A Toolkit for Music Processing and Analysis

Bruno Azevedo

Informatics Engineering Msc
University of Minho

Professor José João Dias de Almeida

February 2013

Context and Motivation

- Textual Musical Notation
  - Abc
  - LilyPond
  - MusicXML

- Unix Philosophy

  “Write programs that do one thing and do it well. Write programs to work together. ...”

  Doug McIlroy

Case Study

Wiki::Score

- Wiki
- Cooperative edition of music scores
- Abc Notation

Solution

Toolkit

- Small problems
- Articulate results

Design Goals

Toolkit

- Automatic Validation
- Error Detection
- Error Fixing
- Statistical and Musical Analysis

- Open-Source
- Unix Philosophy

Design Goals

Musical Corpus

- Corpus of music scores
- Tools for statistical calculation

Design Goals

Musical Information Representation

- Keep order of symbols
- Hold indispensable information
- Facilitate scripting

Design Goals

Musical Information Visualization

- Appropriate format
- Intelligible output
- Reveal some feature

Musical Notations

Abc

- Notation standard
- Plain text
- Compact and clean syntax
- Human readable
- Open source
- Original Goal: Share folk tunes via textual format

Musical Notations

LilyPond

- Software Package
- Plain text
- Friendly syntax
- Open source
- Original Goal: Print scores that are similar to hand engraved scores of the past

Musical Notations

MusicXML

- XML-based format
- Plain text
- Easy to use syntax but not human readable
- Proprietary but freely usable under Public Licence
- Original Goal: Share scores between applications and archive them for use in the future

Projects and Tools

abcm2ps

- Translates Abc to sheet music scores
- PostScript or SVG format
- *De facto* standard typesetter
- Actively maintained

Projects and Tools

abc2midi

- Converts Abc to MIDI
- *De facto* standard MIDI generator
- Part of abcMIDI package

Projects and Tools

EasyAbc

abcpp Preprocessor

Abcp

Music::Abc::Archive

Music21

...

Musical Corpora

What for?

- Calculate patterns
- Assess what is expected
- Calculate similarities
- Generate statistics
- Testing material
- Data sets to train systems that learn from data

Musical Corpora

What can be analyzed?

- Find sets of vertical patterns
- Find significant statistical differences between melodies
- Identify trends and changes throughout a historical time period
- ...

Internal Representation of Musical Information

Structure

- Depends on its final purpose
- Mainly two types
  - Sequential
  - Hierarchic
- Horizontal and vertical dimensions
  - Sparse matrix
- The completeness of a structure is determined by its ability to support all tasks required by its purpose

Internal Representation of Musical Information

Early conclusion

- Sequential and Hierarchic structures are more suited to horizontal readings
- Structures like sparse matrices allow both horizontal and vertical readings
- The latter does not maintain the original order of symbols
- The completeness of a structure is determined by its ability to support all tasks required by its purpose

abcm2ps's approach

Internal Representation

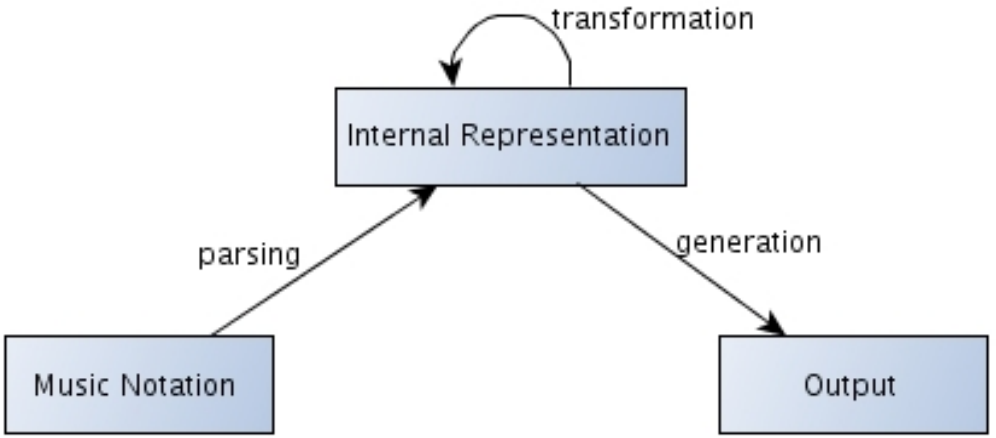
- Sequential structure
- Ordered list of symbols
- Contains all data from an Abc score
- Appropriate to apply scripting
- Basis for a more complex structure

Abc Scripting

Proof of Concept

- Three stage process

1. Data Extraction/Parsing
2. Transformation of the generated representation
3. Output Generation



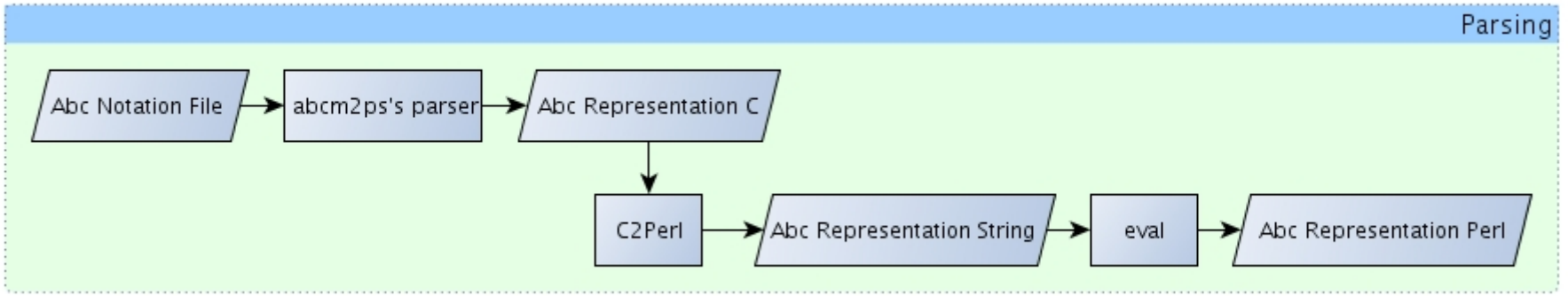
1. Data Extraction

Parser

- Abcm2ps's parser

Internal representation adaptation

- C2Perl program



2. Transformation of the generated representation

Generic Processor

- Higher-Order Processing
- Systematic
- Set of processing strategies
- Inspired in XML::DT

3. Output Generation

- Output in specific format
- Output may have same type as input

Example

```
1  use ABC::DT;
2
3  my $file = "adeste.abc";
4  my %measures = ();
5
6  processor (
7    $file ,
8    (
9      'bar'    =>  sub{ $measures{$voice}++ },
10     '-end'   =>  sub { foreach (sort keys %measures) {
11                           print "$_ has $measures{$_} bar(s).\n"
12                         }
13                       }
14   )
15 );
```

```
Soprano has 21 bar(s).
Alto has 20 bar(s).
Tenor has 21 bar(s).
Bass has 21 bar(s).
```

Conclusions

- Planning the corpus construction is essential for the overall quality of the toolkit
- The internal representation must be complete enough to allow the application of analytic tasks
- The sequential structure is appropriate for scripting
- The proof of concept proves that it is possible to reduce the complexity of an elaborate problem by tackling its parts through scripting

Research Planning

- Resume the development of the processor
- Verify parser's robustness
- Create a set of tests for the toolkit
- Create Unix-like tools for music
- Build Abc Corpus
- Develop statistical and musical analysis tools
- Write the dissertation

A Toolkit for Music Processing and Analysis

Bruno Azevedo

Informatics Engineering Msc
University of Minho

Professor José João Dias de Almeida

February 2013