# Abstract Representation of Music: A Framework for Interoperable Research Tools

December 16, 2018

The wholesale efficacy of computer-based music research is contingent on the sharing and reuse of data, analyses and processing tools between research projects. However, computer systems for the analysis and manipulation of musical data are, in general, not interoperable.

At the technical level, there exist numerous low-level encoding formats for music, each catering to a particular kind of music, notational convention or software tool. In many cases, these encodings implement a shared set of musical concepts, such as pitch and time. However, tools designed to process one format cannot be applied to another, even when the information content is suitable [6].

At the conceptual level, ontologies are used as schemas for semantic representations of musical data and analyses. These models tend to focus on particular application domains, such as meta-data for recorded music collections [8], or audio feature extraction [1]. In many cases, these models contain a mixture of music-generic and specialised domain-specific concepts which can make difficult the large scale integration of knowledge from disparate domains [4].

Technical interoperability can be achieved through properly planned data abstraction. In computer science, abstract data types [ADTs: 3] are used to hide irrelevant implementation details behind simple interfaces which make explicit the concepts and functionality relevant for a specific task. Different concrete encodings can be rendered equivalent when viewed through the same abstract interface.

Conceptual interoperability can be enforced through a top-down approach to conceptual modelling. In knowledge engineering, upper ontologies are used to prescribe a common ontology architecture for connecting, and guiding the development of, interoperable (non-overlapping) domain ontologies [2].

Interfaces at appropriate levels of abstraction combined with high-level, overarching ontologies are required for the effective linking of research projects in collaborative music scholarship [7]. In this paper, we present the Constituent Structure Model, a general purpose music representation system which builds upon previous research on abstract representation [10, 5, 9]. The purpose of the model is to provide researchers with a common conceptual framework for linking research domains, and an abstract interface built from ADTs which supports multiple concrete data formats.

A *constituent structure* is an abstract, multiple-hierarchical data structure for representing music and music-related information in a uniform way. Nodes in the hierarchical structure are called *constituents* [10] and are used to represent diverse kinds of musical objects. Constituents can be fitted with attributes which capture the inherent characteristics of musical objects. The types of attribute values are user defined ADTs; the model is parametric in the set of abstract types required for a given application. The hierarchical relationships between constituents represent the compositional containment of musical objects; a constituent is composed of a (possibly empty) set of other constituents called its *particles*.

The Constituent Structure Model is entirely music-generic, subsuming many existing approaches to representation. It can be seen as a meta-conceptual framework, into which domain-specific representations can exist side by side. The different representational requirements of application domains are accommodated through the introduction of new ADT specifications to specialise the representation. The use of ADTs offers two main advantages: firstly, processing tools can be applied to any concrete encoding format by plugging-in the appropriate implementation. Secondly, ADT specifications provide formal documentation of the specific mathematical and computation properties of the representation of musical objects. Making this theory explicit affords greater transparency and interoperability between information systems by bridging the gap between ontology and programming.

We give an implementation of the Constituent Structure Model in JavaScript. The components of constituent structures are implemented as JSON objects whose interface operations include insert and lookup of constituents, getting and setting of attributes, and extraction of a constituents' particles. ADT interfaces are implemented as JSON objects which can be used in a modular way to extend the model's interface. Implementations of ADTs can be plugged-in to produce executable JavaScript functions. We include JavaScript modules containing ADTs for pitch and time as per Wiggins et al. [10], as well as standard numerical and string implementations of both. The purpose of the library is to aid in the development of interoperable software tools for computational musicology. Tools implemented using constituent structure interface are agnostic of the underlying encoding format and work on any kind of data so long as the requisite implementation is given. This allows the the user to focus on data modelling and application development for his or her intended problem.

We present a demonstrator application which uses our JavaScript implementation to perform simple analysis of music data. We demonstrates how the implementation supports generic processing of musical documents by applying the same analysis algorithm to two different underlying encodings of the same musical material. We demonstrate how the Constituent Structure Model supports the representation of both low-level input data and high-level structural analyses in the in a uniform way. Representation of the output of analysis methods is something which is often neglected by music representations systems, and is a significant advantage of the proposed model.

We conclude with a discussion of the advantages of abstract representation and how the Constituent Structure Model not only affords greater interoperability between systems for digital musicology, but provides a strong basis for a general purpose music knowledge representation system. We briefly mention three areas in which work on this project is proceeding. First, we mention work explores extensions to the model which accommodate addition representational features including musical surfaces, non-hierarchical constituent relations such as similarity, workflow capture, and extrinsic properties of constituents. Second, we mention work carried out to deploy the Constituent Structure Model on the semantic web using a number of OWL ontologies. Third, we mention work proceeding in defining a logic-based language for expressing sophisticated structural properties of constituent structures which supports automated inference.

The presentation needs of this submission are only a projector with standard laptop connections.

## References

[1] Allik, A. and M. Sandler
2016. An Ontology for Audio Features. In *Ismir 2016.*

[2] Arp, R., B. Smith, and A. D. Spear
2015. *Building Ontologies With Basic Formal Ontology.*

[3] Dale, N. and H. M. Walker

1996. *Abstract Data Types: Specifications, Implementations, and Applications*. Jones & Bartlett Learning.

[4] Fields, B., K. Page, D. De Roure, and T. Crawford
2011. The segment ontology: Bridging music-generic and domain-specific. In *Proceedings - IEEE International Conference on Multimedia and Expo*, Pp. 1–6.

[5] Harris, M., A. Smaill, and G. Wiggins
1991. Representing Music Symbolically. In *IX Colloquio di Informatica Musicale*, Pp. 55–69.

[6] Lewis, D., R. Woodley, T. Crawford, J. Forth, C. Rhodes, and G. Wiggins
2011. Tools for Music Scholarship and their Interactions: A Case Study. In *Supporting Digital Humanities*, Copenhagen, Denmark.

[7] Nurmikko-Fuller, T. and K. R. Page
2016. A linked research network that is Transforming Musicology. In *3rd Australasian Association for Digital Humanities (aaDH) Conference*, Hobart, Tasmania.

[8] Raimond, Y., S. Abdallah, M. Sandler, and F. Giasson
2007. The Music Ontology. In *ISMIR 2007: 8th International Conference on Music Information Retrieval*, volume 8, Pp. 417–422.

[9] Smaill, A., G. A. Wiggins, and M. Harris
1993. Hierarchical Music Representation for Composition and Analysis. *Journal of Computing and the Humanities*, 27:7–17.

[10] Wiggins, G. A., M. Harris, and A. Smaill
1989. Representing Music for Analysis and Composition. In *Proceedings of*

*the Second Workshop on AI and Music*, M. Balaban, K. Ebcioglu, O. Laske, C. Lischka, and L. Soriso, eds., number 504, P. 10. Menlo Park, CA: AAAI.