

TR-I-0009

簡易検索言語をもつ音声データベース管理システム
**Speech Database Management System and Easy
Access Language**

武田 一哉

Kazuya Takeda

1987.12

内容梗概

各種研究目的に音声データを共用するために構築された、音声データベース管理システム(音声DBMS)について述べる。音声DBMSでは言語-音声単位毎に関係表現された属性が階層的に構成され、複雑な言語-音声環境を柔軟に表現することができる。また、システム上の簡易検索言語EALを用いて、データの検索・抽出を効率的に行うことができる。巻末には、簡易検索言語EALの使用手引を掲載した。

1. はじめに

音声処理技術の研究開発における、音声データベースの重要性は古くから認識されており、音声データベース構築の試みがこれまで多くの研究機関でなされてきた[1][2]。しかしながら、構築されたデータベースが複数の研究目的に効率的に共同利用されている例は稀である。これは、各研究目的を満たす大規模音声データが収集・蓄積されていても、その中から研究目的に合ったサブセットあるいは特定音声区間を検索・抽出することが、困難であるためと考えられる。大規模な汎用音声データベースを、効率的に種々の研究目的に利用して行くためには、蓄積された音声データの検索・提供をおこなう音声データベース管理システム(音声DBMS)が不可欠である。

我々は、各種の音韻環境下における発声を含み、かつ階層的に音韻環境が記述された、研究用日本語音声データベースの構築・整備を進めており、すでに6万語程度の音声データの収集を完了している[3]。本稿では、収集した音声データを効率的に、各種研究目的に共用するため構築を行った、音声DBMSについて述べる。さらに巻末にはAppendixとして、DBMS上の簡易検索言語EALの使用マニュアルを掲載する。

2. 研究用音声データベース

2.1 格納語彙

収集したデータの発声内容を表1に示す。8人(男女各4名)の発声を各語について2回ずつ収録し、そのうちの1回分について視察によるラベリングを行っている。重要語は三省堂新明解国語辞典から抽出し、文章データはATRで行った電話模擬実験の会話をもとに作成した。現在までにこのうち3名の発声を音声DBMS上に格納しており、他の話者による発声については、ラベリングと並行して音声DBMS上への格納作業を進めている。

項目	語数
重要語	5240語*
会話文	115文 各々につき3種類の 区切り方で発声
英数字 音韻バランス単語等	386語

表1 発声内容

2.2 音声データの階層的な記述法

音声データベースを構築するためには、収集した音声データにラベルを付与して、発声と音声信号区間との対応づけを行うことが必要である。実際の発声においては、先行・後続音韻等の局所的な音韻環境の外に、アクセント核、形態素境界の位置等、さまざまな環境が音韻の物理的特徴に影響を及ぼしていると考えられる。これらの環境を表現するためには、1次元的なラベルの列記のみならず、各種の言語・音声的単位ごとに、音声信号のラベルづけ及び属性の記述を、行ってゆく必要がある。そこで我々は、音声データベースの構築にあたり5階層からなる階層的音韻ラベルを設計し¹⁴⁾、ラベリングを行っている。本ラベルを用いることにより、音声事象(破裂音内の閉鎖,気音区間等)といった音響的にコンパクトな単位で¹⁵⁾音声波形の記述を行うことが出来ると同時に、カナ表記を単位とする形態素境界,アクセント核の位置等の環境も容易に表現することが出来る。音声DBMS上のデータ表現は、これらの階層的ラベルに基づいて行った。

3. 音声DBMSの構築

3.1 音声DBMSの概要

前節で述べたように、音声データはいくつかの言語音声的な単位の持つ属性の組合せとして表現することができる。音声DBMSではこのようなデータ構造を、関係形式のデータ表現で実現している。図1は音声DBMSのシステム概要を示している。システムはUNIX上で稼働する関係データベースUNIFY上に構築されており、データ構造の定義はスキーマファイルに、実際の属性値はデータファイルに各々格納されている。さらに、データアクセスのためのインターフェイスとして簡易検索言語EAL (Easy Access Language) が設計されている。実際の音声データはDBMSとは独立して存在しており、検索の結果得られるデータポインタを通じてアクセスすることになる。

3.2 DBMSにおけるデータ表現

一般に発声は、文,分節,単語,音節等の(言語的)単位の上で行われており、それぞれの単位ごとに様々な変動が引き起こされていると考えられる。一方音声信号を観察すると、調音上の拘束からそれらがいくつかの物理的に安定な区間(音声事象)の組合せとして表現できることも分かる。研究用音声データベースでは、このような音声事象レベルで音声データへのアクセスが可能であるだけでなく、それを取り巻く言語環境に関する情報も格納されている必要がある。本音声DBMS構築にあたっては、これら言語音声単位各々の属性表現を組み合わせること

*同一単語の重複が6組12単語あるため、異なりの語彙数は5228語。

により、音声データを表現している。

図2にDBMSにおけるデータ表現の概要を示す。文章,単語,CV音節,音韻,音声事象及び、話者,収録/AD環境の各属性項目が各々関係形式に基づいて表現されている。音韻,音声事象レコードの各属性項目は、前節で述べた音韻ラベルを整理して作成されている。各言語音声単位の属性項目はAppendix中に示す通りである。例えば音韻属性項目には、無声化,融合化等の属性項目が設けられており、音節,単語の属性項目と組み合わせて、様々な環境の表現が可能となる。また先行・後続する音韻レコードの物理アドレスを利用して、先行・後続する音韻の各属性項目にも、容易にアクセスすることができる。さらに、単語の品詞,形態素構成といった情報は国語辞書から抽出し、音韻,音声事象属性とは独立に単語属性の項目として単語単位の格納されている。これらの言語的な情報は、音韻ラベルより生成される仮名表記(CV音節属性)を介して、音韻,音声事象と対応付けられる。このように、音

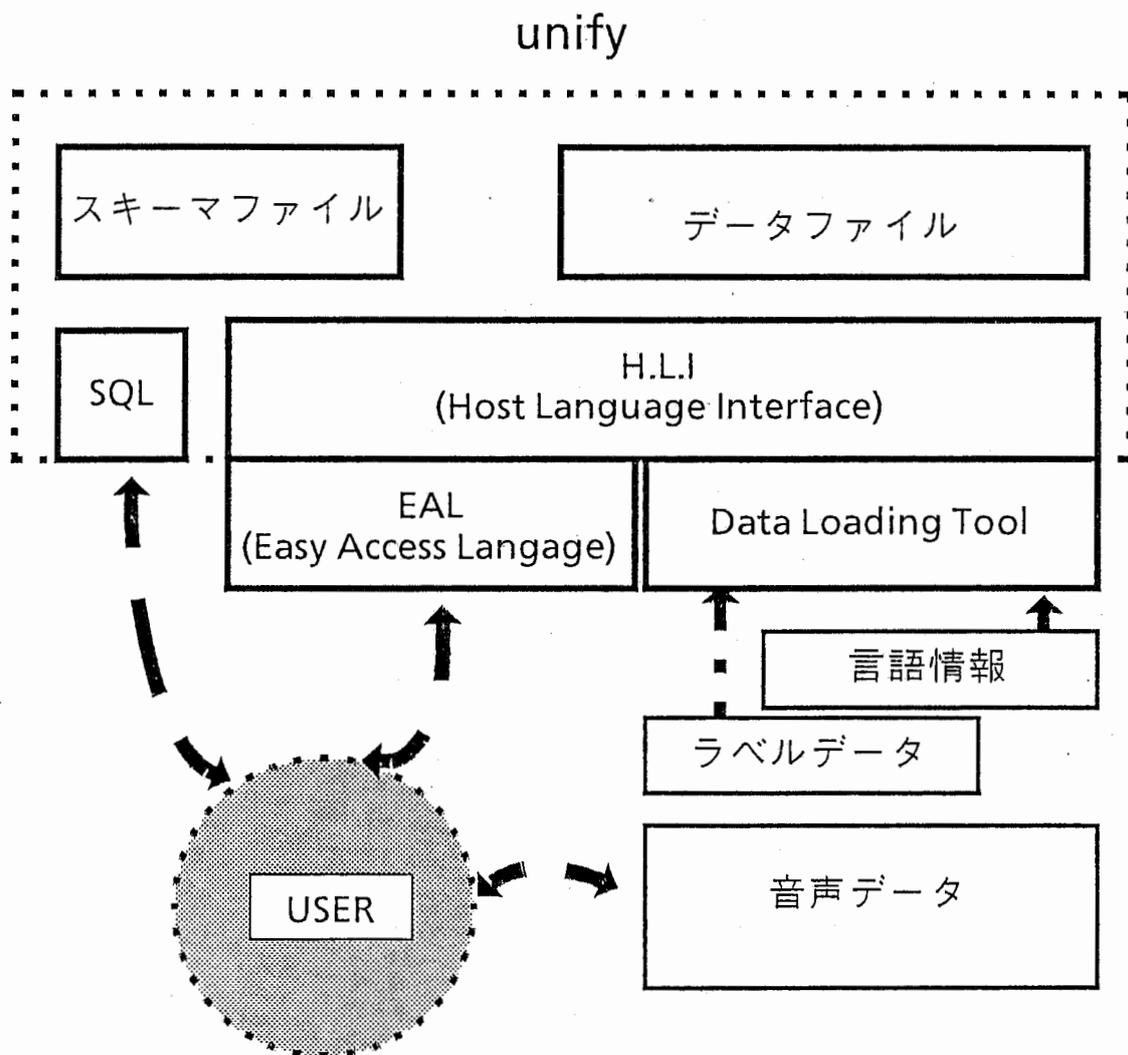


図1. 音声データベース管理システム概念図

声DBMSでは、言語音声単位の属性表現を階層的に構成することにより、各種の音韻環境、言語環境を表現することが可能となっている。

3.3. 検索用インターフェイス

音声DBMS構築の最大の目的は音声データの効率的な共同利用であり、これを実現するためには、データの検索提供機能が重要となる。構築したDBMSにおいてデータの検索提供を行うユーザーインターフェイスには、unifyのH.L.I.(Host Language Interface)を用いて作成した簡易検索言語EALが用いられる。図3は、EALを用いて『破裂音に先行する、語頭音節中の母音の、継続時間及び、データポイント』を検索する例である。検索条件の設定は、図のように論理記号と属性

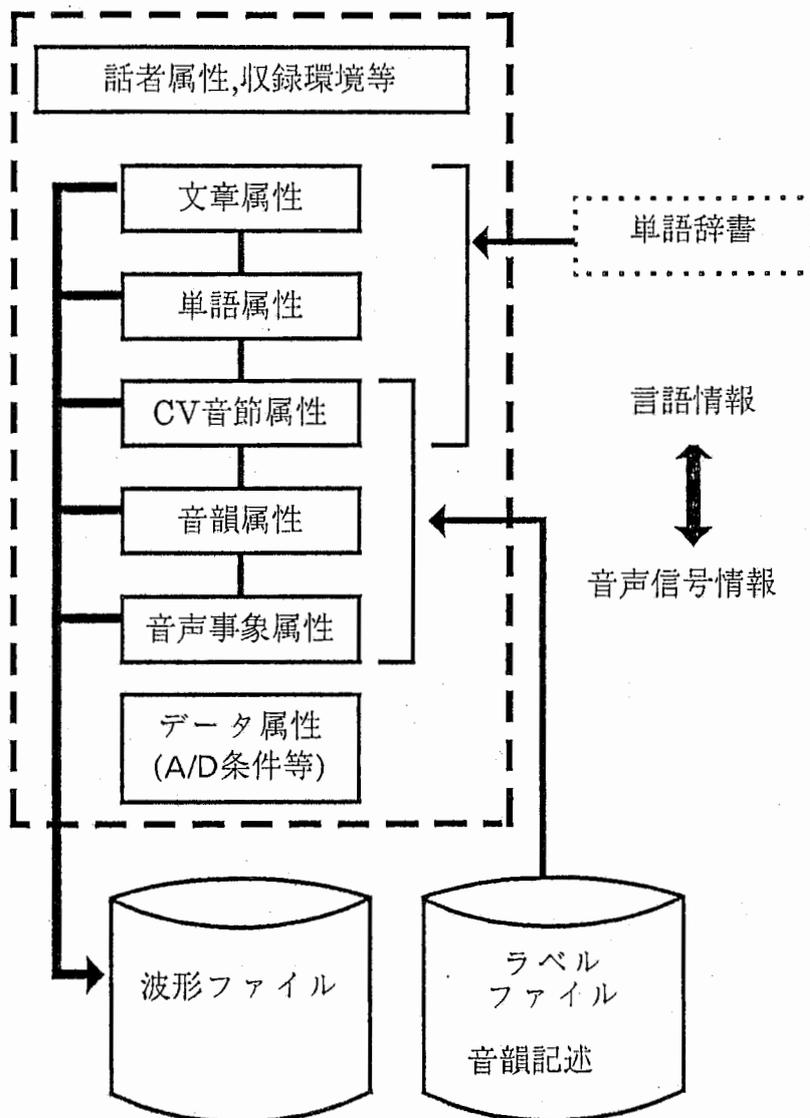


図2 音声DBMSにおけるデータ表現

項目名を用いて行われる。さらに、VOWEL(母音),PLOSIVE(破裂音)等のキーワードも用意されている。また接頭子“pre.”(“fol.”)を複数個用いることにより、2つ以上前(後ろ)の音韻の属性項目も条件式中に取り入れることが可能になっている。(例えば2つ前の音韻の継続時間は pre.pre.phn_dur と表現できる。)このように、簡易検索言語を用いることで、データ構造を意識することなく効率的に希望の音韻環境にアクセスできる。(EALの使用法についてはAppendixで詳細に説明する。)

5. まとめ

音声データベースにおけるデータベース管理システムについて述べた。管理システムでは言語音声的単位毎に関係表現された属性が階層的に構成されている。さらに、システム上の簡易検索言語を用いて、種々の音韻環境におかれた音声データの検索を効率的に行うことが出来る。今後も格納音声データの規模拡大に努めると同時に、音声DBMSの機能向上を計って行く。

```
select    phn_dur,
          phn_dp;
where     (syl_loc == 1) &&
          (fol.phn_name == PLOSIVE) &&
          (phn_name == VOWEL);
```

図3 簡易検索言語による検索例

謝辞

活発な討論を頂いた樽松社長、鹿野室長を初めとするATR自動翻訳電話研究所研究員の皆様に、感謝いたします。また、データロード、及びプログラミング作業に御協力いただいたTIS(東洋情報システム)の森川氏に感謝いたします。

文献

- [1]溝口,前田他,"知的アクセス機能を持つ音声データベース「SPEECH-DB」",情報処理学会誌,24(3),pp.271-279,(1983).
- [2]板橋,"単語音声データベース",音響学会誌,41,1985,pp.723-726
- [3]武田,匂坂,片桐,桑原,"音韻ラベルを持つ日本語音声データベースの構築"音声研資,sp87-19,(1987).
- [4]武田,匂坂,片桐"音声データベース構築のための音韻ラベリング",音講論集,2-5-10(1987.3)
- [5]田中,速水,太田"音声の音素片ネットワーク表現と時系列のセグメント化法を用いた自動ラベリング手法",音響学会誌,42,1986,pp.860-868

APPENDIX

簡易検索言語EAL使用マニュアル

§ 簡易検索言語 E A L 使用マニュアル

はじめに

1. システムの目的と概要
 1. 1. 目的
 1. 2. 概要
2. 使用にあたっての基礎知識
 2. 1. D B 論理構造
3. 簡易検索言語「E A L」使用の手引き
4. 簡易検索言語「E A L」解説
 4. 1. コントロールファイルの記述
 4. 1. 1. 形式
 4. 1. 2. 項目名詳細
 4. 1. 3. コードによる指定が必要な項目
 4. 1. 4. s y n t a x 詳細
 4. 1. 5. 音韻レベルで使用出来る特殊なラベル
 4. 2. 注意事項
5. メッセージ一覧

付録

《簡易検索言語『E A L』モジュール構造》

はじめに

1. 対象者

本書は以下の人を対象にしている。

- 音声DBMS利用者

2. 本書の目的

本書により、音声DBMSに関する概要、簡易検索言語EALの使用法が理解出来る。

3. 関連マニュアル

- UNIX関連マニュアル
- 「UNIFY TUTORIAL MANUAL」
- 「UNIFY REFERENCE MANUAL」
- 「UNIFY PROGRAMMERS' MANUAL」

4. プログラム

今回作成したプログラムは、

ソースファイル：/data2/SPEECHDB_NEW/src

モジュール : /data2/SPEECHDB_NEW/tool

にある。

1. 音声DBMS、目的と概要

1. 1. 目的

音声認識、音声合成の研究用に集められた音声データに関して、ラベラーによって付加されたラベリング・データをもとに、様々な音韻情報をData Baseに持たせることにより、希望するデータの情報検索に便宜をはかることを目的としている。

1. 2. 概要

音韻情報をもとに作成したデータは、リレーショナルDBMSであるUNIFYに管理され、このUNIFYのDBはインデックスDBとして機能している。現段階では、音声DBMSから取り出すことが可能なのは、インデックスDB内の情報に限られているが、将来的には、この情報をもとに音声データ全体、または、その一部をファイルに取り込むことも出来るようにする予定である。

音声DBのアクセス・ツールとして、簡易言語型検索プログラム

- EAL (Easy Access Language)

を、その他にインデックスDBにデータをロードするために、

- ロード用ワークファイル作成ルーチン
- ロードプログラム

を用意している。

これらのプログラム等により音声DBMSを構成している。

システム概要図を次ページの図1に示す。

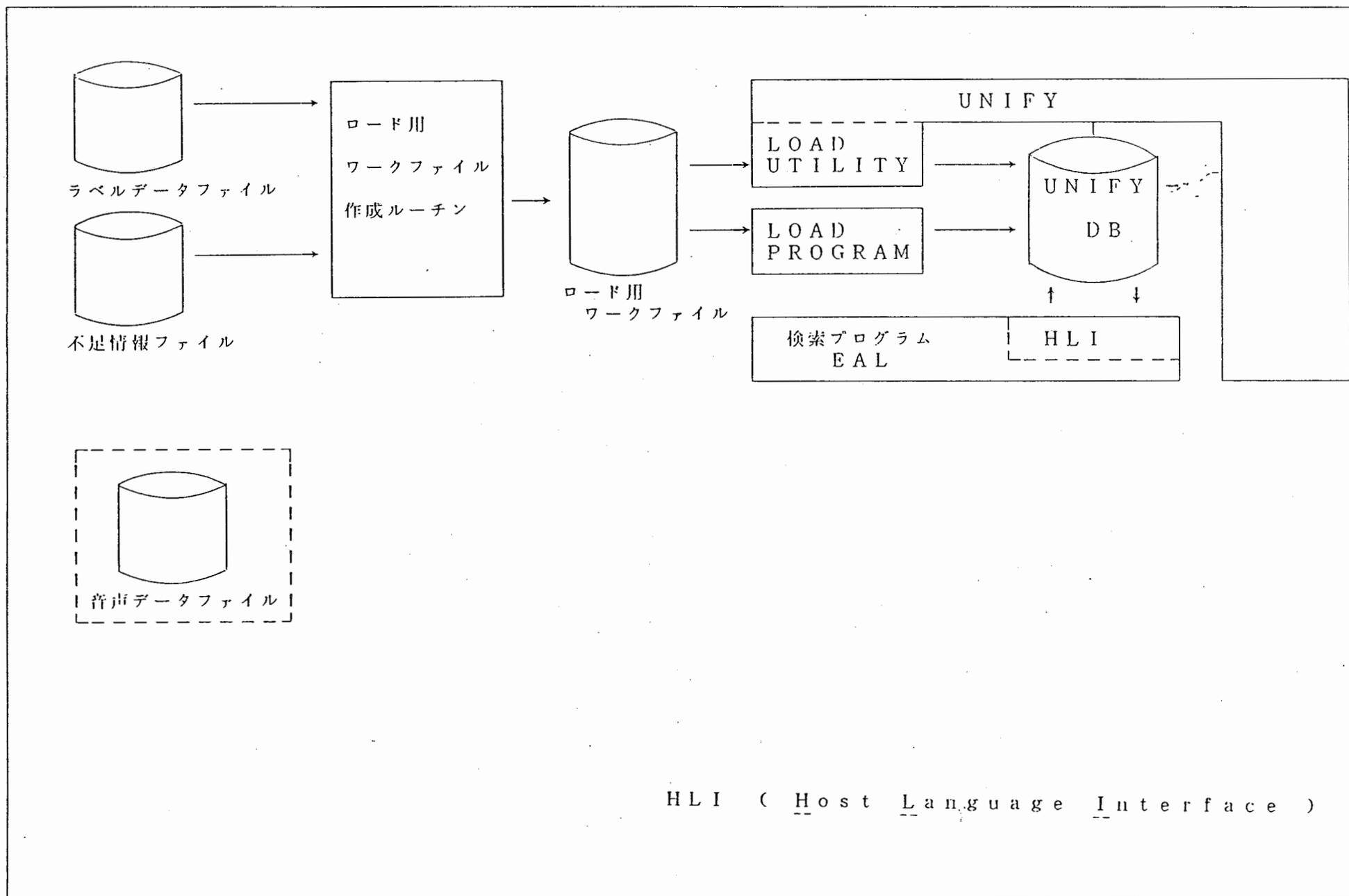


図1. 音声DBMS
システム概要図

2. 使用にあたっての基礎知識

検索システムを利用するための前提条件として、以下の知識を想定している。

- 一般的なUNIXコマンドの使用法と動作
- エディタを用いてファイルを作成出来ること
- 音声DBMSの概要（「1. 音声DBMS、目的と概要」）
- DB論理構造（「2. 1. DB論理構造」）

2. 1. DB論理構造

音声DBには、利用者から見た場合、以下の情報が入っている。

- A/D環境情報
- 発声環境情報
- 話者情報
- 文章属性
- 単語属性
- 音節属性
- 音韻属性
- イベント属性

それぞれの情報について、詳細を次に述べる。

○ A/D 環境情報

- 符号長
- サンプリングレート
- カットオフ周波数
- 減衰率

○ 発声環境情報

- 収録日
- 収録場所
- 使用機材 (マイク)
- タスクコード

○ 話者情報

- 話者名
- 性別
- キャリア
- 出身地
- 方言
- 身長
- 年齢

○ 文章属性

- 文章番号
- 発声内容
- 包含単語数
- データファイル名
- ボリューム名
- データ開始時間
- データ終了時間
- データポインタ

○ 単語属性

- ローマ字表記
- 仮名表記
- 漢字表記
- 品詞
- CV 音節数
- モーラ数
- 形態素数
- 形態素境界
- アクセント型
- 文章内位置
- データ開始時間
- データ終了時間
- データポインタ

○ 音節属性

- 単語内位置
- 音節名
- データ開始時間
- データ終了時間
- データ継続時間
- アクセントフラグ
- 融合ラベル
- 融合開始時間
- 融合終了時間
- 無声化フラグ
- 無声化開始時間
- 無声化終了時間
- 促音フラグ
- 長音フラグ
- データポインタ

○ 音韻属性

- 音節内位置
- 音韻名
- データ開始時間
- データ終了時間
- データ継続時間
- 無声化フラグ
- 融合化フラグ
- 異音化ラベル
- 異音化開始時間
- 異音化終了時間
- 長音フラグ
- 包含イベント
- データポインタ

○ イベント属性

- 音韻内位置
- イベント名
- データ開始時間
- データ終了時間
- データ継続時間
- データポインタ

3. 簡易検索言語「EAL」使用の手引き

- セットアップ

1. moduleが、" /data2/bin" にあるので、
pathに" /data2/bin" を追加する。
2. 環境変数 DBPATH を、
" /data2/SPEECHDB_NEW/bin" に
設定する。

(同一構造をもつ、テスト用の小型のDBが、
" /usr5/morikawa/TESTDB/bin" がある。
使い方に慣れるために使用するのであれば、
環境変数DBPATHを、" /usr5/morikawa/TESTDB/bin"
に設定しても構わない。)

コマンドは、以下の通り。

```
' setenv DBPATH /data2/SPEECHDB_NEW/bin '
```

この環境変数が正しく設定されていない場合、EALは
ただちに終了するので、必ず設定すること。

- 使用法

出力項目、選択条件を記述した、コントロールファイルを
作成し、EALに入力することにより、選択条件を満たす
データの中から、指定した項目を、指定した順序通りに
"標準出力" に出力する。

出力のフォーマットは、次の通り。

```
項目値 ":" 項目値 ":" ..... ":" 項目値 Yn  
":" ..... フィールドセパレータ
```

このプログラムは、バックグラウンドで走らせることが
出来る。

実行例を、次ページに示す。

○ 実行例

```
csh>> EAL cntl_file > out_file &
```

```
csh>> cat cntl_file
```

```
/* 有声の破裂音で包含イベントに破裂部のみで、  
   気音部をもたない音韻に関して、その音韻名、  
   継続長、包含イベント、単語漢字表記を得る。 */
```

```
select      phn_name    ,  
            phn_dur     ,  
            eve_name    ,  
            word_kanji ;  
where       (   
              ( phn_name == plosive )  
              &&  
              ( phn_name == voiceless )  
            )  
            &&  
            ( phn_evelbl == ( [ptk] && !cl ) )  
            &&  
            ( eve_name == [ptk] ) ;
```

```
csh>> cat out_file
```

```
k:120.00:k:勢い  
k:82.50:k:垂直  
k:67.50:k:台所  
t:100.00:t:ちゃん と  
t:10.00:t:取り扱う  
k:52.50:k:取り扱う  
p:15.00:p:ポケット  
k:50.00:k:ポケット  
t:235.00:t:ポケット  
k:72.50:k:掛け合う  
k:32.50:k:カレンダー  
t:75.00:t:終点  
.  
.  
.  
.  
t:60.00:t:隠れピューリタン
```

コントロールファイルの記述に関する詳細は、次の
” 簡易検索言語「EAL」解説 ” で述べる。

4. 簡易検索言語「EAL」解説

4. 1. コントロールファイルの記述

4. 1. 1 形式

コントロールファイルは次のような形式をしている。

```
select      項目名 ,
            .
            .
            .
            項目名 ;

where      ( 条件式 ) [ && , || ]
          ( 条件式 ) [ && , || ]
          .
          .
          ( 条件式 ) ;
```

項目名 : EALで取り出したり、条件付けをするための項目名称。
詳しくは、“4. 1. 2 項目名詳細”で述べる。

条件式 : 検索条件の記述。 次の様な形式をしている。

```
( 項目名 演算子 項目名 )
      または
( 項目名 演算子 値 )
```

演算子には、'=='、'!='、'>'、'>='、'<'、'<=' がある。

音節、音韻、イベント について、前後関係を記述したい場合には、
項目名の前に、“pre.”、“fol.”を付ける。

コントロールファイルの記述に関しての、注意事項等については、
“4. 1. 4 syntax詳細”を参照のこと。

また、コントロールファイル内の任意の所にコメントを記述出来る。
コメントと見なされるのは、'/*'と'*/'の間の記述である。
ただし、コメントの中にコメントを書くことは出来ない。

4. 1. 2 項目名詳細

コントロールファイルの中で使用出来る項目名、データタイプ、および、内容を以下に示す。

○ A/D環境

ad_code	Numeric	符号長	(bit)
ad_smp	Numeric	サンプリングレート	(KHz)
ad_lpf1	Numeric	カットオフ周波数	(KHz)
ad_lpf2	Numeric	減衰率	(-db)

○ 発声環境

rec_date	String	収録日
rec_place	Code	収録場所
rec_mic	Code	使用機材(マイク)
taskcode	Code	タスクコード

○ 話者

spkr_name	Code	話者名
spkr_sex	Code	性別
spkr_car	Code	キャリア
spkr_birth	Code	出身地
spkr_dialect	Code	方言
spkr_stature	Numeric	身長
spkr_age	Numeric	年齢

○ 文章属性

sen_no	Numeric	文章番号
sen_text	String	発声内容
sen_words	Numeric	包含単語数
filename	String	データファイル名
volume	String	ボリューム名
sen_stt	Numeric	開始時間
sen_end	Numeric	終了時間
sen_dp	*	データポインタ

○ 単語属性

word_roman	String	ローマ字表記
word_kana	String	仮名表記
word_kanji	String	漢字表記
word_hinshi	Code	品詞
word_cv	Numeric	C V 音節数
word_mora	Numeric	モーラ数
word_mor_num	Numeric	形態素数
word_mor_bnd	String	形態素境界
word_accnt	Numeric	アクセント型
word_pos	Numeric	文章内位置
word_stt	Numeric	開始時間
word_end	Numeric	終了時間
word_dp	*	データポインタ

○ 音節属性

syl_pos	Numeric	単語内位置
syl_name	String	音節名
syl_stt	Numeric	開始時間
syl_end	Numeric	終了時間
syl_dur	Numeric	継続時間
syl_accnt	Flag	アクセントフラグ
syl_ins	String	融合ラベル (注1)
syl_ins_stt	Numeric	融合開始時間
syl_ins_end	Numeric	融合終了時間
syl_dev	Flag	無声化フラグ
syl_dev_stt	Numeric	無声化開始時間
syl_dev_end	Numeric	無声化終了時間
syl_sokuon	Flag	促音フラグ (注2)
syl_len	Flag	長音フラグ (注3)
syl_dp	*	データポインタ

○ 音韻属性

phn_pos	Numeric	音節内位置
phn_name	String	音韻名
phn_stt	Numeric	開始時間
phn_end	Numeric	終了時間
phn_dur	Numeric	継続時間
phn_dev	Flag	無声化フラグ
phn_ins	Flag	融合化フラグ
phn_alp	String	異音化ラベル
phn_alp_stt	Numeric	異音化開始時間
phn_alp_end	Numeric	異音化終了時間
phn_len	Flag	長音フラグ (注3)
phn_evelbl	String	包含イベント (注4)
phn_dp	*	データポインタ

○ イベント属性

eve_pos	Numeric	音韻内位置
eve_name	String	イベント名
eve_stt	Numeric	開始時間
eve_end	Numeric	終了時間
eve_dur	Numeric	継続時間
eve_dp	*	データポインタ

注1 分離不可能なラベル。例えば、無声化している”k i”という音節には、「k, i」と記してある。

注2 例. 「きっかけ(k i k k a k e)」では、音節”k i”にフラグが立っている。

注3 例. 「ページ(p e e j i)」では、音節”p e”, ”e”及び、音韻”p”, ”e”, ”e”にフラグが立っている。

注4 音韻が含んでいるイベントラベル

その他

☆ データポイントは、出力専用の項目。

☆ F l a g の項目で条件指定をするときは、y / nで指定する。

データタイプが、C o d eとなっているものについて、その詳細を次項で述べる。

4. 2. 3 コードによる指定が必要な項目

• 収録場所

コード	収録場所
1	スタジオ
2	簡易収録設備
3	屋内
9	その他

• 使用機材

コード	使用機材
1	サンケンMU_2C
9	その他

• タスクコード

コード	タスクコード
IM	重要語
B	バランスリスト
NA	数字A
NB	数字B
SY	単音節
F	外来語音節
A	アルファベット
SA1-4	文節A
SB1-4	文節B
SC1-4	区切りなし

• 性別

コード	性別
1	男性
2	女性

- キャリア

コード	キャリア
1	アナウンサー等
9	その他

- 出身地

コード	出身地
1	九州
2	関西
3	中部
4	関東
5	東北
9	その他

- 方言

コード	方言
1	関西方言
2	その他方言
3	方言なし
9	その他

- 品詞

コード	品詞
1	名詞
2	動詞
3	助詞
4	形容詞
5	副詞
6	形容動詞
7	助動詞
8	間投詞
9	接続詞

4. 1. 4 syntax 詳細

○ キーワード

`select` : 出力項目の記述を始める。
`where` : 選択条件の記述を始める。

○ 出力項目の指定

出力項目の記述は、`select`の後から始め、項目名はセパレータ `,` で区切られ、最後は `;` で終わらなければならない。

○ 選択条件の指定

選択条件の記述は、`where`の後から始め、最後は `;` で終わらなければならない。

また、(項目名 演算子 項目名) または、(項目名 演算子 値) という形式のものを、論理記号でつないだ形式でなくてはならない。

○ 記号

選択条件で使用出来る記号は以下の通り。

- 演算子 $==$, $!=$, $>$, $<$,
 $>=$, $<=$

但し、Numeric以外の属性をもつ項目では、`"=="` , `"!="`のみ使用可。

- 論理記号 `&&` , `||` , `!`
- その他 `(,)`

優先順位

`()` $>$ `!` $>$ `&&` $>$ `||`

例.

条件の記述に関して、音韻が `'p'` もしくは `'t'` という場合、

```
( phn_name == p || t )
```

```
( phn_name == p ) || ( phn_name == t )
```

どちらの記述をしても構わない。

但し、

`(phn_name != p || t)` の場合、右辺の結合が先に起こるので、

`(phn_name != p) && (phn_name != t)` と等価になる。

○ 前後関係の記述法

音節、音韻、イベントレベルの項目に関しては、項目名の前に、“pre.”，“fol.”を付けることにより、前後関係の記述が出来る。

例. 先行音素が“a”であるような音素“k”

```
( phn_name == k ) && ( pre.phn_name == a )
```

“pre.”，“fol.”は重ねて使用可能、例えば、“fol.fol.phn_name”は、2つ後の音韻名を表す。

○ レギュラエクスプレッションの使用法

ストリング属性を持つ項目の値として、“*”，“?”，“[••]”といったメタキャラを使用出来る。これらの使用法は、UNIXのものと同様である。

例. ローマ字表記で、“ki”を含むものの指定。

```
( word_roman == *ki* )
```

例. 音韻“py”を含む音節名の指定。

```
( syl_name == py? )
```

例. 無声の破裂音である音韻名の指定。

```
( phn_name == [ptk] )
```

4. 1. 5 グループ記号

音韻名による条件の記述用に、以下のグループ記号が用意されている。これを用いることにより、個々の音韻名を、書き並べる必要がない。

例. (p h n _ n a m e == h || c h)
 ||
 (p h n _ n a m e == g l o t t a l)

グループ記号の詳細は以下の通り。

記号	音韻名
l i p	b, w, m, f, p
d e n t a l	z, j, s, t s, d, t, r, n
p a l a t a l	s h, g, k, y
g l o t t a l	h, c h
v o i c e d	b, w, m, z, j, d, r, n, g, y
v o i c e l e s s	f, p, s, t s, t, s h, k, h, c h
f r i c a t i v e	f, z, j, s, s h, h
a f f r i c a t e	t s, c h
p l o s i v e	b, p, d, t, g, k
s e m i - v o w e l	w, y
n a s a l	m, n
l i q u i d	r
v o w e l	a, i, u, e, o

4. 2. その他注意事項

- イベントラベルで "※c l" を指定したいときは、

```
e v e _ n a m e == \※c l
```

と、記述する。

- 話者名 (s p k r _ n a m e) の指定は、
MAU, MHT, MNM, . . . で行なう。

```
s p k r _ n a m e == MAU || MHT
```

- 音節、音素、イベントに関して、前後のラベルがないものについて条件を設定したい場合、"NULL" を使って、

```
p r e . p h n _ n a m e == NULL
```

という指定をする。

5. メッセージ一覧

簡易検索言語「EAL」を使用した際にエラーメッセージが書き出されることがある。メッセージには、

- EALが出すもの
- UNIFYが出すもの

の、2種類があるが、ここでは、EALが出すものの中で、利用者が対処出来るものについて、挙げる。

- 《メッセージ》' Control file xxx open error'
xxx:コントロールファイル名
《説明》 コントロールファイルをオープン出来なかった。
《対応》 コントロールファイル名を正しく与える。
- 《メッセージ》' Keyword xxx error'
xxx:キーワード名(select, where)
《説明》 コントロールファイル中の記述にキーワードxxxがない。
《対応》 コントロールファイルを正しく記述する。
- 《メッセージ》' Invarid select clause'
《説明》 コントロールファイル中の、selectの項の記述に誤りがある。
 - selectに続く項目がない
 - selectに続く項目名が正しくない《対応》 コントロールファイルを正しく記述する。
- 《メッセージ》' Invarid where clause'
《説明》 コントロールファイル中の、whereの項の記述に誤りがある。
 - selectに続く項目がない
 - selectに続く項目名が正しくない《対応》 コントロールファイルを正しく記述する。
- 《メッセージ》' Where clause format error'
《説明》 コントロールファイル中の、whereの項の記述がsyntaxに合っていない。
《対応》 syntaxを参照し、コントロールファイルを正しく記述する。

- 《メッセージ》 'DBPATH error'

《説明》 環境変数 'DBPATH' が正しく設定されていない。

《対応》 環境変数 'DBPATH' を正しく設定する。
コマンドは次の通り。

```
setenv DBPATH yyy
```

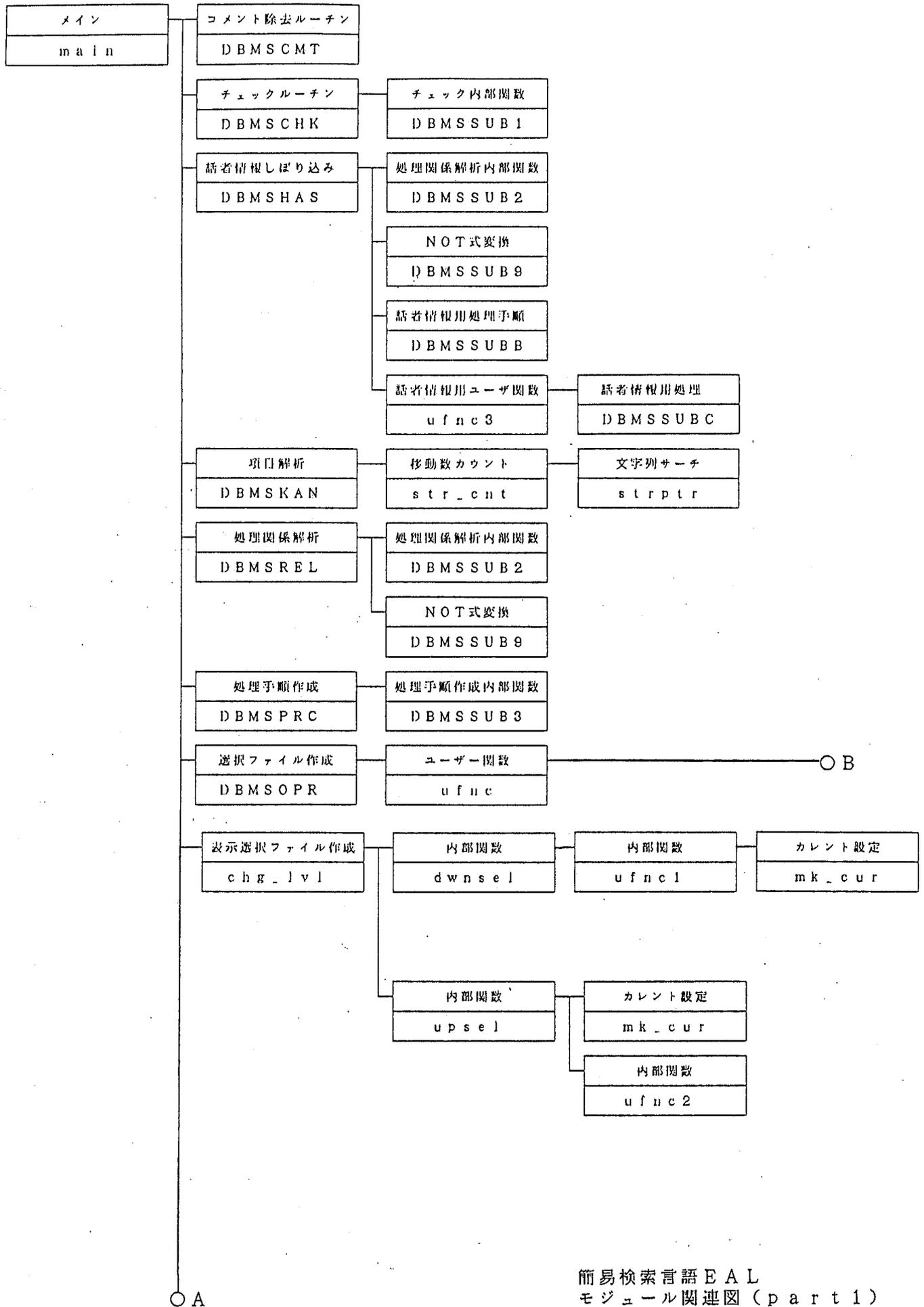
```
yyy: /data2/SPEECHDB_NEW/bin
```

- 《メッセージ》 'Tmp file XXX wrlte failed.
Permission denied.'

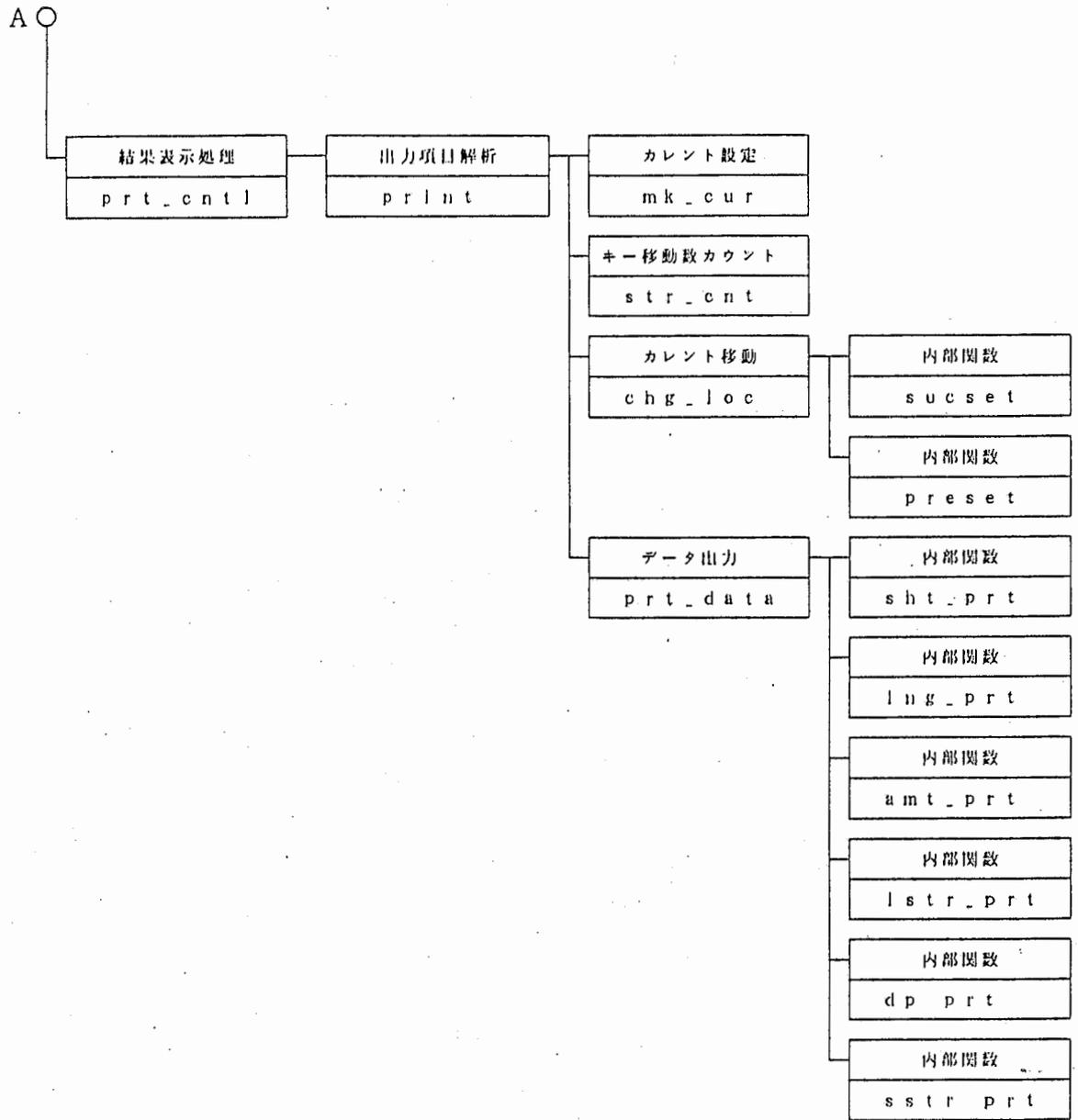
《説明》 一時ファイルを書き出そうとしたが、書込みが出来なかった。

《対応》 書込み可能なディレクトリで実行する。

《 簡易検索言語『EAL』 モジュール構造 》

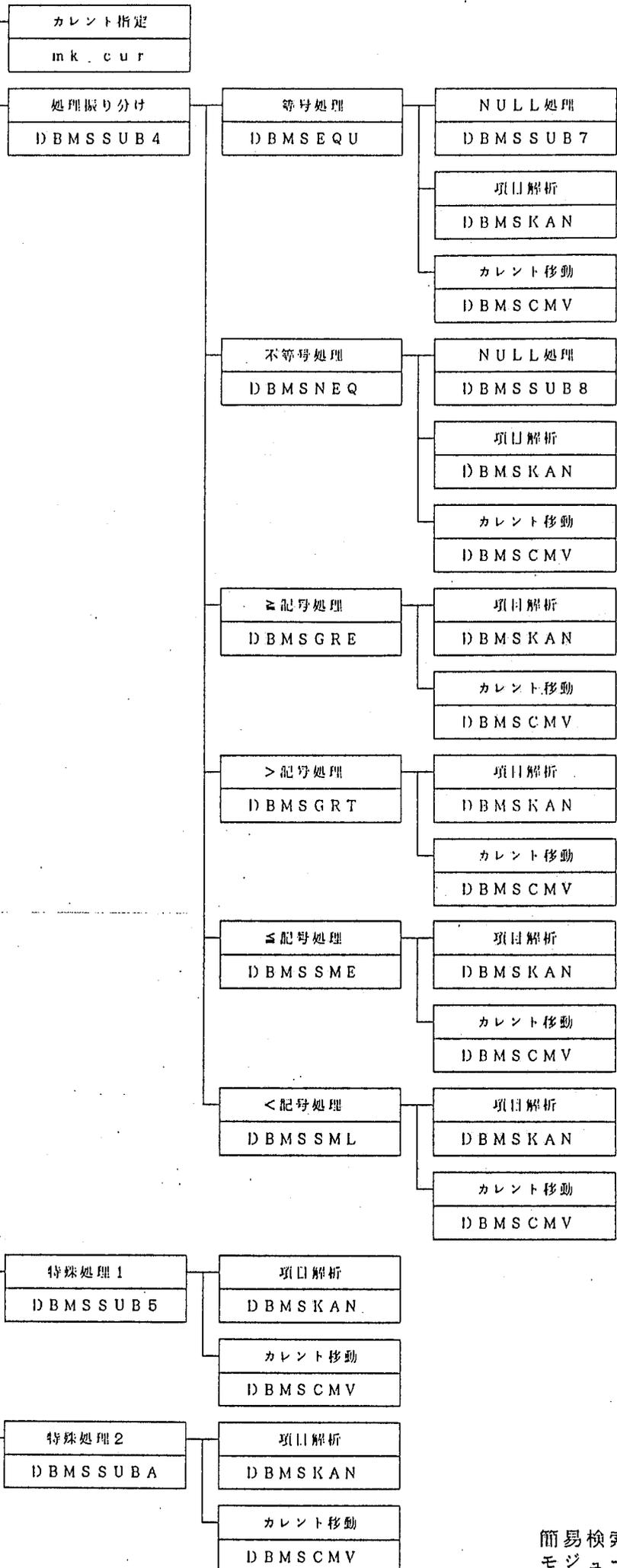


簡易検索言語 EAL
モジュール関連図 (part 1)



簡易検索言語 EAL
モジュール関連図 (part 2)

BO



◇ 簡易検索言語 E A L モジュール一覧

モジュール名	ソースファイル名	機能概要
main	DBMSnln7.c	コントロール
DBMSCMT	DBMSCMT.c	コメント除去
pvt_cntl	DBMSnln7.c	結果出力コントロール
DBMSCHK	DBMSCHK.c	入力パラメータチェック
DBMSSUB1	DBMSCHK.c	チェック用内部関数
DBMSREL	DBMSREL.c	処理関係解析
DBMSSUB2	DBMSREL.c	処理関係解析内部関数
DBMSSUB9	DBMSREL.c	N O T 式変換
DBMSPRC	DBMSPRC.c	処理手順作成
DBMSSUB3	DBMSPRC.c	処理手順作成内部関数
DBMSOPR	DBMSOPR.c	選択ファイル作成
ufnc	DBMSOPR.c	ユーザ関数
DBMSSUB4	DBMSOPR.c	処理振り分け
DBMSSUB5	DBMSOPR.c	特殊処理
DBMSSUBA	DBMSOPR.c	特殊処理
DBMSHAS	DBMSHAS.c	話者情報しぼり込み
DBMSSUBB	DBMSHAS.c	話者情報処理手順作成
ufnc3	DBMSHAS.c	話者情報用ユーザ関数
DBMSSUBC	DBMSHAS.c	話者情報用処理
DBMSKAN	DBMSKAN.c	項目解析
str_cnt	DBMSKAN.c	キー移動数カウント
strptr	DBMSKAN.c	文字列サーチ
DBMSCMV	DBMSCMV.c	カレント移動
DBMSEQU	DBMSEQU.c	等号処理
DBMSSUB7	DBMSEQU.c	N U L L 処理
DBMSNEQ	DBMSNEQ.c	不等号処理
DBMSSUB8	DBMSNEQ.c	N U L L 処理

モジュール名	ソースファイル名	機能概要
DBMSGRE	DBMSGRE.c	≧記号処理
DBMSGRT	DBMSGRT.c	>記号処理
DBMSSME	DBMSSME.c	≦記号処理
DBMSSML	DBMSSML.c	<記号処理
preset	preset.c	内部関数
sucset	sucset.c	内部関数
chg_lvl	chg_lvl.c	表示用選択ファイル作成
upsel	chg_lvl.c	内部関数
dwysel	chg_lvl.c	内部関数
ufnc1	chg_lvl.c	内部関数
ufnc2	chg_lvl.c	内部関数
print	print.c	出力項目解析
chg_loc	print.c	カレント移動
setloc_err	print.c	エラー処理
prt_data	print.c	データ出力
sht_prt	print.c	内部関数
lng_prt	print.c	内部関数
amt_prt	print.c	内部関数
lstr_prt	print.c	内部関数
dp_prt	print.c	内部関数
sstr_prt	print.c	内部関数
mk_cur	mk_cur.c	カレント設定

訂正 (ppA.8)

上から4行目

/* 有声の破裂音で...

↓

/* 無声の破裂音で...