文書（テキスト・データ）のデータベース化についての研究
——パソコン通信ログ・データベースを例として——

真田　克彦*，山下　陸夫*

(1993年10月15日　受理)

Investigation of Constructing Database of Documents (Text Data) As an Example, Database with Log of Telecomputing

Katsuhiko SANADA, Mutsuo YAMASHITA

研究の目的と概要

近年ワープロの普及はめざましいものがあり，それを利用して文書がどんどん作られ，ディスクに電子情報として保存されている。さらに，ネットワークの発達と普及により，それを利用しての文書の交換も盛んに行われている。このようにして蓄積される文書の量は，個人においても，組織においても，膨大なものとなってきている。そのため，蓄積された文書をデータベースとして保存しておき，いつでも必要な文書を検索して取り出せるようにしておくことは，今後非常に重要な問題となるであろう。

しかしながら，従来のリレーショナル・データベースは，必ずしも文書の蓄積と管理を得意とはしていない。そのデータ構造はリレーションという一種の表構造に限定されており，文書のような可変長で構造が一定でなく複雑なデータを扱うには適していない。

さらに，従来のデータベースの特徴は，汎用的なデータをできるだけ多くの利用者に共有してもらうことにより，データベースの経済性を追求してきた。データが多くの利用者に共有されるために，高い頻度でのデータの更新処理が要求され，その時点での最新の情報を管理することが第一の使命とされてきた。このようなリレーショナル・データベースの基本思想もまた，文書をデータとして保存することと相容れないものと考えられる。

文書のデータベースは，文書作成や研究・調査など，創造的かつ少人数による作業を支援することが想定される。また，データの更新の頻度はそう高くなく，過去のデータを繰り返し参照することも多い。しかし，更新作業では，大量のデータの処理が必要である。

*鹿児島大学教育学部数学科
**川内職業能力開発短期大学校

また，従来のデータベースは汎用コンピュータによる集中処理を想定しているが，文書のデータベースでは，少人数の利用を想定しており，ワークステーションやパーソナルコンピュータによる分散処理が適している。

本論文では，第1部で，リレーショナル・データベース・システムによる文書（テキスト・データ）のデータベースを構築する方法について検討するために桐データベース・システム*)を用いて，パソコン通信のログ（log）のデータベースを構築し，その方法や問題点などを明らかにした。

第2部では，最近注目されているオブジェクト指向データベースは，複雑なデータ構造を扱うことを目的としており，分散処理アーキテクチャに適している。このような特徴が，文書のデータベース化に適していると思われるので，オブジェクト指向データベースによる文書のデータベース化の可能性について検討した。

第1部　パソコン通信ログファイルのデータベース構築

1　はじめに

パソコン通信人口も既に150万人を突破し，双方向のニューメディアとして定着している。これにともなって各通信センターに書き込まれたメッセージの量は膨大なものになっている。この蓄積されている情報を有効に利用する手段や方法の開発が急がれる。

NIFTY のプログラム言語フォーラム（FPL）の BASIC の部屋のデータライブラリのログファイルを利用してデータベースの作成を検討中である。現在までにログファイルを利用したデータベースの3つのプロトタイプを作成した。パソコン通信でダウンロードしたファイルから，jgawk を利用してファイル変換処理を行い，約2500個のタイトルリストデータベースを短期間に作成できた。

2　データベース作成の基本的な考え方と使用パソコン

2.1　ログファイルによるデータベース作成の意義

パソコン通信で蓄えられたメッセージのデータベース化には次のような意義が考えられる。

①パソコン通信のメッセージは増加する一方であり，利用方法の開発による効果が期待できる。

②メッセージはコンピュータ入力済みであり，入力なしにコンピュータ利用可能なデータである。

③対象にした電子会議室のメッセージは，Q&A 的なやり取りの中での情報であり，教育的に利用可能な多くの情報が含まれている。これらの情報を有効活用する方法を開発することは有益である。

④情報社会でデータベースを作成することは，個人レベルでもコンピュータの活用を図る観点からも大切であると考えられ，データベース作成実習のカリキュラム開発や社会人を対象にした公開講座に利用可能である。

---

*)桐データベース・システムは，管理工学研究所の商標である。

2.2　データベース作成の考え方

情報化の進展により個人レベルでもデータベースを作成して，活用することが望まれる。身近で，手軽に入手可能な情報の一つにパソコン通信で得られるログファイルがある。

作成にあたっての基本的な考え方は次の通りである。

①当面はデータベース用のソフトは作成することなく，普及している市販のソフトを利用する。（将来は，簡単なものを作成することも考える必要がある。）

②ハードウェアについても汎用のパソコンを対象とする。

③作成手順を簡単化して，出来るだけ自動化を目標にする。このためにテキストファイル処理を行う sed, jgawk 等のフリーソフトウェアを活用する。

④データベースが作成できたら，キーワード等で該当するメッセージを検索した後，メッセージが読み出せる機能を持たせるように考える。

2.3　作成に使用したパソコンシステム

作成の考え方に基づいて，使用したハード及びソフトの概要は次の通りである。

利用対象パソコン：PC-9801シリーズ
（実際には桐を利用できる PC-9801NS/E を使用）

データベースソフト：国内で開発・市販されている「桐」(Ver 4.0)

その他の利用ソフト：テキストファイル変換等に利用
フリーソフトウェアを主体とした基本ソフト
FD（ファイル管理ソフト）
TED（エディタ）
WXP（仮名漢字変換のフロントエンドプロセッサ）
sed 及び jgawk

使用システムは，身近にある利用可能なハードウェア及びソフトウェアを利用している。データベースソフトの選定にあたっては特別な選定理由は存在しない。

3　パソコン通信ログファイルによるデータベースの作成

3.1　作成の目的

蓄積された情報の有効利用のためのデータベースの作成方法を確認し，その活用法を検討する。当面の対象は NIFTY の FPL のデータ資料（BASIC の部屋）によるプロトタイプのデータベースの作成を目標とする。作成対象としたログファイルは BASIC に関する電子会議録であるので，BASIC に関する学習支援データベースとして利用可能であると考えられる。

3.2　NIFTY のログファイルとデータベースの読み込みファイル形式

NIFTY の電子会議室のログの形式は図１－１のようになっており，データベースソフト「桐」が読込み可能なファイル種別は「テキスト」と「K3 フォーマット」である。これらの形式のデー

タのサンプル例を図１－２に示している。ログファイルをこれらの形式に変換する必要がある。今回は変換作業が容易なテキスト形式を採用した。図１－３に変換後のテキストファイルの例を示している。

パソコン通信で得られたログをもとに，データベースの項目等に合わせて，読込みデータのテキストファイルの変換処理を行う必要がある。該当する項目にデータがない場合でもデータを読み込んだとき項目がずれないようにコンマ（,）で区切っておくことが大切である。

```
001/299   SD100379  きだ  あきら        この会議室は
( 3)   93/03/22 01:48                    コメント数：1

  BASIC 言語と、そのプログラミングに関連した話題を取り扱います。

  Pascal にも負けず、C にも誘惑されず、Modula-2 の厳格さも、perl の
チャランポランさも気にせず、ひたすら孤高の道を歩んでいた BASIC も、
最近になって Visual BASIC という新参者とイベントドリブンという新しい
スタイルによる改革の波が迫りつつあるようですが…。

  とにかく BASIC の話は、この部屋までどうぞ。

                                SD100379 (FPL)  きだ  あきら

電子会議 (1:発言  2:コメントを読む  改行のみ: 読む) 通常モード
>
002/299   TAB01427  Harry              RE:この会議室は
( 3)   93/03/23 21:04  001へのコメント

ってえわけで。
何とWindowsの動く環境もないくせにVisual basicを買ってしまいました。
さあ、動くましんを買うぞ!
```

図１－１　NIFTY の電子会議室のログ

```
001,tak,データベースの利用法,
002,yama,データベースの説明,001
003,yuki,データベースの入力法,002
004,ame,データ活用法,001
005,mika,データの再利用,001
006,yama,データベースの説明（2）,002
007,yuki,データベースの入力法（2）,003
008,ame,データ活用法（2）,004
009,yama,データベースの説明（3）,006
010,yuki,データベースの入力法（3）,007
011,ame,データ活用法（3）,008
```

a）テキスト形式

```
"001","tak","データベースの利用法",
"002","yama","データベースの説明","001"
"003","yuki","データベースの入力法","002"
"004","ame","データ活用法","001"
"005","mika","データの再利用","001"
"006","yama","データベースの説明（2）","002"
"007","yuki","データベースの入力法（2）","003"
"008","ame","データ活用法（2）","004"
"009","yama","データベースの説明（3）","006"
"010","yuki","データベースの入力法（3）","007"
"011","ame","データ活用法（3）","008"
```

b）K3 形式

図１－２　データーベース読み込み可能ファイル形式

```
001/299,SD100379,きだ　あきら,この会議室は,( 3),93/03/22 01:48,コメント数：1,BAS
IC,プログラミング,Visual BASIC
002/299,TAB01427,Harry,RE:この会議室は,( 3),93/03/23 21:04,001へのコメント,Visua
l basic,
003/299,PFA03560,鮫提督,はじめましてのごあいさつ,( 3),93/03/23 22:13,コメント数
：1,ＴｒｕｅＢＡＳＩＣ,Ｂａｃｋ　ｔｏ　ＢＡＳＩＣ,ポダートマス大のケメニーとカー
ツ
```

図１－３　変換後の図１のテキストファイル

3.3　データベース作成手順と項目の決定

パソコン通信のログファイルを利用したデータベースの作成手順は次の通りである。データ入力の手数は省略できるが，データの変換作業が問題となる。

ア）データベースの項目の決定

イ）データベース用ファイルへの加工・編集

ウ）データの読込とデータベースの作成

エ）作成データベースの試用と評価

（1）データベース項目の決定

図１－１に示すログを参考にすると，データベースの項目の候補としては次の項目が挙げられる。

ア）メッセージ番号，イ）ID，ウ）ハンドルネーム，エ）タイトル，オ）登録日時，

カ）会議室番号，キ）関連メッセージ番号，ク）コメント数，ケ）キーワード，コ）メッセージ

ア）～コ）の項目から必要に応じて適切な項目を選定してプロトタイプのデータベースの項目として定めた。試作した各プロトタイプの項目は次の通りである。

プロトタイプ１　上の項目の候補からメッセージを除いた項目で作成

プロトタイプ２　上の項目の候補からキーワードとメッセージを除いた項目で作成

プロトタイプ３　上の項目の候補からキーワードを除いた項目で作成

（実際の作成はタイプ２にメッセージを結合して作成している）

キーワードは，メッセージを実際に読んで本文中から適当な語句を選定してキーワードとしており，その選定にはかなりの時間と労力を必要とする。今後は，メッセージとキーワードとの関係を更に検討する必要がある。また，メッセージの長さは一定ではなく，長いメッセージは分割して

データベース化する必要がある。データベース項目の決定にはメッセージの長さを考慮する必要がある。

（2）メッセージ番号の付け方

NIFTY (FPL) の BASIC の部屋のこれまでのメッセージは約500個毎に整理され，データライブラリに格納されている。現在，既に約2500個のメッセージが5回に分けられて分割保存されている。保存されたメッセージは各回とも001から約500までの番号が付けられており，データベース化にあたっては，次の例に示すように格納回数を上位1桁に1～5の番号で付加して全体のメッセージ番号を付けている。メッセージを項目に持つプロトタイプ3では，メッセージが長いものはメッセージを分割して作成したので，その分割番号を下位に付加している。

例　3302　3回目に格納された302番のメッセージ

　　11502　1回目に格納された150番のメッセージで分割された2番目のメッセージ

3.4　読み込みファイルの加工・編集

使用したデータベースソフト「桐」では，1レコードは1000文字以内と制限されている。メッセージを項目にいれる場合にはこの点を考慮すべきである。以下に各プロトタイプのデータベース用読込ファイルの編集・加工について述べる。

（1）プロトタイプ1の場合

ログファイルをエディタ (TED) に読み込み，手作業で区切り記号の挿入及びキーワードの選定を行い，読込用ファイルを編集・加工した。編集作業及びキーワードの選定を手作業で行うにはかなりの労力が必要である。

（2）プロトタイプ2の場合

sed でタイトルリスト一覧の部分を切り出し，区切り記号の半角のスペースをコンマ (,) に変換し，更にメッセージ番号に分割のブロック番号を付け加える作業を jgawk を利用して行って読み込みファイルの編集・加工を行った。項目で空白の場合には項目がずれないように区切り記号を必ず挿入するように注意が必要である。

（3）プロトタイプ3の場合

ログファイルからメッセージ番号とメッセージのみを切り出して，メッセージ本文を1項目となるように擬似改行コードを挿入して編集作業を行う。メッセージが長くて1項目の範囲を超える場合は分割してメッセージ番号の最下位に分割番号を付加する。メッセージ番号とメッセージのデータベース表を作成し，プロトタイプ2と結合してプロトタイプ3を作成した。

メッセージの編集は現段階では手作業で行っており，エディタの編集機能を利用して「擬似改行コード」を挿入している。メッセージの編集を手作業で行うにはかなりの労力が必要なので，今後編集作業を改善する余地がある。

ここで「擬似改行」とは，エディタなどで1行の長さに制限がある場合に，擬似的な改行コードを加えることによって1行とみなすようにすることを言う。使用した TED の場合には “CTRL”

キーと“O”キーを同時に操作すると挿入される。

3.5　表形式と帳票形式

「桐」によるデータベースでの入力・表示・編集形式には，表形式と帳票形式があり，表形式は各メッセージ1行しか表示できない。したがって，メッセージを表示させる場合にはカード型の帳票形式に表示を切り替える必要がある。プロトタイプ3では，帳票形式の表示も可能となっている。帳票形式画面と表形式画面との切り替えは“HOME／CLR”キーを操作する。

帳票形式の表示を行う場合のファイルの作成手順は次の通りである。

①表形式のデータベース表を作成する（この表を表示対象表と言う）。

②帳票の定義を行い，帳票定義ファイルを作成する。表示するには，上で作成した表示対象表と帳票定義ファイルとを使用して帳票形式の表示を行うことができる。

3.6　sed, awk 言語によるファイル編集作業の簡単化

プロトタイプの作成に際してはフリーソフトウェア sed と jgawk を利用した。特に，プロトタイプ2では jgawk を用いて，約2500件のタイトルデータベースを短期間に作成できた。

sed とは Stream EDitor の略であり，標準入力（又は指定されたファイル）から1行ずつ読み込み所定の編集作業を行い，標準出力（又は指定されたファイル）に結果を出力する。awk は A. V. エイホ，P. J. ワインバーガー，B. W. カーニハンの3人によって，sed の機能を強化し扱いやすくしたフィルタとして設計され，「awk」とは3人のイニシャルである。基本的にはテキストファイルを入力し，それをあらかじめ作成されたスクリプトに従って操作してその結果のテキストファイルを出力する。

sed はログファイルからタイトルリストの切り出しに使用した。その入力ファイルを図1－4に，sed で処理された出力ファイルを図1－5に示している。jgawk はログファイルのタイトルリストをデータベースに読み込み可能なファイルに編集するのに使用した。jgawk のスクリプト例を図1－6に示し，図1－5を入力ファイルとしたときの jgawk で処理した出力ファイルを図1－7に示している。

現在，メッセージファイルの編集を簡単化する方法を検討中であり，将来はパソコン通信のログ

```
このファイルに収録されている発言の時期は 87/12/20-88/08/23 です。

編集：SDI00379　きだ　あきら

001 SDI00232 87/12/20 00:12       1 ＢＡＳＩＣについて話しましょう
002 SDI00147 87/12/24 00:12 001   1 あのＨｕＢＡＳＩＣは何処へ？
003 SDI00232 87/12/24 23:00 002   1 Hu-BASICですか、なつかしい
004 PAF00600 87/12/27 13:09 003   1 はじめまして！
005 SDI00232 87/12/27 22:56 004     Ｓｏさん、いらっしゃい
006 PAG00076 88/01/03 00:12       1 QUICK-BASICさわってみました。
007 SDI00232 88/01/04 21:25 006     コンパイラは変わりました
続き（改行で表示　E:終了）
>
008 PAG01551 88/01/19 13:36       1 ＭＳ－ＤＯＳ　ＢＡＳＩＣのＬＯＡＤのしか
009 SDI00232 88/01/19 23:15 008   1 何をお使いですか？
010 PAG01551 88/01/21 08:21 009   1 ９８についてきたＮ８８－ＢＡＳＩＣです。
```

図1－4　sed 入力ファイルの内容

```
001 SDI00232 87/12/20 00:12       1 ＢＡＳＩＣについて話しましょう
002 SDI00147 87/12/24 00:12 001   1 あのＨｕＢＡＳＩＣは何処へ？
003 SDI00232 87/12/24 23:00 002   1 Hu-BASICですか、なつかしい
004 PAF00600 87/12/27 13:09 003   1 はじめまして！
005 SDI00232 87/12/27 22:56 004     Ｓｏさん、いらっしゃい
006 PAG00076 88/01/03 00:12       1 QUICK-BASICさわってみました。
007 SDI00232 88/01/04 21:25 006     コンパイラは変わりました
008 PAG01551 88/01/19 13:36       1 ＭＳ－ＤＯＳ　ＢＡＳＩＣのＬＯＡＤのしか
009 SDI00232 88/01/19 23:15 008   1 何をお使いですか？
010 PAG01551 88/01/21 08:21 009   1 ９８についてきたＮ８８－ＢＡＳＩＣです。
```

図1－5　sed で処理した出力ファイルの内容

```
# 名前:ktlforml.awk
# 機能:LOG TITLE　にNo.やコンマ,を入れて出力するスクリプト
# 使用法:jgawk -f ktlforml.awk infilename > outfilename

 length($5)<3 { printf ("1%s %s %s %s %s %s %s %s %s %s %s %s%¥n",$1 "," $2 "," $
3 ",," $5 "," $6$7$8$9$10$11$12)}
 length($5)>3 { printf ("1%s %s %s %s %s %s %s %s %s %s %s %s%¥n",$1 "," $2 "," $
3 ",,," $5$6$7$8$9$10$11$12)}
 length($5)==3 { if (length($6)==1)  printf ("1%s %s %s %s %s %s %s %s %s %s
%s%¥n",$1 "," $2 "," $3 ",1" $5 "," $6 "," $7$8$9$10$11$12)
                 else { printf ("1%s %s %s %s %s %s %s %s %s %s %s %s%¥n",$1 ","
 $2 "," $3 ",1" $5 ",," $6$7$8$9$10$11$12)}
            }
```

図1－6　igawk のスクリプト例

```
1001,SDI00232,87/12/20,,1,ＢＡＳＩＣについて話しましょう
1002,SDI00147,87/12/24,1001,1,あのＨｕＢＡＳＩＣは何処へ？
1003,SDI00232,87/12/24,1002,1,Hu-BASICですか、なつかしい
1004,PAF00600,87/12/27,1003,1,はじめまして！
1005,SDI00232,87/12/27,1004,,Ｓｏさん、いらっしゃい
1006,PAG00076,88/01/03,,1,QUICK-BASICさわってみました。
1007,SDI00232,88/01/04,1006,,コンパイラは変わりました
1008,PAG01551,88/01/19,,1,ＭＳ－ＤＯＳ　ＢＡＳＩＣのＬＯＡＤのしか
1009,SDI00232,88/01/19,1008,1,何をお使いですか？
1010,PAG01551,88/01/21,1009,1,９８についてきたＮ８８－ＢＡＳＩＣです。
```

図1－7　igawk で処理した出力ファイルの内容

ファイルから簡単にデータベースの読み込み可能ファイルに変換できる方法を開発すべきである。

4　試作データベースの概要

現在，３つのプロトタイプのデータベースを試作して評価を行った結果，最終的にはプロトタイプ３にキーワードを付加したものを作成する方が望ましいと考えている。以下に作成した３つのタイプの概要を述べる。

(1) プロトタイプ１

キーワードの項目を持っていることが特徴である。しかし，初めて作成したタイプであることもあり，図１－８に示すようにデータに冗長がある。また，キーワードの選定にかなりの労力を要するので，メッセージの項目を持つプロトタイプ３で，データベースの機能を利用して，直接メッセージを検索する方法が，容易にデータベースの作成ができるものと考えられる。ハンドルネームや会議室番号は省略可能である。

:メッセージ:	ID	:ハンドル:	タイトル	:会議室:	日時	:閲覧(後:	コメント数	:	キーワード	:
001/299	SD100379	きだ あきら	この会議室は	(3)	93/03/22	0		コメント数：1	BASIC プログラミング Visu	-BASIC
002/299	TAB01427	Harry	RE:この会議室は	(3)	93/03/23	2	001へ		Visual-basic	
003/299	PFA03560	鯨提督	はじめましてのごあいさつ	(3)	93/03/23	2		コメント数：1	TrueBASIC Back-to-BASIC 米	トマス大のケメニー
004/299	TAB01427	Harry	QBについての質問	(3)	93/03/24	1		コメント数：1	QBのEOF QBのユーザー定義型の変数の引き渡し	
005/299	JAD00247	J.T.	QBのEOFについて	(3)	93/03/25	1	004へ	コメント数：2	QBのEOF QBでは最終レコードでEOFは偽 次のレコードをGETした	
006/299	HGD03067	じゃん	RE:QBのEOFについて	(3)	93/03/25	2	005へ		シーケンシャルファイルのEOFは最後のデータを読んだ時点で真	
007/299	TAB01427	Harry	RE:QBのEOFについて	(3)	93/03/26	0	005へ	コメント数：1	QB C EOF	
008/299	SD100379	きだ あきら	RE:QBのEOFについて	(3)	93/03/26	2	007へ	コメント数：1	BASIC QB EOF シーケンシャルファイル ランダムファイル EO	
009/299	HGD00201	英斗恋	EL_BASIC ver.1.53Tの不具合について	(3)	93/03/27	0			EL_BASIC 登録	
010/299	TAB01427	Harry	RE:QBのEOFについて	(3)	93/03/27	0	008へ		QB 書き方	
011/299	NBA00671	Nydela	これはこれは	(3)	93/03/27	1			新しい部屋の開設 VBDOS イベント・ドリブン	
012/299	QGA01641	きゃろ	ＶＢがきました、けど(･_;)	(3)	93/03/27	1		コメント数：1	ＶＢ-for-DOS ＶＢのハング	
013/299	TAB01427	Harry	RE:ＶＢがきました、けど(･_;)	(3)	93/03/27	1	012へ	コメント数：2	VB	
014/299	QGA01641	きゃろ	ＶＸはだめ？！	(3)	93/03/28	1	013へ		ＤＡ ＶＸ ２８６ ＶＭ VBDOS	
015/299	QGA01641	きゃろ	原因はわかった	(3)	93/03/29	1	013へ		VBDOS メモリドライバとの相性 NEC純正のEMM386.SYS	
016/299	HBA00270	Still	初めまして！ さっそく質問なんですが	(3)	93/03/29	1		コメント数：1	QBのPalette文 QuickBasic	
017/299	HGD02533	めんたんぴん	８６ＢＣ、注文しました	(3)	93/03/29	2			FA(factory-automation)のシステム設計 QB VB	
018/299	HBA00270	Still	RE:初めまして！ さっそく質問なんですが	(3)	93/03/29	2	016へ	コメント数：2	Cで使うARGV(0)の(自分自身の実行ファイル名をフルパスで所得)QB	
019/299	GFC01740	あちゃ	ＢＡＳＩＣの定義って…？	(3)	93/03/31	0		コメント数：2	QuickBASIC TrueBASIC N88-BASIC X6800	
020/299	QFF03453	坂田　靈	QBの質問／文字列の文字抽出方法？	(3)	93/03/31	0		コメント数：2	QBの決まった文字列の抽出 文字列が入力された変数A$から、半角英数字の	
021/299			メッセージなし							
022/299	TAB01427	Harry	RE:QBの質問／文字列の文字抽出方法？	(3)	93/03/31	2	020へ		QB CSNG$関数	
023/299	NAC03056	玉井　孝	届いたぞぉ	(3)	93/04/01	2			ＴＫＷ－８６ＢＣ Ｖｅｒ５．５２	
024/299	SD100379	きだ あきら	RE:初めまして！ さっそく質問なんですが	(3)	93/04/02	0	018へ	コメント数：1	Cのargv[0]に相当するものを得るプログラム	
025/299	SD100379	きだ あきら	RE:ＢＡＳＩＣの定義って…？	(3)	93/04/02	0	019へ	コメント数：2	BASIC の定義 各BASICに共通している点、代入記号が「=」	
026/299	GFC01740	あちゃ	RE:ＢＡＳＩＣの定義って…？	(3)	93/04/02	1	025へ		戦略的 意識した他言語に敏べ、分かりやすいように作ったと言える意味でBA	
027/299	HBA00270	Still	発言者削除	(3)	93/04/02	1	025へ			
028/299	HBA00270	Still	RE:初めまして！ さっそく質問なんですが	(3)	93/04/02	1	024へ		統合環境QB.EXEを実行プログラムではその名前を得る事ができました 設	
029/299	PDD01714	とりちゃん	86BC ver 5.52 ・・・	(3)	93/04/02	2				
030/299	PDD01714	とりちゃん	BASICとは？	(3)	93/04/02	2		コメント数：2	文字変数を***$で扱う PRINT FOR IF GOTO GOSUB	
031/299	HGD03067	じゃん	RE:ＢＡＳＩＣの定義って…？	(3)	93/04/03	0	019へ		ベーシックは、教育用にできたもの 変数を宣言しなくても使えるのはベーシッ	
032/299	NBA00671	Nydela	コマンド文字列の取得	(3)	93/04/03	0	018へ	コメント数：2	MS-DOSの3.3かそれ以降であれば、入力した文字列は環境変数領域に保	
033/299	HBA00270	Still	RE:コマンド文字列の取得	(3)	93/04/03	1	032へ	コメント数：1	PSPのセグメントアドレス所得 int21-62HはDOS-Ver-3.	
034/299	PAC03057	KITA	RE:コマンド文字列の取得	(3)	93/04/04	0	033へ		コマンドライン引数 QB45の標準関数 >QB/LGENで起動	
035/299	PDD01714	とりちゃん	re:int21 62H を使わずに実行プログラム名	(3)	93/04/04	0			comファイル起動時のコードセグメント値がPSP exeファイル起動時の	
036/299	SD100379	きだ あきら	発言者削除 :誤字修正	(3)	93/04/04	0	030へ	コメント数：1		
037/299	SD100379	きだ あきら	RE:BASICとは？	(3)	93/04/04	0	030へ		文字変数を***$で扱うことはしないBASIC PRINT FOR IF	
038/299	SD100379	きだ あきら	RE:コマンド文字列の取得	(3)	93/04/04	0	032へ		DOSの3.XXと 2.XX #24と同じ原理でargv[0]をセットし	
039/299	QGB01521	ockeghem	再びBASICの定義	(3)	93/04/04	1	036へ		Visual-Basic にはINPUTがありません Ver2.0以降の	
040/299	FFA00237	Fling Rock	RE:BASICの定義?について	(3)	93/04/05	0			BASASICの略号 Beginer's All-purpose Sym	
041/299	HGD00201	英斗恋	EL_BASICテスト公開期間の終了について	(3)	93/04/05	2		コメント数：1		
042/299	HGD00201	英斗恋	RE:EL_BASICテスト公開期間の終了について	(3)	93/04/05	2	041へ		ver.1.54 複数のCONST変数を一度に宣言できない 文法エラー	
043/299	NBF01226	Ｌａｙ	ようこそ 鯨提督さん	(3)	93/04/06	0	003へ		True Basic	
044/299	GAA01062	快人	BASICでオプション取得・・　快人	(3)	93/04/06	1		コメント数：1	MS-BASIC QBでのオプションの取得する方法	
045/299	GBF00535	Doc Hollid	RE:BASICでオプション取得・・　快人	(3)	93/04/06	1	044へ	コメント数：1	COMMAND$で取得できます	
046/299	GBF00535	Doc Hollid	RE:BASICでオプション取得・・　快人	(3)	93/04/06	1	045へ		ＱＢでの話です ＭＳは知りません	
047/299	HBA00270	Still	文字列の切り出しについて	(3)	93/04/10	0		コメント数：1	漢字を含んだ文字列変数 バイト単位で効率的に切り分ける方法 PSP セグ	
048/299	JAH02230	しげちゃん2	QBとFEPについて、教えてください。	(3)	93/04/11	0		コメント数：1	QB自作のソフト 入力時に自動的にFEPの起動 FEPはATOK7	
049/299	TAB01427	Harry	RE:QBとFEPについて、教えてください。	(3)	93/04/11	1	048へ	コメント数：1	拡張ライブラリ ファンクション	
050/299	TAB01427	Harry	QBでいきなり暴走	(3)	93/04/12	1			QB 巨大プログラム	

図１－８　プロトタイプ１の表形式

(2) プロトタイプ２

ログファイルの中のタイトルリストのファイルを利用して，jgawk を使用し，約2500件のメッセージのタイトルデータベースがある。ID やタイトルで検索等が可能で，作成したプロトタイプ

の中で利用可能なデータベースである。例えば，タイトル中に“QB”を含むものを検索したところ206件選択され，更に“N88”で絞り込むと7件のメッセージがあることが分かった。その結果の出力を図1－9に示している。

	：番号：	ＩＤ	：年/月/日	：関連	：数：	タイトル
1	1197	MGG01464	89/02/08			N88QB第二版
2	1222	MGG01464	89/03/13		2	まだまだ、N88->QBスクリプト
3	1237	PBA00236	89/03/19	1227		N88：QBがOh！PCに載ってます．
4	1471	MHC02162	89/09/30		1	N88→QB コンバータ
5	2274	GAC01147	90/08/11			QB・N88比較マニュアルについて
6	5079	QFF03453	92/07/17		1	N88→QBの変換ソフトを教えて
7	5304	GBF00535	92/11/24	5079		RE：N88→QBの変換ソフトを教えて

図1－9　プロトタイプ2での検索結果例

：番号	：ＩＤ	：年/月/日	：関連	：数：	タイトル	：MI：
1001	SDI00232	87/12/20		1	BASICについて話しましょう	この会議室では、BA
1002	SDI00147	87/12/24	1001	1	あのHuBASICは何処へ？	BASICを皆々と会
1003	SDI00232	87/12/24	1002	1	Hu-BASICですか、なつかしい	HuBASICですか
1004	PAF00600	87/12/27	1003	1	はじめまして！	はじめまして！Soで
1005	SDI00232	87/12/27	1004		Soさん、いらっしゃい	Soさん、いらっしゃ
1006	PAG00076	88/01/03		1	QUICK-BASICさわってみました．	MICROSOFT
1007	SDI00232	88/01/04	1006		コンパイラは変わりました	QUICK-BASI
1008	PAG01551	88/01/19		1	MS-DOS BASICのLOADのしか	Huはヒューでも私は
1009	SDI00232	88/01/19	1008	1	何をお使いですか？	さてLOADの仕方と
1010	PAG01551	88/01/21	1009	1	98についてきたN88-BASICです．	ちと質問の仕方が片手
1011	SDI00232	88/01/22	1010	1	MS-DOSが欲しければ	そういうことですか．
1012	PAG01551	88/01/22	1011	1	BASICでこれだけのことができるとは！	そんなにおっしゃるな
1013	SDI00232	88/01/22	1012		納得（笑）	あ、あれですか（笑）
1014	SDI00232	88/03/03		1	BASIC/98は使えているか？！	BASIC，なんや
10142	SDI00232	88/03/03		1	BASIC/98は使えているか？！	そのときと比べると隔
1015	PBB00556	88/05/03		2	初めまして，ちょっと言わして．	初めまして，この会
10152	PBB00556	88/05/03		2	初めまして，ちょっと言わして．	3）局所的に変数を使
1016	SDI00379	88/05/04	1015	1	プリプロセッサを使ってはどうでしょう？	面白－Toshさん、
1017	SDI00232	88/05/05	1015		そういう時代がありました	MSXは最近動きがあ
10172	SDI00232	88/05/05	1015		そういう時代がありました	コンバータなんていう
1018	PBB00556	88/05/07	1016		皆さんコメントありがとう，がんばります	紀太さん、ヤンさんコ
1019	PBB00507	88/05/08	1014	1	同等品（サブセット）ですが	えと、実はNEWBA
1020	SDI00232	88/05/09	1019		お願いします	OSSIDさんも言語

図1－10　プロトタイプ3　表形式

番号　　1015
ＩＤ　　PBB00556
年/月/日　　88/05/03
関連
数　　2
タイトル　初めまして。　ちょっと言わして。
MI初めまして。　この会議室は９８ベーシックが中心みたいですけど、ちょっとだけ見てやって下さい。　ＭＳＸはベーシックマシンとして見た場合、ちょっと面白い特徴があります。え～・えむえすえっくす～？　なんて言わないでね。　８ビットが低機能なのは誰だって知ってるんだから。ＭＳＸにはＤＯＳがありまして、ＭＳＸ－ＤＯＳと言います。　ま、よーするに、ＭＳ－ＤＯＳのＶ１．２５版なんです、これが。
　ちょっとＭＳ－ＤＯＳと違うのはコマンドにＢＡＳＩＣというのがあることです。
　これは何かと言うと、読んで字のごとく、ＤＯＳからＢＡＳＩＣに移るコマンドなのです。　逆にＭＳＸ－ＢＡＳＩＣには、ＣＡＬＬ　ＳＹＳＴＥＭというコマンドがありまして、これは何かと言うとＢＡＳＩＣからＤＯＳへ移るものなのです。　使って見ると便利でして、めんどくさいファイル操作はＤＯＳ上で、また、ＤＯＳ上で（ＤＯＳ－ＴＯＯＬＳなどで）作ったファイルの加工をＢＡＳＩＣで行えたりするんですね。さて、ＢＡＳＩＣの弱点は有名ですが、１）行番号をつけなきゃいけない。２）ラベルがつかえない。

番号　　10152
ＩＤ　　PBB00556
年/月/日　　88/05/03
関連
数　　2
タイトル　初めまして。　ちょっと言わして。
MI３）局所的に変数を使えない。なんて言うのは致命的ですね。３）はしょうがないですけど、１）、２）は上のコマンドがあるので、使いかた次第です。まず、（ＢＡＳＩＣ付属でなく、もっと便利な高機能の）エディタで入力します。行番号は付けません。ラベルを使います。（このために高機能エディタを使うのです。）　入力したら、セーブ。このファイルは当然アスキーファイルです。ＢＡＳＩＣではこれをシーケンシャファイルと見なして各行に番号（行番号）を付けます。（こういうプログラムはＢＡＳＩＣの初歩的なものですね。）セーブしたらもう一度エディタに戻ってラベルを行番号に「置換」します。これでプログラムの出来上り。　デバッグはＢＡＳＩＣですからお手のものですね、改良も簡単。大体ＣＴＲＬ－ＳＴＯＰで止まる。（暴走しない。）　もっとも、これはコンパイルと同じではないかと、言われそうですが、本物のコンパイル言語に比べれば、デバッグ・改良（手直し）が死ぬほど簡単ですから許してやって下さい。　それになにより、金がかからない！のです。　ま・こんなとこです。　ぐだ聞いてくれてありがとうございました。　Ｔｏｓｈ

図1－11　プロトタイプ3　帳票形式

（3）プロトタイプ3

プロトタイプの中では一番実用的なタイプと考えられるが，作業すべきデータ量が多く，手作業で行っている編集作業の簡単化の方法を検討中である。実験的に試作したデータベースの出力の結果を図1－10，11に示している。表形式と帳票形式との両方の表示画面の機能を持っていることが特徴である。帳票形式の表示画面の文字数からメッセージ項目の文字数が制限されることに留意して，メッセージの分割を行う必要がある。

5　第1部のまとめと今後の課題

パソコン通信のログファイルを使用して3つのプロトタイプのデータベースを作成し，作成法等を検討した。その結果，プロトタイプ3のデータベースが望ましいことが分かった。今後はこのタイプのデータベースを完成させ，利用法や必要な機能の検討を行う予定である。

今後の課題としては

1）汎用性のある作成方法の確立のために，ファイル変換法等の自動化の検討

2）キーワードや検索語の選定方法の検討

3）作成したデータベースの活用法の検討

4）NIFTY の FPL 以外の他のパソコン通信センターのログファイルにも本手法を適用して本手法の有効性を確認すること

などがある。

第2部　オブジェクト指向データベースの可能性の検討

1　はじめに

オブジェクト指向データベースは，知識ベース，マルチメディアデータ，エンジニアリング・データなど，データベース技術の高度応用や情報システムの情報資源管理指向に支えられて急速に発展しつつある。

オブジェクト指向データベース・システムについては，1989年に Bancilhon らが示した「オブジェクト指向データベース（OODB）システム宣言」がよく知られている。しかし，オブジェクト指向データベース・システムの確固たる定義や強固な数学的基盤はいまだ確立されるには到っていない。しかし，少なくともある程度の合意は得られつつあるという段階である。

ここではまず，オブジェクト指向の基礎概念について述べ，さらにオブジェクト指向データベースの条件と概要を述べる。次に文書のデータベース化について検討するために，パソコン通信ログを例にとり，オブジェクト指向データベースの設計について検討する。最後にオブジェクト指向データベースとリレーショナル・データベースの比較検討を行う。

2　オブジェクト指向とは

2.1　オブジェクト指向的アプローチ

オブジェクト（object）とは認知できるもののことであり，実世界のモデル化である。オブジェクト指向プログラミング（Object-Oriented Programming, OOP）とは，オブジェクトを使ってプログラムを作成することであり，実世界をコードの中により直接的に表現することを可能にするプログラミングである。オブジェクト指向データベース（OODB）とは，オブジェクトを記憶し管理することができるデータベース，すなわち，実世界をモデル化し記憶・管理するデータベースということになる。

プログラムは，オブジェクトの動作を表現できるもので，データに対して操作を加えることができる手続きである。データは，オブジェクトの状態を表現するものである。したがって，その立場を代えると，オブジェクトとは，手続きとデータの両方の性質を兼ね備えたものであり，データとそのデータを操作する手続きを合わせ持つものである。

オブジェクト指向の最も重要な特徴の一つは，類似のオブジェクトの集合とそれに付随する演算子の集まりを定義する抽象データ型をサポートしていることである。抽象データ型を定義する言語構成概念をクラスという。クラスによって記述されるオブジェクトの集合に属する要素をインスタンスという。

オブジェクト指向アプローチとは，データをカプセル化するように対象物をモデル化することである。カプセル化とは，データ操作に必要な機能は，すべて手続きの集まりによって提供されていなければならないし，逆に，データを操作するためには，手続きを経由しなければならないモデル化のことである。手続きはデータ型のインタフェースの役割をしており，データは，手続きによって保護されている。

2.2　オブジェクト指向の基本概念

オブジェクト指向の基本概念は次のようにまとめられる。

①オブジェクト（object）

オブジェクトとは，実体（データ）と実体に固有の操作（手続き）の両方を一体化した情報である。オブジェクト内のデータは，オブジェクトに定義されている手続きを通してのみ操作することができる。

②メソッド（method）

オブジェクト中に定義されている手続きをメソッドと呼ぶ。

③メッセージ（message）

オブジェクトのメソッドを実行させるためにオブジェクトに送られる文字列のこと。

④クラス（class）

類似したオブジェクトの共通した特徴を規定したものを，クラスと呼んでいる。

⑤インスタンス（instance）

クラスに属するオブジェクトの具体値をインスタントと呼んでいる。

⑥クラス階層

クラスは階層構造を作ることが多い。あるクラスの上位のクラスをスーパークラス，あるクラスの下位のクラスをサブクラスと呼ぶ。

⑦インヘリタンス（継承，inheritance）

自分のクラスに存在しないメソッドの実行が要求されるとき，上位クラスに定義されているメソッドの実行によって代用することをインヘリタンスと呼ぶ。

⑧マルチプル・インヘリタンス（多重継承）

あるクラスのスーパークラスが複数あり，各スーパークラスからメソッドまたはデータを継承することをマルチプル・インヘリタンスと呼ぶ。

オブジェクト指向システムにおいては，世界は独立なオブジェクトの集合とみなされ，それらは手続き（メッセージ）を通じで相互に通信する。オブジェクトは能動的な実体であり，手続き（メッセージ）は受動的な実体であり，一つのオブジェクトから他のオブジェクトへ引数とともに引き渡される。オブジェクトは受け取った要求を調べ，それに対して動作を起こす。したがって，オブジェクトが中心となるシステムである。

2.3　クラス

クラスは，抽象データ型の構造とともにその演算の定義を合体している。したがって，クラスは抽象データ型を定義するものであって，クラスによって記述されるオブジェクトの集合に属する要素はインスタンスと呼ばれる。

図２－１は，パソコン通信ログを表すクラスの例である。図２－２にはインスタンスの例を示した。

クラスの定義には最小限次の項目が含まれる。

①クラスの名前

②クラスのインスタンスを操作するための外的演算子（インタフェース演算子）

③内部表現

④インタフェースの内部実現

クラスの内部表現は，インスタンス変数に保持される。クラスのインスタンスは，そのクラスに対して“New”という演算子を呼び出すことによって生成される。

例えば，図２－１，図２－２より

B06-0001＝New（LOG, “B06-0001”, “QB の使い方”）

によって，クラス LOG 型のオブジェクト B06-0001 が作られる。

2.4　インヘリタンス（継承）

オブジェクト指向を特徴づける第二の強力な概念は，インヘリタンスである。インヘリタンスを

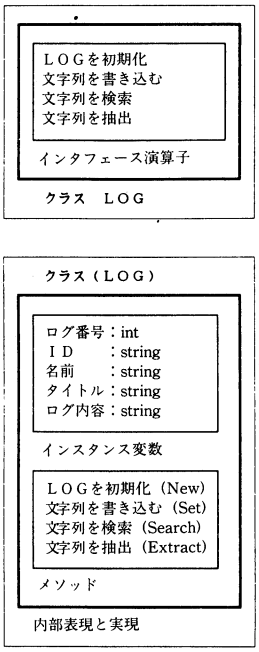

図２－１　クラスとして表されたログ

名前	タイトル
B06-0001	ＱＢの使い方
B06-0002	ＱＢで分割コンパイル
B06-0003	常駐関連ライブラリ
B06-0004	ＢＥＥＰの音程変更
B06-0005	入力について
B06-0006	ファイルの日付を得る
B06-0007	時間の取得について
B06-0008	ＢＡＳＩＣtoC
……	……

図２－２　インスタンスの例

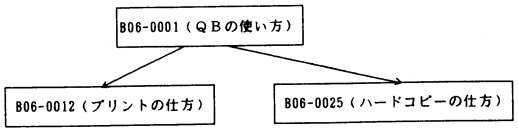

図２－３　継承の階層

用いることにより，特殊化のより少ない既存のクラスの階層の上に，新しいクラスを組み立てることができる。新しいクラスは，その振る舞い（演算，メソッドなど）と表現（インスタンス変数）の両方を既存のクラスから継承することができる。

例えば，図２－３に示すように，クラス B06-0001（QBの使い方）はサブクラス B06-0012（プリントの仕方）と B06-0025（ハードコピーの仕方）があり，サブクラスでは，スーパークラス B06-0001（QB の使い方）のデータや手続きを用いることができる。

3　オブジェクト指向データベース（OODB）

3.1　オブジェクト指向データベースとは

オブジェクト指向データベースは，オブジェクト指向システムをデータベース機能によって強化したシステムということができる。データベース機能は

①高水準の質問言語と基底のシステムにおける質問最適化能力

②持続性と原子的トランザクションのサポート（並行処理制御と回復機能を含む）

③高速で効率的な検索のための複雑なオブジェクトの記憶法，索引およびアクセス法のサポート

であると考えられる。したがって，オブジェクト指向データベースは

オブジェクト指向システム＋①②③

であると定義することができる。

3.2　オブジェクト指向データベースの必須条件

1989年に京都で開催された第１回演繹・オブジェクト指向データベース国際会議（DOOD89）

表２－１　OODB システムが具備すべき条件　（OODB システム宣言，1989年京都）

（１）必須条件：オブジェクト指向関連（８項目）
①複合オブジェクト
②オブジェクト識別子
③カプセル化
④型あるいはクラスがあること
⑤クラスの階層とインヘリタンス機能
⑥オーバーロードとレイトバインディング（遅延束縛）
⑦計算完全性
⑧拡張性

（２）必須条件：DBMS（５項目）
①永続性（persistency）
②二次記憶管理
③並行処理管理
④復旧
⑤簡易問い合わせ機能

（３）オブジェクト指向データベースの追加条件（５項目）
①多重継承
②型チェックと型推論
③分散
④設計トランザクション
⑤バージョン管理

（４）オブジェクト指向データベースの補足条件（４項目）
①プログラミング・パラダイム
②表現システム
③型システム
④均一性

で，Bancilhon らは「オブジェクト指向データベース（OODB）システム宣言」という論文を発表し，オブジェクト指向データベースが具備すべき条件を，オブジェクト指向データベースの必須条件13項目，オブジェクト指向データベースをより使いやすくするための付加条件５項目として発表している（表２－１参照）。

ここでは，文献３）５）を参考にして，必須条件についてのみ述べる。

［１］必須条件：オブジェクト指向関連（８項目）

（１）複合オブジェクト

基本となるオブジェクトを組み合わせて作られたオブジェクトを複合オブジェクトと呼ぶ。オブジェクト指向データベースの目的の一つは，実体（現実のもの）をできるだけ忠実に扱うことであり，基本となるオブジェクトから実体を表現するように複合オブジェクトを構成しなければならない。

複合オブジェクトを構成するためには，基本オブジェクトとオブジェクト構成子が必要である。基本オブジェクトは，親言語の基本データ型（整数型，浮動小数点型，文字型，ポインタなど）である。オブジェクト構成子は，基本オブジェクトまたはすでに定義されている複合オブジェクトを組み合わせて，新たな複合オブジェクトを構成する機能を提供するものである。C＋＋ならば，構造体，配列，クラス定義機能などであり，また集合，バッグ，リスト，キュー，スタックなども提供されることが望ましい。

（２）オブジェクト識別子

リレーショナル・データベースでは，データベースのデータと現実のもの（実体）との対応をキー（一意的に識別できる名前）によって管理している。オブジェクト指向データベースでは，階層構造をもつオブジェクトを扱っており，利用者が見ることができる識別情報（キー）に加えて，

利用者には見えない階層が下位のオブジェクトを探索してデータを取り出すことができるように，識別子（記憶場所と密接に関係するポインタ）を提供する。識別子によりオブジェクトをたどることを，ナビゲーション（navigation）という。

（3）カプセル

データベースにカプセル化を導入する主な目的は，次のようなことである。

①データと手続きを一体として扱う

②オブジェクトの仕様と実現を区別する

③オブジェクトのモジュール性を高める

④データ保護を強化し，アクセス制御を容易にする

カプセル化の具体的方法としては，情報隠蔽（information hiding）が必要である。その理由は次のようなことである。

①オブジェクト内部のデータに変化が起きても，オブジェクト内部の手続きを変更するだけで対処でき，アプリケーション・プログラムを変更する必要はない

②オブジェクト内でエラーが起きたときは，オブジェクト内部の手続きだけを調べればよく，アプリケーション・プログラム全体を調査する必要はない

（4）型あるいはクラスがあること

クラスは，データ型の考えを拡張したものであり，データと手続きの両方が定義されている。またクラスは，インスタンスを生成あるいは操作するときに必要となる情報を含んでいる。

（5）クラスの階層とインヘリタンス機能

クラス階層を使うことにより，共通するデータや手続きは下位のクラスで定義することができるようになる。これにより，対象が分かりやすく表現されるだけでなく，プログラム・コードを短くし，記憶領域を有効に活用することができる。

クラス階層とインヘリタンス機能は，データと手続きの再利用を行うための基本となるものである。

（6）オーバーロードとレイトバインディング（遅延束縛）

オーバーロードとは，一つの記号を複数の意味に使うことである。レイトバインディングとは，実行時にデータと実際に呼び出すべき関数とを結びつけることである。オブジェクト指向アプローチを生かすためには，この2つの機能が必要である。

（7）計算完全性

計算完全性とは，データ操作言語（DML）を用いて，任意の計算可能な関数を表現できることを意味している。これは，既存のプログラミング言語との融合により実現できる。

（8）拡張性

拡張性とは，システムが提供するデータ型と，利用者が定義したデータ型の使い方に何の違いもないことである。拡張性が要求されるのは，技術進歩が速いために，システムをオープンな作りに

し，新しい技術を柔軟に取り入れなければならないからである。

［2］必須条件：DBMS（5項目）

（1）永続性（persistency）

データを処理するプログラムの実行が終了しても，そのデータがシステム内に存続し続ける性質を，永続性と呼んでいる。永続性は，データ型と直交していなければならない。すなわち，どんなデータ型に対しても永続性という性質を付けることができなければならない。また，永続性は暗黙的でなければならない。

（2）二次記憶管理

主記憶と二次記憶とのアクセス速度の差から，データのインデックス付け，クラスタリング，バッファリング，問い合わせの最適化などの機能が必要である。

（3）並行処理管理

データベースは，複数の利用者が同時に使うことを前提としているため，利用者間で調和のとれた処理が行われることを保証しなければならない。並行管理は，データ処理の原子性（atomicity）と直列実行可能性（serializability）に立脚したものである。データ処理の原子性とは，一連のデータ操作がすべて正しく実行されるか，全然実行されないかのどちらかであって，その中間はないことである。直列実行可能性とは，複数のプログラムを並行処理しても，独立に実行したのと同じ結果が得られることである。

（4）復旧

データベースには，高い信頼性が要求される。オブジェクト指向データベースにも復旧機能が必要である。

（5）簡易問い合わせ機能

オブジェクト指向データベースにも，簡易問い合わせ機能が必要である。

4　オブジェクト指向データベースの設計

4.1　オブジェクトの設計とは

オブジェクトはある応用環境において活動する実体の表現であることを認識して，オブジェクトの設計にとりかかる必要がある。そして環境における実体を定義したとき，その実体はどのように振る舞うのか，すなわち，どのような固有の操作を提供するのかを考える。一応応用における機能を定義したとき，その機能を実現するための操作を提供する実体は何かを考える。

文献6）によれば，オブジェクト指向データベースの設計手順は，次のようになる。

①個々のオブジェクトを識別し，多様なオブジェクトが構成するオブジェクト世界の構造的な性質，すなわちオブジェクト間の構造的な関連性を明らかにする（構造の抽象化）。

②オブジェクトの振る舞い，すなわち操作的な性質を明らかにする。

③他のオブジェクトの振る舞いとの関連，すなわち振る舞いについて，他のオブジェクトに何を

要求し，また他のオブジェクトから何を求められるかを明らかにする（振る舞いの抽象化）。

4.2　オブジェクト構造の抽象化

（1）類型化（Classification）

オブジェクトの構造に関する性質を特性という。類型化とは，共通の特性を持つオブジェクトの集合を考えることをいう。この集合をクラスという。

クラスに属するオブジェクトの特性の集合を型という。クラスに属する各オブジェクトをクラスまたは型のインスタンスという。基本クラスは，integer, real, string, symbol, data などである。

クラス E のオブジェクトの特性 p は

$p: e \varepsilon E \rightarrow e' \varepsilon E'$

であり，E' を E の特性 p の領域という。

（2）集約化（aggregation）

クラス E が，それぞれクラス $E_i (i=1, \cdots, n)$ を領域とする p_i, すなわち

$p_i: e \varepsilon E \rightarrow e_i \varepsilon E_i \ (i=1, \cdots, n)$

を持つとして，オブジェクト e はオブジェクト e_i (i=1, …, n) の集約化によって定義されているという。e を集約オブジェクト，e_i を e の成分オブジェクトという。E を成分クラス E_i を持つ集約クラスという。

①has-constituent: E のオブジェクトの存在が E_i のオブジェクトの存在に依存するとき，E と E_i の関連を has-constituent (E, E_i)，または constituent-of (E, E_i) で表す。

②has-component: E_i のオブジェクトの存在が E のオブジェクトの存在に依存するとき，E と E_i の関連を has-component (E, E_i)，または component-of (E, E_i) で表す。

③汎関連（general relationship): p_i が E_i を領域とする E の特性であるとき，p_i によって対応づけられる E のオブジェクトと E_i のオブジェクトの存在が互いに相手の存在に依存しないとき，この特性 p_i を汎関連と呼ぶ。このとき一般に，E_i には E を領域とする p_i とは逆の特性 q_i が定義れれている。

（3）汎化・特化（generallization, specialization）

いくつかのクラスの共通的な性質に着目し，その共通性によって一般化されたクラスを定義することを汎化という。汎化の逆，すなわちあるクラスに対していくつかの特性を付加することによって，特殊化されたクラスを定義することを特化という。

①subtype-of: クラス E がクラス E_i に特化されているとき，subtype-of (E_i, E)

②role-of: ある環境におけるクラス E のインスタンス e が，別のいくつかの異なる環境において，それぞれある役割を持つクラス E_i のインスタンス e_i としてみられることがある。この場合 e_i は e と同じオブジェクトである。このとき role-of (E_i, E)

4.3　パソコン通信ログの検索システム

図2－4は，パソコン通信ログの検索システムにおける，オブジェクトの構造的な性質を図式的

に表現するオブジェクト図である。長方形はクラスを表し，長方形の左肩に付けられた名前は特性名である。＊印を付けた特性は，対応する領域の複数のオブジェクト，オブジェクトの集合を値としてとる多価特性である。長方形 E_i を長方形 E の内部に描くことによって，E が集約クラス，E_i がその成分クラスであることを表す。図２－４において，「文書検索」は「ログ」と「検索」の集約クラスであり，すなわち，次の関連が表現されている。

has-constituent（文書検索，ログ）

has-constituent（文書検索，検索）

また，「ログ」は「文書」と「索引」の集約クラスであり，「文書」は「文書.文書」と「索引語」の集約クラス，さらに「索引」はいくつかの「索引語」の，「検索」はいくつかの「質問文」の集約クラスである。そして次の関連を仮定している。

has-component（ログ，文書）

has-component（ログ，索引）

has-component（文書，文書.文書）

has-component（索引，索引語）

さらに，「文書」と「索引」の間に，汎関連を表す特性「文書.索引語」と「索引.索引語」が定義されている。互いの索引語は関連しており，互いに逆の関係にある。

クラス「索引」の「索引語」のオブジェクトは，「検索」のオブジェクトの中で特別な役割を持つので

role-of（索引語，検索）

が定義される。すなわち，検索のインスタンスが索引のインスタンスに現れると考えられる。一般には１つのインスタンスは複数のクラスのインスタンスになりうる。また，パソコン通信のログは，一般の文書の特別なものと考えられるので，

subtype-of（ログ，一般文書）

が定義されることになる。

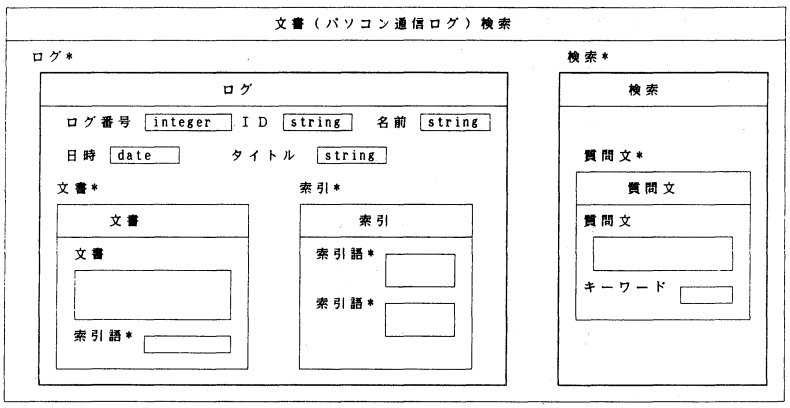

図２－４　文書（パソコン通信ログ）検索オブジェクト

5　オブジェクト指向データベースとリレーショナル・データベースの比較

文献3）などを参考にして，オブジェクト指向データベースとリレーショナル・データベースの比較を行う。

（1）データ型

オブジェクト指向プログラミング言語では，アプリケーションに適したデータ型（抽象データ型）を新たに定義することができる。オブジェクト指向データベースは，これを永続するオブジェクトとしてコンピュータ内に蓄積することができる。

リレーショナル・データベースのデータ型は，文字・数値に制限されており，非常に長い文字列は一般的には扱えない。したがって，文書のデータベースには適さない。

（2）データ構造

オブジェクト指向データベースは，データ構造として階層性を導入し，複雑で動作を含む対象をモデル化するのに適したものになっている。一方リレーショナル・データベースはレコードの集合であるリレーションというデータ構造だけを使い，データベースの簡単化を図っている。パソコン通信のログのように，ある話題のログに対して，そのレスポンスがあり，さらにそのレスポンスというように，ログの階層構造が作られており，オブジェクト指向のデータ構造に適している。

（3）実体の識別子

オブジェクト指向データベースでは，オブジェクト識別子を使ってオブジェクト（実体）を識別する。オブジェクト識別子はデータの値とは独立して存在するものであるから，オブジェクトの存在とオブジェクトの値とを概念のレベルでもはっきりと区別するのに役立つ。

一方，リレーショナル・データベースは利用者が定義したキーによってレコードを識別している。キーによるレコードの識別は，物理的な記憶アドレスからの高い独立性を保証するものであるが，キー管理の煩雑さを利用者に強制するものである。このような実体の識別方法の違いは，適用される応用の範囲を決定しているものと考えられる。

（4）データ操作

オブジェクト指向データベースでは，階層構造を扱っているので，階層構造に沿ったデータ操作を行うことが多い。利用者が直接意識するオブジェクトはキーによって識別され，意識しないオブジェクトはDBMSが管理する識別子によって識別される。階層の深い位置にあるオブジェクトは，オブジェクト識別子によりデータベースの階層構造に沿って順次にデータが操作される。これをナビゲーションによるデータ操作と呼ぶ。

リレーショナル・データベースは，キーを手がかりにデータを操作している。この方法はリレーションという簡潔なデータ構造と調和し，非手続き的な操作言語によってデータ操作を行う。しかし，リレーションにまたがるデータ操作を行うためにはジョイン演算という時間のかかる処理を実行しなければならない。文書などは，リレーションにあてはめるには無理があり，また複数のリレーションにまたがるのが普通になり，リレーショナル・データベースには不向きである。

（5）親言語

リレーショナル・データベースの親言語は，COBOL, FORTRAN, C, PL／1 などである。リレーショナル・データベースの目的の一つは，高度なデータ独立を達成することであり，プログラミング言語とデータ操作言語をはっきり区別することを重視していた。このためアプリケーションを開発するためには，プログラミング言語とデータ操作言語という少なくとも２種類の言語を知らなければならないという結果になり，プログラミング・インタフェース・ギャップを生み出す結果になった。

オブジェクト指向データベースは，プログラミング・インタフェース・ギャップを解消することを目標として，オブジェクト指向データベースにはプログラミング言語とキー操作言語との自然な融合を目指している。したがって，オブジェクト指向プログラミング言語を理解していれば，基本的にオブジェクト指向データベースを使うことができるわけである。オブジェクト指向データベースのほとんどが C＋＋, C, Smalltalk を親言語として採用している。

（6）形式性

リレーショナル・データベースは，E. F. Codd が定義したデータ構造と操作演算を備えた形式性の利点を生かしたデータベースである。

オブジェクト指向はプログラミングの経験から生み出された手法である。そのため統一的な形式に乏しい。オブジェクト指向システムの統一的な形式の欠如は，そのままオブジェクト指向データベースにも持ち越されている。

（7）標準化

標準化は，プログラム・レベルの仕様を統一するためのものである。リレーショナル・データベースの標準化は1980年代後半から行われ，製品の機能面での品質を高めることができ製品の普及に貢献している。オブジェクト指向データベースの標準化は1990年代初めから検討されており，期待されている。

（8）適用分野

リレーショナル・データベースは，リレーションを事務処理に限定しており，設計目標であるデータモデルの簡潔化に成功した。しかし，簡潔なデータモデルは複雑なデータ構造をモデル化する場合の足かせになった。オブジェクト指向データは，複雑なデータ構造をモデル化することを目的としている。適用分野は，CAD／CAM, 文書処理，地図およびシステム・インテグレーションと呼ばれる各種ツールの統合である。オブジェクト指向データの適用分野は，リレーショナル・データベースが不得意とする分野であるということもできる。

特に，オブジェクト指向データベースでは，文書処理をさらに発展させて，ハイパーメディアや知識ベースと組み合わせて，知的データベースへと発展させる研究が進められている。

（9）適用マシン

リレーショナル・データベースが実用化された当初は，処理能力の面から汎用大型コンピュータ

に限られていたが，その後ワークステーションやパーソナルコンピュータにも搭載されるようになってきた。一方オブジェクト指向データベースは，CAD などエンジニアリング業務を対象としており，文字・数値に加えて図形，画像，テキスト（非常に長い文字列）などの扱いが重要になる。そのため，ワークステーションを手始めに実装されてきた。さらに分散処理にも対応することが求められてきた。最近ではパーソナル・コンピュータでも利用できるようになってきている。

6　第2部のまとめと今後の課題

オブジェクト指向データベースが，文書（テキスト・データ）をデータベース化する場合に，どの程度の可能性があるのかについて検討してきた。その際，オブジェクト指向プログラミングと人工知能研究が密接に関連していることがわかってきた。オブジェクト指向プログラミングでは，実世界の対象物，すなわち，オブジェクトがそのデータと振る舞いとともにモデル化される。その結果，実世界で生起するさまざまな事象を容易にシミュレーションしたり，またプログラムの生産性を向上させることが可能となってきた。一方類似の概念は人工知能の分野にもみられ，例えば，フレーム理論はその典型である。さらに，この2つの概念はオブジェクト指向データベースとも深く関わっている（文献7）。

今後の研究の方向としては，オブジェクト指向プログラム（例えば，C＋＋）による，オブジェクト指向データベースの実験的試作や人工知能関連の研究を進めたい。

本研究のまとめ

本論文の第1部は山下，第2部は真田が主として担当した。本報告は著者等がデータベース研究に関わった研究の第一歩の報告であり，まだまだ未熟な面が多いが，今後さらに本格的に研究を深めていきたいと考えている。

謝辞：sed, jgawk の使用方法等に関して有益な助言を頂いた川内職業能力開発短期大学校の新山亘教官に感謝の意を表する。

参　考　文　献

1）松岡　他（1992）：テキストデータ料理学，翔泳社
2）桐 Ver 4.0 リファレンス1，2　管理工学研究所
3）宇田川佳久（1993）：オブジェクト指向データベース入門，ソフトリサーチセンター
4）K. Parsaye, M. Chignell, S. Khoshafian & H. Wong（1992）：知識データベース，オーム社
5）田中克己（1991）：オブジェクト指向データベースの基礎概念，情報処理，Vol. 32, No. 5
6）酒井博敬（1991）：オブジェクト指向データベースの設計，情報処理，Vol. 32, No. 5
7）増永良文（1991）：次世代データベースシステムとしてのオブジェクト指向データベースシステム，情報処理，Vol. 32, No. 5