

Desenvolvimento e implementação de uma codificação para definir estruturas musicais

Pedro Kröger¹

¹Escola de Música da UFBA – Parque Universitário Edgard Santos, Canela
40110-150 Salvador, Bahia

kroeger@pedrokröger.net

Abstract. *There are many musical codifications for different purposes, however, to our knowledge, there is not so many research to develop a specific codification to define musical structures in a higher level of abstraction like “return all subjects in the fugues of Bach’s Well-Tempered Clavier” or “return all real answers” (which actually mean “get all answers, transpose a fifth below, compare with the respective subjects and return the ones that are real”). Our goal is to develop a specific codification for music fragments that allows a music description in higher levels beyond note/duration, including formal structures and idiosyncratic elements and to develop a prototype in Lisp to test and help the development of this codification.*

Resumo. *Existem codificações musicais para diversos fins, contudo não é de nosso conhecimento nenhuma pesquisa envolvendo uma codificação específica para definir estruturas musicais em um nível mais alto de abstração como “mostre 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”). Nosso objetivo é desenvolver uma codificação específica para fragmentos musicais que permita a descrição musical em níveis mais altos além da nota/duração—incluindo estruturas formais e elementos idiosincráticos—e desenvolver um protótipo em Lisp para testar e auxiliar o processo de desenvolvimento dessa codificação.*

1. 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 por diante). Codificações musicais têm sido usadas desde os primeiros esforços do homem para transcrever sons. Ainda que a codificação musical não esteja limitada à aplicações para computador é nessa área que encontramos o maior número de pesquisas, dadas as possibilidades de processamento oferecidas pelo computador [Selfridge-Field, 1997, p. 3–5]. Existem codificações para performance (e.g. MIDI), síntese sonora (e.g. Csound), jogos (e.g. *Music Macro Language*), notação musical (e.g. *Common Music Notation*, SCORE, GUIDO), braile, bibliografia musical (e.g. Plaine, Easie) e análise de canções folclóricas (e.g. Essen, FRELMUS). Novas codificações como SDML e MusicXML [Good, 2001] em geral lidam apenas com elementos de notação como compasso, notas, durações, etc. Tem se dado uma grande relevância para o desenvolvimento de sistemas para retornar dados a perguntas por “melodias cantadas” [Uitdenbogerd, 2000] e a criação de ferramentas para procura em bibliotecas digitais, retornando dados como gênero, tempo, executante, compositor,

etc. [Clausen et al., 2000, Dunn, 2000, Bainbridge et al., 2003]. 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 é de nosso conhecimento nenhuma pesquisa envolvendo uma codificação específica para definir estruturas musicais em um nível mais alto de abstração. Um trabalho semelhante [Holger, 2001] usa o formato GUIDO, não tem abstração de alto-nível e seu objetivo principal é o de codificação para banco de dados, não de estruturas musicais.

Acreditamos ser importante ter um sistema que seja inteligente o suficiente para poder lidar com dados musicais em um nível mais alto de abstração como “mostre 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. Esse sistema deveria usar uma codificação específica para permitir esses recursos.

2. Objetivos

Nosso objetivo é desenvolver uma codificação específica para fragmentos musicais que permita a descrição musical em níveis mais altos além da nota/duração, incluindo estruturas formais, e elementos idiosincráticos (como a noção de sujeito/resposta em fugas, ou grupos de temas na forma sonata, ou motivos em certos tipos de composições) e desenvolver o protótipo de um programa de computador para testar e auxiliar o processo de desenvolvimento dessa codificação.

3. Justificativa

Essa pesquisa se justifica não só pela originalidade, mas porque vários problemas de representação podem ser investigados, como elementos que se sobrepõem [Roland, 2000], como obter um todo a partir de segmentos, como desenvolver codificações otimizadas para certas tarefas como acordes para o estudo de orquestração, acordes no estudo de harmonia, redução analítica em relação à versão original, redução para piano em relação à versão original.

4. Metodologia

Primeiramente será desenvolvido o núcleo básico do programa que servirá como base para a implementação das codificações. Esse núcleo será baseado em um conjunto de funções em Lisp que já implementamos para lidar com dados musicais. Para garantir que o processo da pesquisa ocorra tranquilamente, durante essa fase vamos oferecer um treinamento rápido aos bolsistas nas tecnologias usadas (Lisp e LilyPond) e elementos de metodologia científica.

Em seguida desenvolveremos as quatro etapas básicas da pesquisa. Em cada uma serão analisados os problemas de codificação de um grupo específico de problemas oriundos de grupos de exemplos musicais. Os grupos escolhidos foram as Fugas do Cravo Bem Temperado de Bach, Corais Harmonizados de Bach, diversos acordes de orquestra (para estudo de orquestração), e Sonatas para Piano de Beethoven. Os recursos para a codificação necessária serão implementados no protótipo; os exemplos serão codificados, e uma série de testes serão conduzidos para confirmar a viabilidade e eficiência da codificação. Para tanto o programa deverá ser escrito de maneira extensível, para permitir que novos recursos possam ser acrescentados facilmente, inclusive por usuários.

Em cada uma dessas quatro etapas serão produzidos resultados parciais que também serão usados nos testes das etapas seguintes, garantindo uma compatibilidade maior.

Para implementar o protótipo usaremos 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 [Graham, 1993]. Usaremos o programa de tipografia musical LilyPond¹ como motor básico para gerar partituras visuais a partir da nossa codificação.

5. 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*² 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.

References

- Bainbridge, D., Cunningham, S. J., and Downie, J. S. (2003). Analysis of queries to a wizard-of-oz mir system: Challenging assumptions about what people really want. In *ISMIR 2003: Proceedings of the Fourth International Conference on Music Information Retrieval*.
- Berggren, U. (1997). Encoding of compositional units. In Selfridge-Field, E., editor, *Beyond MIDI: the handbook of musical codes*, pages 451–458. The MIT Press, Massachusetts.
- Clausen, M., Engelbrecht, R., Meyer, D., and Schmitz, J. (2000). Proms: A web-based tool for searching in polyphonic music. In *ISMIR 2000: Proceedings of the First International Conference on Music Information Retrieval*.
- Dunn, J. (2000). Beyond variations: Creating a digital music library. In *ISMIR 2000: Proceedings of the First International Conference on Music Information Retrieval*.
- Good, M. (2001). Musicxml for notation and analysis. In Hewlett, W. B. and Selfridge-Field, E., editors, *The virtual score: representation, retrieval, restoration*, volume 12 of *Computing in Musicology*, pages 113–124. The MIT Press, Massachusetts.
- Graham, P. (1993). *On Lisp: Advanced Techniques for Common Lisp*. Prentice Hall.

¹Cujo desenvolvimento nós fazemos parte (lilypond.org/web/about/thanks).

²<http://www.themefinder.org>

- Holger, H. (2001). Guido/mir—an experimental musical information retrieval system based on guido music notation. In *ISMIR 2001: Proceedings of the Second International Conference on Music Information Retrieval*.
- Huron, D. (1997). Humdrum and kern: Selective feature encoding. In Selfridge-Field, E., editor, *Beyond MIDI: the handbook of musical codes*, pages 375–401. The MIT Press, Massachusetts.
- Huron, D. (1999). *Music Research Using Humdrum: A User's Guide*. Center for Computer Assisted Research in the Humanities, Stanford, California.
- Roland, P. (2000). Xml4mir: Extensible markup language for music information retrieval. In *ISMIR 2000: Proceedings of the First International Conference on Music Information Retrieval*.
- Selfridge-Field, E., editor (1997). *Beyond MIDI: the handbook of musical codes*. The MIT Press, Massachusetts.
- Uitdenbogerd, A. (2000). Music ir: Past, present, and future. In *ISMIR 2000: Proceedings of the First International Conference on Music Information Retrieval*.