターネットの検索サービスを利用して経験したことは、或るキーワードを入力してヒットする件数が 200 程度までは、なんとか個別の詳細データにアクセスしますが、それ以上になると面倒です。順位が下位になるものは、結果として参照もれになります。この数を絞り込む本格的な方法は、SQL などの助けを借りて、複数のキーワードを使う論理積の演算です。画像の場合、画像の特徴を言葉に直さなければなりません。しかし、それよりも、複数のサムネイルの一覧を示す目録の画面から、目的の画像を撰んでクリックすれば、そこから詳細データにアクセスできるようにすると使い易くなります。橋の切手の場合、本格的なデータベースの形式にこだわる前に、単純な作業で検索ができる目録を EXCEL のワークシートに作成することから始めます。表の行方向の項目は、大きく 2 種類に分けます。一つは切手固有の情報であって、切手のカタログから転載します。二つは、橋の情報です。これは、橋の名前と架設位置などをキーワードにして、別のデータベースにアクセスする情報を載せます。データ項目を欲張りたくなるのですが、これは、例えば、藤井邦夫資料にアクセスする情報にします。ドイツ人との交流を考えると、並列して英訳が必要になりました。橋梁名や地名などの固有名詞は、読みが分からないこともありますので、フリガナに代えてローマ字表記を併用することが役に立ちます。

5.3.4 二種類の画像データを編集する

インターネットを介して、画像データを家庭のパソコンでも閲覧できるようになったのは、2000 年、デジタル通信技術 **ADSL** (asymmetric digital subscriber line) が一般電話回線で利用できるようなったこと、さらに、光ファイバー利用技術が普及してきた 2004 年以降、大きな進展が見られました。実践的なインタフェースとして、小さな画像 (サムネイル) を案内表示にしておいて、それをクリックすると、大きな寸法の画像を表示させる方法です。これに対応するためには、元の画像から二種類の画像データを準備します。通信回線を介して効率的にデータを送受信し、パソコンのモニタで閲覧することを考えて、一般的な画像データを、筆者は下のようになに決めました。

観賞用画像の最大横寸法	640 ピクセル以下
最大縦寸法	480 ピクセル以下
サムネイルの画像寸法	91×91 ピクセル

A4 版のレポートに使う場合、普通の写真データとしてプリントして見やすい画像寸法は、**キャビネ判** (縦 11.5×横 16.5cm)、または一回り小さめの**手札版** (7.5×11 cm) です。インターネットエクスプローラの表示画面をプリンタに出力して利用することも考えて、上の最大ピクセル数を決めました。

5.3.5 ユーザが利用する方法

一般ユーザが橋の切手を見たいときは、何かの手段でウェブサイトのアドレスを探すことから始めます。筆者の切手情報は、差し当たり中日本建設コンサルタントのホームページに載せてあります。

中日本建設コンサルタント株式会社：

http://www.nakanihon.co.jp/gi_jyutsu/Shimada/stamp/BridgeStamps.html

何段階かのツリー構造がありますが、図 5.3 に例示した目録画面が親画面です。この目録のサムネイル画像をクリックすれば、その詳細情報の画面にリンクするようになっています。図 5.6 は、厩橋（うまやばし：BJP123 をヒットしたときの、観賞画面です。これらの表示に使う閲覧ソフトはインターネットエクスプローラ（IE）です。IE の機能として、表示画面のプリントができますし、HTML ファイルのソーステキストも見ることができます。これとは別に、全体を HTML-Help に編集したファイルも準備してあって、ファイル名は StampJapan.chm です。ファイルの寸法は約 9 MB です。このファイルは、教育利用を考えて作成したものです。インターネットと接続していないパソコンでも閲覧できますが、閲覧ソフトに IE は必要です。このファイルをインターネットのオフライン環境で使うときは、プリンタへの出力も、またソーステキストも見ることにはできません。HTML-Help に作成する目的は、このコンパイラがソースデータとする HTML ファイルの文法エラーを指摘してくれますので、ウェブサイトに登録する前のテキストの校正用ツールとして筆者は使っています。

なお、橋の名前から切手の画像にアクセスするようにした目録の表示画面を図 5.5 に示しました。こちらは、切手の識別番号をクリックするように設計してあります。実は、この目録には、橋の名前が分からないデータのリストが抜けています。また、橋の切手の紹介が目的ですが、筆者の専門からの興味として、元になる切手には塔構造物も含めてありますし、天橋立のような自然地形も入れてあります。そこで、全体を通して、順に閲覧できるように、「前のページ；次のページ」のリンクを付けてあって番号順に画像を観賞できるようにしてあります（図 5.5）。これは、手の込んだ作業を必要とします。



図 5.3 サムネイルを使った切手画像の目録画面



図 5.4 橋名からアクセスすることを目的としてテキスト形式で表した切手の目録画面

5.3.6 データの管理者側の資料整理

橋を図柄に持つ切手を画像データとしてデータベース的な利用では、この元資料の中身をユーザ側が変更する使い方をしません。しかし管理者側では、新しい切手の情報を追加することを考えておきます。上で例示した筆者の方法、特に HTML-HELP のファイル形式に編集することは、すべての HTML ファイルをあらかじめ作成しておいてコンパイルに利用しますので、手が掛かり過ぎます。したがって、データの追加作業が必要になるファイルの数を減らし、クライアント向けの送信用ファイルを自動的に作成する方法があると助かります。これは、クライアントの検索要求が有る度に、動的に HTML ファイルを作成しますが、これを保存しません。これを目的としたソフトウェアの開発が、第 4 節の最後に紹介した ASP です。この作業に使うファイルは、ここでは、①EXCEL で作成したワークシート、②元にする画像データファイル、③そのサムネイルの画像データファイルです。ASP で作成するファイルは、図 5.3 ~ 図 5.5 です。手作業でこれらのファイルを作成するときは、元データの EXCEL のワークシートから、列の項目を並べ変えたうえで、コンマ区切りのテキストファイル (CSV 形式) に変更し、そのレコードを HTML ファイルに組み込みます。作業の方法は、第 3 章で解説した目次と索引の作り方と本質的に同じです。図 5.4 は、橋の名前から目録を作成したのですが、この他に、下の表 5.2 のような見方で索引を作成することを設計すれば、クライアント側には親切なソフトになるでしょう。

表 5.2: 橋の写真の索引に使う分類

全体索引	切手の発行年月日順に並べた全リストです。
サムネイル目録	縮小した図柄 (サムネイル) を並べて索引にしたものです。
橋名による索引	名前が分かっている場合です。名前の分からない場合が以下の項目です。
鉄道橋	図柄の主題は機関車や車両ですが、橋を点景として含む場合です。
公園と橋	日本庭園の図柄などには、橋が描かれていることが多く見られます。
塔と建物	塔は建物や橋脚にも使われます。構造力学的な見方をするため含めます。
記念切手	図柄に、絵画の中の橋が描かれているような場合も含めます。
ふるさと切手	主に都道府県の観光地案内に発行されたものです。
その他、写真切手など、	ふるさと切手よりも、やや私的な意義で発売された切手です。

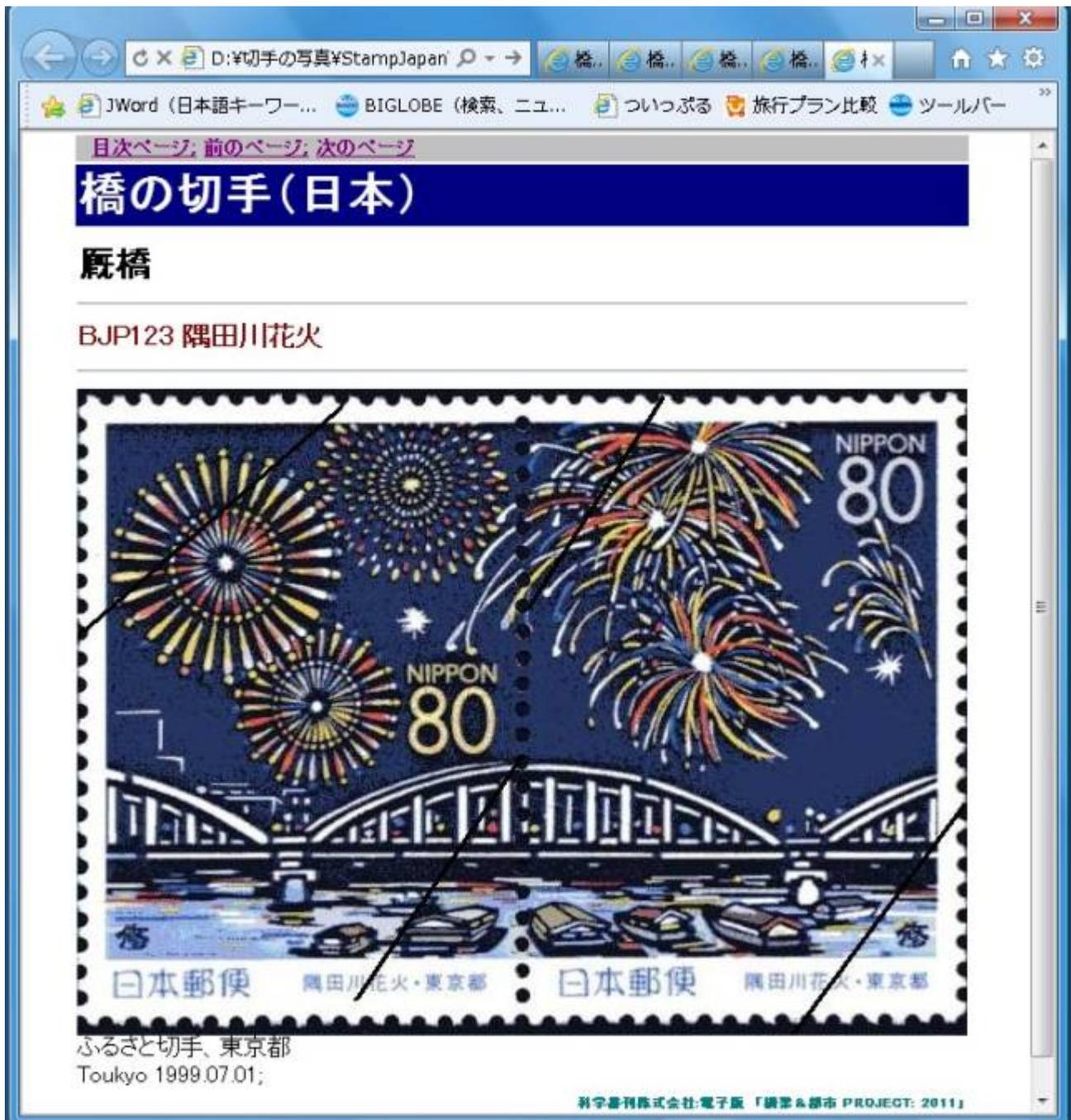


図 5.5 橋の切手の WEB 上の閲覧画面 (例)

終わりに

結論から言うと、本格的なデータベースを、個人のパソコン環境で構築し、閉鎖的に使うことは無理があります。もともと、データベースは、複数の人が利用することを考えたツールが目的だからです。したがって、どこか、共用できる環境にデータベースシステムを置いて、共同利用ができることが理想です。そのためには、同じ目的意識を持った利用者がデータを持ち寄ってデータ量を増やし、システムエンジニアの協力が得られる共用個所の運営が必要です。そのシステムを構築するには、データベースについて、共通した理解があることが望まれます。その中の、かなりの知識が文字処理、また言語学的な常識にあります。そのことを解説した参考書は少ないので、教育利用を考えて、この報文をまとめることにしました。