

O processo de desenvolvimento de uma codificação para definir estruturas musicais

Pedro Kröger, Marcos Sampaio, e Givaldo de Cidra
Universidade Federal da Bahia
pedro.kroger@gmail.com

Sumário:

O trabalho de codificação de estruturas musicais com um alto grau de abstração ocupa um espaço ainda não preenchido na área de computação musical. Esse artigo descreve a pesquisa onde linguagens de domínio específico estão sendo criadas em Lisp para testar possibilidades de codificações segundo um modelo bottom-up. Ou seja, diferentes aspectos de uma codificação são testados em um protótipo e sua utilidade é validada ou descartada. Até o momento alguns sub-problemas foram identificados com a ajuda dessa abordagem, como a utilização de eventos, codificação das durações, e codificação vertical. O uso de ferramentas específicas de software livre colabora com a organização e dinamismo dessa pesquisa e todo o código gerado é licenciado sob a Licença Pública Geral (GPL).

Palavras-Chave: Informática em música, codificação musical

Introdução

A codificação musical é a busca da representação de elementos sonoros. Exemplos simples de codificações incluem a representação das notas musicais por letras (C para dó, D para ré, e assim por diante) ou números (0 para dó, 1, para dó#, 2 para ré, e assim sucessivamente). Codificações musicais têm sido usadas desde os primeiros esforços do homem para transcrever sons. Ainda que a codificação musical não esteja limitada a aplicações para computador é nessa área que encontramos o maior número de pesquisas, dadas as possibilidades de processamento oferecidas (Selfridge-Field 1997, p. 3–5). Existem codificações para performance, síntese sonora, jogos, notação musical, braille, bibliografia musical, análise de canções folclóricas, dentre outras (Kröger 2005). Novas codificações como SDML e MusicXML (Good 2001), em geral, lidam apenas com elementos de notação como compasso, notas, durações, etc. Algumas pesquisas lidam com o conceito de “unidades composicionais” (Berggren 1997) e reconhecem a necessidade de agrupar a representação em unidades como “frases” (Roland 2000), contudo não há nenhuma pesquisa envolvendo uma codificação específica para definir estruturas musicais em um nível mais alto de abstração.

Nós estamos desenvolvendo uma codificação que permite lidar com dados musicais em um nível mais alto de abstração que alturas e duração como “mostrar todos os sujeitos das fugas do Cravo Bem Temperado de Bach”, ou “mostre todas as respostas reais das fugas do Cravo Bem Temperado” (o que na verdade significa “pegue todas as respostas, transponha uma quinta abaixo, compare com os sujeitos e mostre as que são reais”). Outros exemplos incluem acordes para o estudo de orquestração, elementos formais, harmônicos e motivicos.

```
(event 'coral-5
  :key '(c major)
  :time 4/4
  (event 'frasel
    :partial 4
    (event 'soprano
      :relative 5
      :notes "c c8 b a b c d e4 d d c")
```

Figura 1: Trecho de codificação em Lisp

O processo de codificação

Em vez de escolher a abordagem de definir primeiramente uma representação e implementá-la em seguida (modelo *top-down*), escolhemos a abordagem de escrever programas para aprender conceitos (Krishnamurthi 2006). Ou seja, diferentes aspectos de uma codificação são testados em um protótipo, e sua utilidade é validada ou descartada. A vantagem dessa abordagem é que idéias e problemas sobre uma codificação podem ser rapidamente experimentados e entendidos antes de serem definidos mais formalmente (como em uma especificação).

Para implementar o protótipo nós usamos a metodologia *bottom-up* de desenvolvimento e a linguagem de programação Lisp devido a seu poder, flexibilidade e eficiência. Lisp tem se mostrado uma excelente linguagem para a criação de linguagens de domínio específico (LDE) (Graham 1993). Dessa maneira nós criamos uma LDE em Lisp que define a codificação desejada. Isso permite testar na prática a codificação o mais cedo possível.

A codificação é definida usando expressões simbólicas como pode ser visto na figura 1. Uma das vantagens dessa abordagem é que não é necessário construir um analisador léxico e sintático para a linguagem já que a linguagem é “embutida” em Lisp. O parseamento é provido por Lisp.

Problemas específicos

O trabalho de criação de uma codificação para estruturas musicais revelou alguns problemas específicos relativos a cada tipo de estrutura envolvida. Até o presente momento trabalhamos com corais e acordes de orquestra. Em cada um deles há várias estruturas que podem ser consideradas relevantes para descrever sintaticamente. As estruturas trabalhadas até agora foram forma e harmonia no coral; e orquestração nos acordes de orquestra.

Eventos

Utilizamos a idéia de evento para criar a codificação para as estruturas musicais. Uma das vantagens desta escolha é que os eventos podem ser hierarquicamente organizados e que algumas estruturas musicais podem ser enquadradas em uma hierarquia. No caso do coral, percebemos que dois tipos de elementos podem ser organizados hierarquicamente: seções formais e a harmonia. No que se refere às seções formais, há uma hierarquia entre os períodos, frases e cadências. As cadências são contidas por frases, que são contidas por períodos, e o coral — a estrutura completa — contém todos. Em relação à harmonia, os acordes estão contidos em regiões cênicas. Estas regiões podem ainda estar contidas

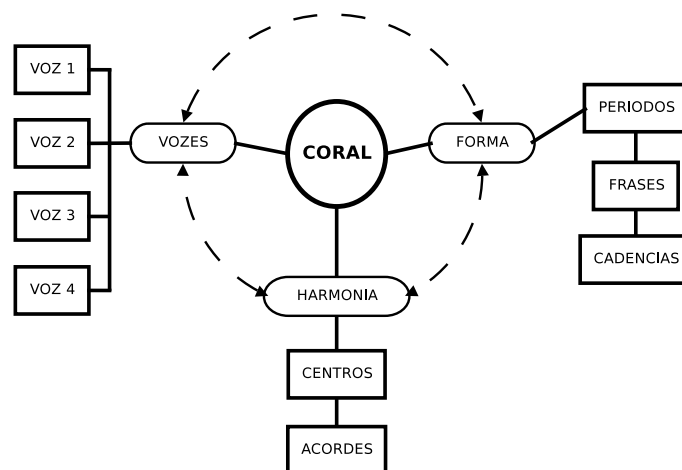


Figura 2: Hierarquias do coral

em regiões maiores. O ítem mais elevado desta hierarquia é também o coral como um todo. Há outros elementos musicais que são hierarquicamente contidos pelo coral em si. Um exemplo é o conjunto das vozes (soprano, contralto, tenor e baixo).

Os elementos “forma”, “harmonia” e “vozes”, no entanto, não estão relacionados entre si de forma hierárquica. Eles são, ao mesmo tempo, dependentes e independentes entre si. (c.f. figura 2). Este tipo de relação pode ser entendida pelo seguinte exemplo: as vozes não estão sub-relacionadas à harmonia ou à forma, porém, a forma e a harmonia são o resultado do comportamento das vozes. Podemos pensar também que as vozes se comportam de acordo com uma determinada forma. Um outro exemplo é a harmonia, que é pensada a partir do baixo mas não depende necessariamente dele. Basta saber o tipo de acorde e a inversão para codificá-la.

Codificação das durações

A codificação de ritmos binários é bastante simples. Utilizamos o padrão que determina que uma mínima equivale ao número 2, uma semínima ao número 4, uma colcheia ao número 8, e assim sucessivamente. Essa é uma representação inversa dos valores das figuras em relação a semibreve. No entanto a codificação de ritmos que envolvem quáteras e notas ligadas pode se tornar mais complicada. A figura 3 mostra um padrão rítmico que pode ser codificado em frações da seguinte maneira:

$$\frac{1}{8}, \frac{1}{8}^+, \frac{2}{3} \left[\frac{1}{8}, \frac{1}{8}^+, \frac{4}{5} \left[\frac{1}{32}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32} \right] \right]$$

Esta operação pode ser simplificada matematicamente para se obter o valor fracionário total de cada nota, porém isso torna mais difícil obter a representação rítmica original.

$$\frac{1}{8}, \frac{1}{8}^+, \frac{1}{12}, \frac{1}{12}^+, \frac{1}{60}, \frac{1}{30}, \frac{1}{60}, \frac{1}{60} \rightarrow \frac{1}{8}, \frac{5}{24}, \frac{1}{10}, \frac{1}{30}, \frac{1}{60}, \frac{1}{60}$$

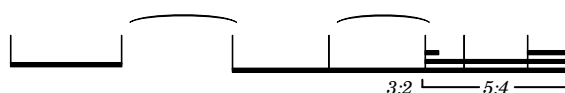


Figura 3: Padrão rítmico

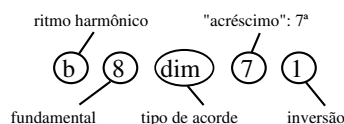


Figura 4: Codificação de acordes (harmonia)

Codificação da harmonia

Identificamos as características mais importantes dos acordes e os representamos como uma lista contendo a fundamental, o ritmo harmônico, o tipo de acorde, acréscimos, e inversões (na fig. 4 pode ser visto um exemplo para o acorde cifrado como $b^{\flat}_{5}^{6}$). Para simplificar a codificação criamos um acorde com os valores das características estabelecidas por padrão: duração de uma semínima, maior, e na primeira inversão. Para codificar a harmonia, porém, precisamos identificar as regiões cêntricas. Um dos maiores problemas na codificação da harmonia foi compreender o comportamento dos centros.

Codificação de acordes (p/ orquestra)

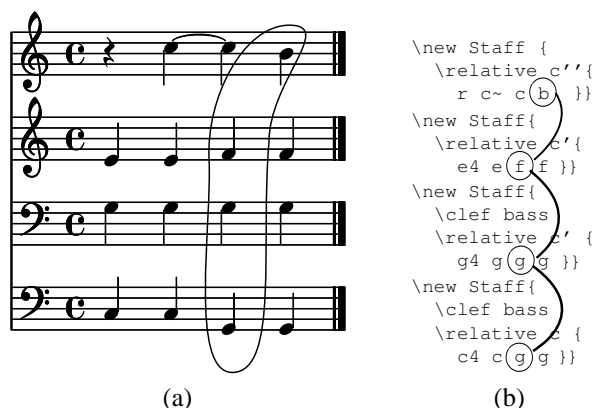
Este tipo de codificação se assemelha bastante à codificação do coral. A principal diferença é que ela contempla elementos como dinâmicas, articulações, indicações de caráter e outras informações importantes para o entendimento da orquestração. O problema específico da codificação de acordes orquestrais está relacionado aos casos de textura não homofônica.

Seleção vertical

A codificação de estruturas musicais verticais foi, até agora, o principal problema encontrado na nossa pesquisa. Apesar de tais estruturas serem escritas verticalmente, o código é escrito horizontalmente (c.f. figura 5b). Para selecionar este tipo de estrutura vertical, é necessário combinar a posição exata das suas notas em cada um dos eventos (vozes). A figura 5a traz um pequeno trecho de textura a quatro vozes com um acorde destacado (a seleção vertical), e a figura 5b, a codificação deste trecho com um destaque para a referida seleção.

Pensamos em duas formas de codificar este tipo de estrutura:

1. Localizar a estrutura no código. Esta seria uma forma em que contaríamos a posição da seleção em cada evento no próprio código. Por exemplo, no trecho exibido na figura 5a, diríamos que a seleção vertical se encontra no evento soprano, na quarta nota; e nos eventos contralto, tenor e baixo, na terceira nota.
2. Localizar a estrutura numerando a posição dentro do compasso. Esta forma indica o número do compasso, o momento onde se inicia e onde se encerra a seleção, em cada evento (soprano,



(a) Musical notation showing four staves (treble and bass clefs) with a vertical selection bracket highlighting a specific note in the second staff. (b) Corresponding MusicXML code for the same selection, with the selected note circled in the original image.

```
\new Staff {
  \relative c'' {
    r c~ c(b) }}
\new Staff{
  \relative c' {
    e4 e(f) f }}
\new Staff{
  \clef bass
  \relative c' {
    g4 g(g) g }}
\new Staff{
  \clef bass
  \relative c {
    c4 c(g)g }}
```

Figura 5: Seleção vertical em notação e código

contralto, tenor e baixo).

A desvantagem da primeira opção é a possibilidade de erro, pois, caso haja alteração em alguma voz, na duração de alguma nota anterior à seleção, todo o evento se deslocará, tornando necessária a recodificação da seleção. Já a segunda opção tem como desvantagem a prolixidade da codificação, sendo necessário indicar, voz por voz, o início e o fim da seleção.

Trabalhos relacionados

O Humdrum foi criado para facilitar a possibilidade de propor perguntas e respostas à questões relacionadas a pesquisa no campo musical (Huron 1997). Ele pode extrair trechos usando expressões regulares ou através do comando `extract`, que extrai colunas dentro da representação “tabular” do Humdrum (Huron 1999). Ele provê grande extensibilidade, contudo não de uma maneira padrão (existem mais de 40 formatos). O *Themefinder* (www.themefinder.org) é um exemplo de uso do Humdrum. Ele permite a busca por notas, intervalos, grau da escala, e contorno. Essa é uma demonstração da limitação desse tipo de codificação que lida apenas com elementos como compasso, notas, durações, etc, conforme citamos anteriormente. Apesar de novas *spines* (uma coluna com informação) poderem ser criadas é difícil criar representações em um nível mais alto de abstração.

O MusicXML foi criada para ser uma representação para notação intercambiável entre diferentes representações. Ela possui suporte para notação musical, performance, análise, e extração de dados. Contudo ela foi criada para ser “suficiente” e não “otimizada” para essas aplicações (Good 2001). O MusicXML não só não possui abstrações em níveis mais alto como explicitamente divide a representação em compassos, tornando difícil a representação de estruturas que englobam partes de compassos.

Conclusão

O trabalho de codificação de estruturas musicais com um alto grau de abstração ocupa um espaço ainda não preenchido na área de informática em música. A pesquisa tem sido feita de acordo com a metodologia *bottom-up*, que tem se mostrado adequada para o problema escolhido. A codificação é

definida usando expressões simbólicas permitindo re-definir lexicamente e sintaticamente a codificação sem maiores esforços.

Os próximos passos da pesquisa compreendem a continuação do desenvolvimento do protótipo para testes que incluirá os corais harmonizados, fugas e acordes de orquestração. A partir do protótipo poderemos expandir o trabalho com a inclusão de outras estruturas musicais como notas auxiliares à harmonia, melodias em larga escala e contornos melódicos. Este protótipo poderá servir como base para a criação de um banco de dados que armazene estruturas musicais com um alto grau de abstração em diversas obras da literatura.

Referências

- Berggren, Ulf. 1997. "Encoding of Compositional Units." In *Beyond MIDI: the handbook of musical codes*, edited by Eleanor Selfridge-Field, 451–458. Massachusetts: MIT.
- Good, Michael. 2001. "MusicXML for Notation and Analysis." In *The virtual score: representation, retrieval, restoration*, edited by Walter B. Hewlett e Eleanor Selfridge-Field, Volume 12 of *Computing in Musicology*, 113–124. Massachusetts: The MIT Press.
- Graham, Paul. 1993. *On Lisp: Advanced Techniques for Common Lisp*. Prentice Hall.
- Huron, David. 1997. "Humdrum and Kern: Selective Feature Encoding." In *Beyond MIDI: the handbook of musical codes*, edited by Eleanor Selfridge-Field, 375–401. Massachusetts: MIT.
- . 1999. *Music Research Using Humdrum: A User's Guide*. Stanford, Ca: Center for Computer Assisted Research in the Humanities.
- Krishnamurthi, Shriram. 2006. *Programming Languages: Application and Interpretation*.
- Kröger, Pedro. 2005. "Desenvolvimento e implementação de uma codificação para definir estruturas musicais." *Proceedings of the 10th Brazilian Symposium on Computer Music*. 307–310.
- Roland, Perry. 2000. "XML4MIR: Extensible Markup Language for Music Information Retrieval." *ISMIR 2000: Proceedings of the First International Conference on Music Information Retrieval*.
- Selfridge-Field, Eleanor, ed. 1997. *Beyond MIDI: the handbook of musical codes*. Massachusetts: MIT.