Data Modelization Within and Requests From an Acoustic Phonetic Data-Base

Nadine Vigouroux *

ABSTRACT

This is an introduction to an Acoustic Phonetic Data-Base (APDB) that is constructed on an object-oriented model of representation. This model allows both to express and to handle acoustic and phonetic objects that are either multiple or complex. Access to this base will be illustrated through a few sample requests.

INTRODUCTION

Systematic studies of phonetic message structure requires both setting up and operating data-bases as well as multi-media knowledge-bases (ref. 1, 2). Any approach to vocal message characteristics --based on an observation of acoustic phenomena-- is bound to be complex. This observation phase of acoustic data cannot mean much, unless both a fairly important sampling of data, and a language that is adapted to representation of this data are available. Indeed, those investigating the feasibility of Automated Speech Processing (ASP) are eager to have on hand a full set of analytic functions that can deal with the acoustical, the statistical and the learning aspects of their question. In order to meet their needs, it is necessary to define an acoustic analysis station that adequately corresponds to the criteria involved. In the present article, a presentation is made of the object formalism that is used to design the base. This is followed by a description of how this formalism is set to work both in managing acoustic phonetic data and in consulting this data.

REPRESENTATION MODEL

Description of Acoustic and Phonetic Information

The APDB is equipped with a whole set of functions that allow the user to analyze the vocal signal, to label it according to a cognitive model, to visualize it, to handle at will the results issuing from these various tasks, etc. The APDB software system accommodates acoustic information items as a collection of so many attributes which characterizes a speech entity at the acoustic phonetic level. These attributes are either numerical or symbolical. Some are uni-, others pluri-dimensional (vectors). The acoustic and phonetic information we consider is limited to the following items:

- quantitative data that consists of numerical acoustic items that are yielded by an application of acoustic analysis modules, such as filter-banks, Fast Fourier Transforms, as well as by an application of functions designed to compute spectral, formantic, prosodic, ... cues (ref. 3).
- qualitative data that amounts to an identification of conceptual events --interesting linguistics (phoneme, acoustic phase, ...)-- which are detected by a phonetician (ref. 4) during labelling sessions.

Already, the description supplied above points not only to the diverse nature (acoustic, phonetic, intuitive) of the information used, but also to the plurality of its types, or structures (numerical, conceptual). The problem, faced here, consists in having to define, on the one hand, a data model that is suitable for both representation and handling of these entities within a data-base and, on the other hand, a convivial environment for consultation of this base.

Concept Used

Various investigative efforts bearing on both data- and knowledge-bases, would suggest a representation of the "structured" type --e.g., script, frame, object (ref. 5). These would make it possible to take into account, within one and the same structure, both the diverse characters of type and the semantic aspect of data. An object modelization appears to be more suitable, however, for this kind of formalism allows to define the concerned semantic entities within a data-base.

* Laboratoire C.E.R.F.I.A., UA-CNRS 824, 118 Route de Narbonne, 31062 Toulouse Cedex, France.
The semantics involved corresponds to both the structure properties and the behavior of the objects brought to use. Our approach, therefore, consists in designing the data-model as a collection of structure objects affording both sequencing and equivalence data-links. The APDB model consists of:
- a design-diagram that represents the set of constraints and rules for the structuring of information,
- the base of object-descriptions, containing the images of modelized information items.

**Defining an Object**

In the proposed system, an object is defined (ref. 6) by:
- its identification (name, class-name),
- a set of instance-variable declarations granting a status to the object,
- a set of methods altering a state.

More formally, objects can be defined through using the following syntax:

\[ \text{OBJECT} = (\text{Object-name}, \text{Class-name}, \text{State-name}, \text{Method-name}) \]

where:
- **Object-name** is the name of the object,
- **Class-name** is the name of the class from which it inherits,
- **State-name** is the name of the table specifying the object,
- **Method-name** is the name of method table of the object.

**The State-name table** contains the characteristics of attributes; i.e., their number, their type and domain of definition. These attributes can also be pointers trained at objects belonging to other classes. There are as many entries, in this table, as there are attributes to specify in order to describe an object.

**The Method-name table** contains the list of all these functions that may be applied either globally to the object (either to create it or destroy it) or locally onto an attribute. Each recorded item includes the name of the function, the list of both input and output parameters.

**THE APDB**

**Concept-Design Diagram**

Designing in terms of object-type makes it mandatory both to sequence and to rearrange data into classes. The concept diagram meets this requirement. Figure 1 shows the chronology of the acoustic and phonetic information items produced. Signal acquisition has been realized already (ref. 7). An analysis of these items allows to assign:

- the spectral-cue and formantic entities to the general class PARAMETER,
- the Fine Frequential Labelling (FFL) and Broad Labelling (BR) entities to the class LABEL.

For example, all objects generated in the class LABEL inherit both the "expertise" type and the "alphabetic" type. Applying the method ACOUSTIC-ANALYSIS to the SIGNAL-object will impart to it a new state, while generating a new SPECTRUM-object --thereby setting an inheritance relationship. A bar-chart links up the SIGNAL- to the SPECTRUM-object, while a (canonical) correspondence or "link" will associate a LABEL-object to this SPECTRUM-object --thereby setting a synchronic relation. This is a concept phase that is entirely achieved by an expert.

**Object Description**

By analyzing both data and requests, it can be shown that two types of attributes are needed to define the following objects:

- "Data"-type attributes, which represent properties (e.g., for a sample spectrum, its description will be: energy, duration, fundamental, etc).
- "Relation(ship)"-type attributes. In order to solve the problems involved in acoustic phonetic decoding, it is necessary to "bridge" all numerical and conceptual information items. These attributes allow to secure correspondences among signal-, spectral-, label- and phonetic transcription-block numbers (addresses). Therefore, they allow to establish collateral relations among objects in the base (e.g., the SPECTRUM- and the LABEL-object are set in correspondence through the spectral sample number). These attributes are akin to "joining", as it is used in classical DBMS's (Data-Base Management System).

This modelization design fully meets the requirements set by acoustic phonetic decoding for, within one and the same structure, it allows to associate data items of several different types, and to interrelate all the created objects through a meshwork.
CONSULTING APDB

Goal
The goal is to offer a set of means (both objects and methods) to the user who wishes to construct his/her own experimentation base—e.g., a phonetician may design a procedure for experimenting. Therefore, the goal is both to give the best of transcriptions for this procedure—consisting in tests—and to allow the phonetician to acquire some item of knowledge and/or to validate a hypothesis.

Any request, on the user's part, is changed into a message sent to the system, so that a piece of information is looked for and/or generated. This is not dealing with automated learning procedures.

For any request to be carried out, three phases are involved:
- user's specification of request,
- request transcription under internal form language,
- execution of request.

Request Classification
Two different levels of request complexity should be distinguished:
- Simple requests. Those consisting in selecting a previously defined object, and in applying to it a number of methods. For example:

What are the addresses, within File xx, of all realizations of phoneme /i/?

Every request comes into being through reiterating several sequences of this type:
- Selection of an object,
- Application to it of a set of methods.

Therefore, the above request translates as follows:

Select-Object : Phoneme
Apply-Methods : Instance the attribute name-phoneme by /i/
Select-Object : File
Apply-Methods : Instance the file name by xx?,
Open file.

===> Creation of a temporary object: PHONEME-FILE to which a filtering operation is to be applied.

- Complex requests. They call upon already defined objects (within the base) to intervene. Such objects have a more complex structure; constructing them brings about a need for logical operators.

Example: What are the realization of phonemes /i/ and /a/ that enjoy a fricative modality and an intensity greater than 50dB.

Diagrammatically, every request specification consists of the following three tasks—which constitute so many stages of embodiment of the request:
- Designation of an object; i.e., its name and its class,
- Instancing within context of this object,
- Conditions set on the attributes of this object.

Both context and conditions can, in turn, be defined in terms of objects.

Figure 2 illustrates the different relations between objects.

The carrying out of the request constitutes, per se, the instancing of the objects, specified by the user. In this sort of request, instancing calls in but algorithmic processes, whether this be for computing properties or for locating information items already in the base.

CONCLUSION

In this article, we describe how the concept of object is used within an APDB, in connection with production, with management and with the handling of data. A set of basic objects and of methods are already implemented, and they satisfy the previously described concepts. The selection of objects and the instancing of these both allow a user to create an experiment base, on which it is possible to let learning procedures operate (ref. 8). In ref. 9, the description-base of APDB objects, the request specification system and their execution should become more explicit.
ACKNOWLEDGMENT

I gratefully acknowledge the contribution of J.F. Malet of CSU Sacramento, who translated the original French text, and provided many insightful remarks about this article and for future work.

REFERENCES