Introduction to the PhonDat Database of Spoken German

Christoph Draxler
(abridged version of Practical Applications of Prolog '95 conference paper)

Department of Phonetics and Speech Communication
Ludwig-Maximilians-University Munich
Schellingstr. 3
D - 80799 Munich
Tel: +49 +89 2866 9968
draxler@phonetik.uni-muenchen.de[1]

Abstract

The PhonDat project within the German Verbmobil research initiative aims at creating and making accessible a very large database of symbolic and signal data of spoken high German. Currently, the PhonDat database consists of one corpus of sentences containing all phoneme combinations of high German, and of one corpus of sentences from a train enquiries scenario. All symbolic data is held in a Prolog system with a powerful database management system extension; signal data is stored in external files.

The database is accessed through queries over the symbolic data. The result of a query evaluation is either again symbolic data, or a reference to signal files and signal fragments within these files. Two access modes are supported: a toolbox of predefined high-level query predicates for standard, albeit complex, queries; and the full Prolog programming language for custom applications. The PhonDat database interacts with external signal analysis and display applications through interprocess communication.

The PhonDat database has been used in various research applications, e.g. speech recognition training, segmentation comparisons, and statistical phoneme analyses. It is now being extended to hold the Verbmobil multi-language spontaneous speech corpus collected in a scheduling scenario.

Keywords: Phonetic database, PhonDat, Prolog, spoken language processing

1. Introduction

Spoken language processing (*SLP*) deals with the relationship between speech signal, symbolic transcriptions of signals, and orthographic text. SLP is considered one of the key technologies in telecommunications. Speech data collection is actively pursued in many countries, and SLP applications are now becoming available on workstations and PCs.

In SLP, the term *database* refers to a body of speech material. Such a database consists of signal data of recorded speech, a symbolic representation of the signal data, and administrative data. Examples of such databases are TIMIT [19], Verbmobil/PhonDat [14], KTH [1], and

1. This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil project under grant 413-4001-01IV 102 L/4.

SAM [15]; they are used as reference databases in research and for the development of SLP applications, e.g. speech recognition, speech synthesis, speech verification, speaker identification, etc.

Currently, SLP databases consist of little more than the data itself: access to and manipulation of the data has to be programmed explicitly in any application using such a database. Only recently have there been attempts to store data independent of applications in relational (*RDB*), object-oriented (*OODB*), or deductive database systems (*DDB*).

Early attempts based on RDBs (e.g. [8], [11], [12], [16]) have proven to be too restricted for SLP data. The relational model does not support structured or bit-stream data; futhermore, recursion is needed to access internal elements of structures and for the computation of the transitive closure. OODBs provide complex datastructures including bit-stream data. In general, the data manipulation language is integrated into a programming language for full computational power, e.g. CLOS in [10], C++ in [2]. However, there is a considerable semantic gap between the logic based formalisms used in phonology and linguistics and OODB languages. Furthermore, there is not yet a standard OODB language. DDBs provide complex datastructures and a powerful logic based data manipulation language; however, bit-stream data is difficult to handle in DDBs.

For the implementation of the PhonDat/Verbmobil database a hybrid approach using a persistent Prolog system was chosen for the following reasons:

- Prolog is very close to first order predicate logic, the predominant formalism in linguistics.
- Powerful and efficient Prolog environments with access to external or internal persistent databases are available.
- The implementation of database query languages on top of Prolog is straightforward.
- External applications can be accessed through foreign language interfaces and interprocess communications.

Finally, one of the main tasks in SLP is that of mapping signal, transcription, and orthography. This is a complex alignment problem involving multiple representation levels, which can be described elegantly in a logic based formalism and implemented efficiently in Prolog.

The structure of the paper is as follows: in chapter 2 the PhonDat database is presented together with the terminology needed in later chapters. Chapter 3 gives an overview over applications for which the PhonDat database has been used already and chapter 4 discusses performance, query language and interprocess communication issues. Chapter 5 gives a short conclusion.

2. The PhonDat database

The PhonDat database consists of two corpora: PhonDat I is made up of 200 phonemically balanced artificial German sentences plus the North wind fable and a Butter story [9]. PhonDat II consists of 64 pseudo spontaneous (i.e. dialogues recorded at a train enquiries counter were transcribed orthographically and read in a studio) speech sentences from a train enquiries scenario [14].

2.1 Symbolic data

At the International Phonetics Asscociation (*IPA*) meeting in Kiel in 1989 a convention for the computer representation of individual languages (*CRIL*) was defined. According to CRIL, SLP data should be represented on three symbolic levels: orthographic representation, phonemic representation, and phonetic transcription. PhonDat strictly adheres to the CRIL convention.

The *orthographic* representation is a standard text representation. In PhonDat, a *sentence* is a sequence of words. A *word* is a sequence of characters; words are delimited by blanks or punctuation marks.

```
Guten Tag, wann geht morgen vormittag ein Zug nach Frankfurt?
```

In the *citation form* (or *phonemic*) representation a reference pronunciation of words (spoken in isolation in high German) is given. In PhonDat, an extended SAM-PA alphabet is used [14], [15]. Each word in the dictionary has a citation form or *canonic word* associated to it. A canonic word is a sequence of canonic units. A *canonic unit* is a sequence of SAM-PA characters.

```
Feiertag            f 'aI6 t a: k
Frankfurt           f r 'a N k f U6 t
Freitag             f r 'aI t a: k
frueh               f r 'y:
fuer                f y:6+
geben               g 'e: b @ n
...
```

The *phonetic transcription* of an utterance is the result of a segmentation and labelling procedure. In PhonDat, the segmentation and labelling is performed on a phonetic workbench [20] by trained human segmenters or an automatic segmentation program [17]; the output of a segmentation session is a *transcription*.

A *transcription* is associated to a signal, a segmenter, and a segmentation session. It consists of a sequence of segments. A *segment* has a begin time and a duration, and a *label* which is either a marker symbol for paralinguistic information or an action operator with canonic units as arguments.

```
13031 #c:
13031 ##%g
13735 $'u:-'u
14846 $t
15924 $@-
15924 $n
18641 ##t
19437 $'a:
22085 $k
24905 #,
24905 #p:
...
```

2.2 Signal data

Signal data is stored in signal files which consist of a header for administrative data (date of recording, speaker, sampling rate. recording setup, etc.) and the signal data. In PhonDat, data is recorded at a sampling rate of 16 KHz with 16 bit encoding (i.e. 256 KB/s). Typically, a signal file is about 80 to 300 KB of size for an utterance.

2.3 Size of the database

Segmentation and labelling of the PhonDat I database is still under way – only a fraction of the longer texts have been transcribed, whereas the PhonDat II database is now completed. The current sizes of the databases are given in (Tab. 1)

	PhonDat I	PhonDat II
Speakers	200	15
Signal files	> 20000	> 3000
Size (signal files)	4 GB	600 MB
Transcriptions	> 3400	> 5200
Dictionary entries	> 350	> 200

Tab. 1: PhonDat I and PhonDat II database sizes

2.4 Prolog representation of the PhonDat database

The PhonDat database is held in six Prolog relations

```
ipa(IPA,IPAName,Type).
is_a(SuperType,SubType).
sampa_ipa(Sampa,IPA,SampleWord).
word_canword(WId,Word,CanonicWord,FunctionWord).
word_in_sentence(WId,Corpus,SentNo,Pos).
segment_file(FileName,Speaker,City,Segmenter,SentNo,Version,Segments).
```

`ipa/3` matches IPA symbol codes to IPA symbol names and articulatory properties:

```
?- ipa(308, 'lower-case u', Type).
Type = [back, high, rounded]
```

IPA symbols may be classified according to their phonetic properties, and a *type hierarchy* is constructed by successive abstraction from these properties (the top-most element in the hierarchy is a *phone*; on the next lower level there are *vowel*, *consonant* and *diacritic symbol*, etc.). The third argument of `ipa/3` contains the type information, and the hierarchy itself is stored in the `is_a/2` relation (cf.[10] for a similar type hierarchy).

`sampa_ipa/3` maps SAM-PA symbols to IPA symbols and gives a reference word for each SAM-PA symbol.

```
?- sampa_ipa(u,IPA,'Kulisse').
IPA = 308
```

`word_canonic_word/4` is the dictionary of the PhonDat database. `WId` is a system provided unique integer identifier for a word (to distinguish homographs), and `FunctionWord` is a marker.

```
?- word_canword(WId, 'Zug', CanonicWord, FunctionWord).
CanonicWord = [[103], [132], [501, 308, 503], [109]]
FunctionWord = nf
WId = 201
```

word_in_sentence/4 captures the occurrence of a word within a sentence

```
?- word_in_sentence(201, Corpus, SentNo, Pos).

Corpus = train
SentNo = 501
Pos = 9
```

segment_file/7 contains all data relevant to a phonetic transcription. Segments is a list of tuples segment(Begin,Duration,ErrorMeasure,Label), where Label is either a paralinguistic label (e.g. sentence_begin, word_begin, punctuation(Code), error), or a phonetic label. A phonetic label is either a tuple elision(CanUnit), insertion(CanUnit), replacement(CanUnit1, CanUnit2), a pause label, or a canonic unit.

```
?- segment_file(FileName,'AWE','D','CHK',501,0,Segments).

FileName = 'AWED5010.S1'
Segments = [segment(12849, 0, 0, sentence_begin),
segment(12849, 0, 0, word_begin),
segment(12849, 886, 0, arbitrary([110])),
segment(13735, 1111, 0, [501, 308, 503]),
... ,
segment(68838, 0, 0, punctuation(977))]).
```

2.5 Implementation

The PhonDat database is implemented in Eclipse, the logic programming environment of ECRC [4], and LPA MacProlog on the Macintosh.

The user selects the preferred signal display and/or analysis applications and queries the symbolic database. The result of a query in the symbolic database is either output to the screen, a reference to one or more signal fragments within signal files, or is added to the symbolic database as new knowledge.

The signal analysis and display applications are external applications. The PhonDat database accesses them via remote procedure calls or message passing (via AppleEvents on the Macintosh and TCP/IP under UNIX). The external applications receive from the database addresses within signals and can then load the corresponding signal fragment from the signal database (e.g. on CD-ROM).

3. PhonDat Application Programs

The PhonDat application programs can be divided into three categories

- Conversion programs
- Predefined query predicates
- Complex applications

All PhonDat database predefined queries and sampel application programs are described in [3].

3.1 Conversion programs

Conversion programs convert a given form into another representation. Typical examples are the translation of SAM-PA to IPA and back, or the translation of user input into an internal format.

sampa_ipa/2 converts canonic words, canonic units and segments into the corresponding IPA representation. It is implemented to work in both directions.

user_input/2 requires in its first argument a type specification in a Prolog string (or atom) and returns the corresponding IPA representation.

```
?- user_input(''a:,plosive,f',IPA).

IPA = [[501,324,503],plosive,[103]]
```

3.2 Predefined query predicates

Predefined query predicates constitute the toolbox for the phonetic database – they provide the basic set of operations which are of interest to the phonetician.

As an example, the predicate type_in_canonic_unit(Type,CanUnit) checks whether the canonic unit CanUnit is of a given type Type.

```
?- type_in_canonic_unit([*,vowel],[501,308]).

yes
```

The following complex goal checks whether the canonic word corresponding to a word contains canonic units of the type vowel, and if so, returns the SAM-PA form of the typed canonic units.

```
?- word_canonic_word(_,Word,CanWord,_), member(CanUnit,CanWord),
     type_in_canonic_unit([*,vowel,*], CanUnit),
     sampa_ipa(SampaCU,CanUnit),
     sampa_ipa(SampaCW,CanWord).

Word = 'Aachen'
CanWord = [[113],[501,304,503],[140],[322],[116]]
CanUnit = [304]
SampaCU = 'a:
SampaCW = Q 'a: x @ n
```

The set of predefined query predicates consists of approx. 40 predicate definitions.

3.3 Complex applications

Predefined query predicates are ideal for ad-hoc queries to the database. For statistical computations which compute values over collections of solutions, a more powerful database access is needed. Furthermore, the efficiency of a query evaluation of a predefined query predicate depends on the instantiation of its arguments. Application programs, for which the instantiation of arguments does not change, can be optimized for efficiency.

Currently, Prolog application programs have been developed for these tasks:

- Labelling consistency comparisons
- Analysis of vowel duration in high German

Labelling consistency comparisons

Transcriptions of the same signal produced by *i*) different and *ii*) the same segmenters were compared to allow a classification of phones into clear and unclear cases with respect to the transcription labels used [5], and with respect to the segment boundaries [6]. The main result is that the intraindividual consistency is not significantly better than interindividual consistency and that the consistency depends strongly on the class of phone found in the signal.

Analysis of vowel duration

For a complete phonetic theory of a language, duration data for the phone inventory of the langauge is needed. The PhonDat I corpus contains all dyadic phoneme combinations of high German, and is thus well suited for the analysis of durations. Furthermore, since the speakers were instructed to speak in a casual speaking style, the recorded speech is close to natural speech.

The main objective of these analyses is to gather empirical data for phonetic theories for spoken German.

4. Discussion

Two aspects of the implementation are now being discussed.

- Performance
- Interapplication communication between the database and external applications

4.1 Performance

The main goal of the PhonDat database implementation has been to provide the functionality needed for the tasks to be worked on, and efficiency has been a lesser objective only. However, efficiency is fully sufficient for interactive work.

The following benchmarks (Eclipse 3.4.5 on a Sparc 10) may serve to give an impression of the performance. The database contains 207 dictionary entries, 677 word occurrences, and 5268 segmentations with a total of over 350.000 segments:

- Compute the number, sum, and the average duration of all phonetic (i.e. non-zero length) segments: 68.5 s
- For each type class of phonemes, if there exists more than one manual segmentation of an occurrence of a canonic word, compare the manual segmentations with the automatic segmentation: 236.6 s

4.2 Interapplication communication

The PhonDat database is one tool among others for workers in the field of SLP. As such, communication with other applications is of great importance since this allows to make use of the many signal analysis, signal display, and other signal processing applications.

On the Macintosh, AppleEvents are used to communicate with other applications. For example, to display the signal corresponding to a given type, the database is queried and the resulting signal address is sent to a display software together with a command to select the appropriate signal fragment, to compute its spectrogram, and to output it via speakers (Fig. 1):

In Eclipse, the interprocess communication built-ins are used to drive external applications via sockets in the TCP/IP domain. This includes network access to the PhonDat database, as well as interchange with applications on remote machines.

5. Conclusion

Despite its restricted amount of transcription data, the PhonDat database has shown to be a valuable tool for empirical studies in Phonetics. The expressive power of the database queries is very high, and the class hierarchy of articulatory descriptors is a powerful means of formulating queries.

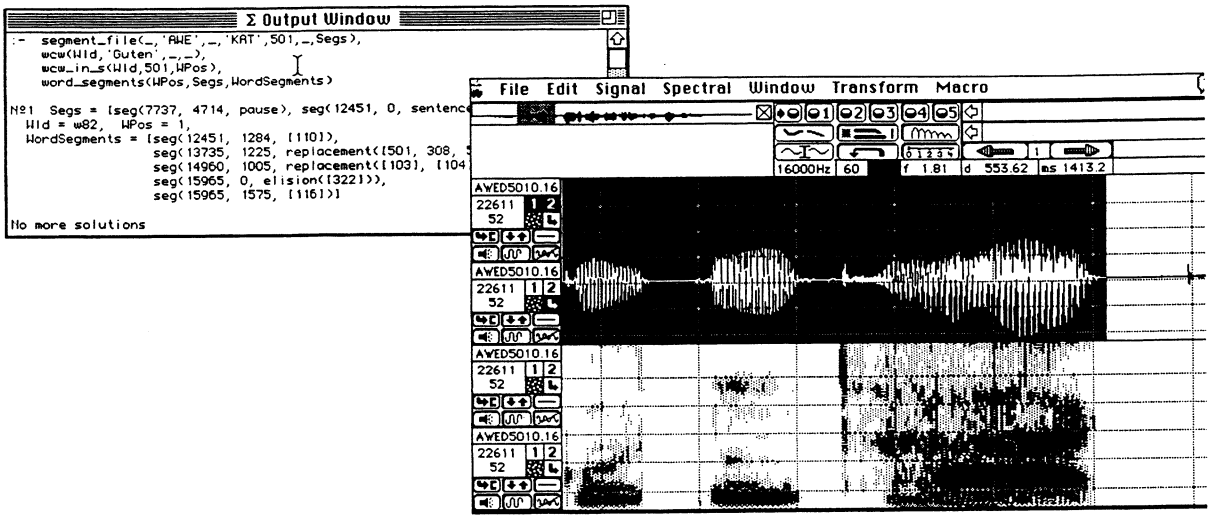

Fig. 1: Signal Output for Database Query

However, Prolog is not well suited as a database query language, at least for novice users: identifying arguments by their position instead of by name is problematic for predicates with more than five arguments. Efficiency has never been a problem; the advantage of having portable code clearly outweighs any platform specific optimizations.

Future work will include predicates that call sound processing applications into the database language, e.g. for filtering, signal computations, etc. Finally, additional layers of symbolic information, e.g. prosody, should be integrated into the database.

References

[1] R. Carlson, B. Granström, L. Nord: The KTH Speech Database. Speech Communication Vol. 9, No. 4, 1990

[2] SPEX: S. Swagten: Speech Processing EXpertise centre, Leidschendam, The Netherlands: private communication

[3] C. Draxler: The PhonDat Database Handbook. Internal report, Institut für Phonetik, Universität München, 1994

[4] ECL^{i}PS^{e}: ECRC Common Logic Programming System, User Manual, ECRC 1992

[5] B. Eisen, H. Tillmann, C. Draxler: Consistency of Judgements in manual labelling of phonetic segments: The distinction between clear and unclear cases. ICSLP Banff, 1992

[6] B. Eisen: Reliability of Speech Segmentation and Labelling at Different Levels of Transcription, Eurospeech 93, Berlin

[7] J. Esling: Computer Coding of the IPA: Supplementary Report. Journal of the International Phonetic Association, vol 20, No 1, 1990

[8] J.P.M.Hendriks: A Formalism for Speech Database Access, Speech Communication, vol. 9, No. 4, August 1990, ppg. 381 - 388

[9] THE PRINCIPLES OF THE INTERNATIONAL PHONETIC ASSOCIATION, 1949 (Reprinted 1984), International Phonetics Association, London, 1949

[10] M. Karjalainen, T. Altosaar: An Object Oriented Database for Speech Processing. Eurospeech 93, Berlin

[11] L. Kuffer: Der Einsatz eines relationalen DBMS zur Verwaltung von segmentierten und etikettierten Sprachsignal-Daten. Magisterarbeit, Institut für Phonetik, Universität München, Nov. 1991

[12] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, K. Shikano: ATR Japanese Speech Database as a Tool of Speech Recognition and Synthesis. Speech Communication, Vol. 9, No. 4, 1990

[13] D. Stott Parker: Stream Data Analysis in Prolog. In: L. Sterling, The Practice of Prolog, MIT Press, Cambridge MA, 1990

[14] B. Pompino-Marschall: PhonDat Daten und Formate. Institut für Phonetik, Uni München, 1992

[15] SAM: Assessment, Methodology and Standardisation in Multilingual Speech Technology. Int'l Symposion on Coordination and Standardisation of Speech Database and Assessment Techniques for Speech Input/ Output, Nov. 1993

[16] Chr. Saßenrath: Entwurf und Implementierung eines datenbank-gestützten Verwaltungssystems für Sprachanalysedaten, Diplomarbeit Uni Erlangen, 1991

[17] F. Schiel: An automatic segmentation program based on HMMs (working title), internal report, to appear.

[18] D. Searls: Signal Processing with Logic Grammars. TR Paoli Research Center, Unisys Co., 1989

[19] St. Seneff, V. Zue: Transcription and Alignment of the TIMIT Database, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA, 1988

[20] H. Tillmann, M. Hadersbeck, H. Piroth, B. Eisen: Development and Experimental Use of PHONWORK, a New Phonetic Workbench, ICSLP 1990