PREDBASE—CONTROLLING SEMANTICS OF SYMBOLIC STRUCTURES IN MUSIC

Oliver Zabora
MultiMedia Lab, Institut für Informatik, Universität Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
zabora@ifi.unizh.ch

Abstract

PREDBASE is a database management system, capable of handling flexible and open semantic data structures. It was developed with the complexity of musical score notation in mind, as part of the RUBATO performance workstation. The high demands of the classical musical score relative to handling polymorphism and polymony of data are satisfied by a unique design, developed according to perceptions of structuralist semiotics and predicate logic. Moreover, the possibilities given by logical and geometric motivation allow for new ways to retrieve, inspect and relate the data at hand.

1 Introduction

The representation and processing of musical score data is an often discussed issue in the field of computing musicology and algorithmic composition. Problems do not only arise when capturing music with a given set of predefined signs, but also when translating these signs into a machine readable and processible format. This is due to the context dependent semantic flexions and determinations inherent to most natural human sign systems:

- In natural sign systems, signs coexisting on the same level represent objects living in completely different parameter spaces.
- The number and type of parameters may change from instance to instance.
- There are cases where no concrete or default value is available to be assigned to a certain parameter.

EXAMPLE

The notes of a piece, found on a page of music written in common notation, may be fully represented in the space of (note, pitch, dynamics, duration), whereas the rests, for example, live in the sub-space of (notes, duration) and do not have length or pitch parameters.

To cope with the above problems, a database management system must fulfill the following specifications:

- The data structures must be flexible with respect to hierarchy and type, user configurable and expandable at run-time.
- The assignment of meaning—i.e. the input of concrete values like strings and numbers—may be postponed, while the system remains operable.

Common database management systems know until now do not allow for appropriate representation and manipulation of such complex sign systems (Vossen, G.).

2 PREDBASE and the Predicate Calculus Theory

By further study of the diversity of musical signs and connecting this with structuralist semiotics, we introduced the concept of Predicates to our database management system, called PREDBASE.

The Predicates in PREDBASE are an adapted and expanded version of the well-known structuralist sign model of Saussure, where signs are units of significant—signifier and the mediating signification (Saussure, F.). To be able to use such signs on a formal level—for the operations required in analysis of music, say—the traditional references to signifier (name) and signified (value), are enriched by the form resp. type of signification.
2.1 Signification and Form

The signification is the process of connecting the two parts significans and signifié, i.e. the sensibility of the sign. In our specialized context, the signification tells first whether or not the value of a Predicate is a reference to another Predicate. If it is a reference, it gives further information about the kind of referenced Predicate at hand. Otherwise, it tells what kind of value is to be expected for retrieval. In our model, the signification is represented by a Form of a certain type. Let us first explain how the Form works.

The Form describes the structure of a Predicate and therefore also the admissible operations—in particular mathematical ones—on any type of Predicate. Intuitively, the Forms describe all possible instances of Predicates. According to the classical Aristotelian distinction of form and substance, the actual Predicate instances (of a particular Form) are called the substance of this Form.

2.2 Simple and Compound Predicate Types

In addition to the classical simple Predicates, there exist more complex composed ones, the compound Predicates. They are references to other simple Predicates or collections of compound Predicates again.

Simple Predicates are those referring to a single value representing number or text information, or no value at all—e.g. (empty), 0 (a size), 1 (an integer), f (a floating number), (a boolean value).

Compound Predicates are those which refer to one or more other Predicates, namely: Synonym (the reference to any other Predicate: “Flute”⇒“Flute”), Product (a fixed list of Predicates, as in a parameter list: “Note”(Onset, Pitch, Loudness, Duration), Copredicate (choice of one from a given list of possible Predicates: “Accidented” a Sharp or Flat), Power Set (a list of variable length composed of equal Predicates: “SharLegato”()“Note1”, “Note2”, “Note3”)).

Now that we know all the components, let us give the recursive definitions of Form and Predicate:

**Definition 1**

*Form(NAME, TYPE, VALUE)*

The Form is an arbitrary sign with a fixed signification paradigm, denoted by the Form's type. It represents the convention how Predicates are built in a certain environment. Since they reference other Forms, compound Forms are recursive by nature.
**DEFINITION 2**

Predicate(NAME, FORM, VALUE)

The Predicate is an arbitrary or motivated sign built according to the convention of its Form. The signification paradigm is restricted to a single signification by assignment of a value.

**FIGURE 3**

Within Predibase, the Predicate is an object containing the signifier's string value and appropriate substance values, according to its referenced Form.

- **NAME**: the signifier of the Predicate, normally identical to the signifier of its Form
- **VALUE**: a computational value
  - simple
  - compound
  - a reference to one Predicate: (+, 1)
  - a reference to a list of Predicates: (π, (1))

- **FORM**: the common signifier used by the Predicates of a Form
  - **NAME**: a computational default value
    - simple
    - compound
  - **VALUE**: a reference to a list of Predicates
    - simple
    - compound
    - a reference to one Form
      - simple
      - compound

2.3 Total Order of Predicates

Any database management system must be able to store, retrieve and visualize data in a desired logical order. If the data at hand has no genuine order, artificial ordering by indexes, with its undesirable side effects, takes place. In Predibase, all substance of simple or compound, arbitrary or motivated Predicates is canonically ordered, since Forms naturally define a unique linear order among all Predicates.

**DEFINITION 3**

Simple Predicates are ordered according to the linear orders on their value domains (real numbers, strings).

Products are ordered by the cartesian product of their factors.

Coproduts are ordered by the indexed disjunct union of the orders of the cofactors.

Power sets are linearly ordered by their substance. For two non-empty sets A and B we define ("max" is only defined for non-empty differences!)

\[ A > B \iff (A \supset B) \lor (\max (A - B) > \max (B - A)) \]

3 The Paradigm of Logic and Geometric Motivation

One of the difficult tasks in musicoLOGY is to transform the given symbolic structure of a musical piece into a derived symbolic structure, suitable to be processed by analytical algorithms. Assuming all relevant data is available as Predicate instances, we need tools to retrieve and relate this data in order to make new Predicates. In other words, the application of operators produces motivated Predicates from a given set of arbitrary (or already motivated) Predicates. For this transformation, we introduce logical and geometric operators.

- **Logical operators**: negation, conjunction, disjunction, universal and existence quantifiers.
*Geometric operators: pseudoaction and retract of products resp. restriction of coproducts and their canonical extension to power sets of products and coproducts, fiber products of Predicates.*

Observe, that all these operators act on the *instances* of Predicates. In general, before applying any other operator to a global substance, the substance must be restricted and projected onto a common Form since, by definition, all operators only operate on Predicates of one and the same Form and result always in Predicates of the original Type. In case of negation, conjunction and disjunction, the resulting Predicates are even of the original Form. Since we cannot predict what Forms and operators will be combined while using an actual PrediBase, it is essential to know that logical and geometric operators—no matter how they are applied—do not create new (unknown) types of Predicates.

### 3.1 Visualization

Since we cannot rely on any fixed record structures, an intelligible visualization of the Predicate substance stored in a PrediBase seems to be a difficult task. But the way how Forms define a canonical order among Predicates relieves us from imposing unnatural orders and eases visualization concepts. Although we dispose of approaches—mainly generalizations of the usual view of sounds within parameter frames, as they appear in sequencers, as well as the lexical search geometry in a dictionary—to visualize PrediBase substance in two-dimensional views, we still are in need for the tools to intuitively navigate through multidimensional virtual data spaces. Some approaches are seen in the field of Virtual Reality—for example an *Audio Browser*, shown at the ICMC94 (Whitehead, J. F.), where the user has the impression of moving around inside a database of sound objects, laid out in a three-dimensional space.

### 4 Summary

Designed with the richness and complexity of classical musical scores in mind, PrediBase allows for records with variable dimension, a priori unknown values and semantic depth. Data—i.e. Predicate instances—is always kept in a canonical linear order determined by the Predicates’ respective Forms. This order defines natural ways of navigation through and visualization of the data. Predicates can be related, interconnected and retrieved as desired through logical and geometric operators. Interestingly enough, the results of the application of these operators are Predicates again, which may undergo the same processes.

Currently, a PrediBase prototype—implemented as a hierarchy of Objective-C classes under the NEXTSTEP development environment—is used as the underlying database system for the representation and retrieval of symbolic score data in the RUBATO performance workstation (Mazzola G. & Zahorka O.).

### 5 Acknowledgements

I would like to thank Guerino Mazzola, without whom none of this work would have been possible. Financially, this research was made possible by a grant from the Swiss National Science Foundation.

### 6 References


