AN ONTOLOGY FOR ABSTRACT, HIERARCHICAL MUSIC REPRESENTATION

Nicholas Harley
Queen Mary University, London
n.harley@qmul.ac.uk

Geraint A. Wiggins
Queen Mary University, London
geraint.wiggins@qmul.ac.uk

ABSTRACT

This work presents an overview of the CHARM Ontology, an OWL ontology for describing abstract, multiple-hierarchical musical structures on the Semantic Web. The ontology is motivated by the desire to promote interoperability between applications that perform search and analysis of musical material, and it provides an anchor point for more domain-specific knowledge representation. An overview is given of the principle classes and relations in the ontology, as well as the way that it can be used to develop entirely interoperable tools for musical analysis. The demonstrator application performs analysis of linked data documents structured by the ontology and displays the results graphically. It is shown how a single analysis program is able operate on multiple different encodings of the same musical material and obtain the same results.

1. INTRODUCTION

Many music research tasks involve search and analysis of large music corpora and this often requires building structured digital resources from low level encodings. However, high-level musical concepts (such as melody, cadence, etc.) which are common throughout these encodings, are only implicit from the data and not directly represented. Fully utilising computers for meaningful search and analysis of these corpora requires that these high-level concepts be formalised and made explicitly available. The fields of knowledge representation and Semantic Web [2,4] have been widely used in attaining this goal, however the multiplicity of non-interoperable music encoding formats still largely prohibits the widespread reuse of data and applications.

The CHARM Ontology, presented here, is based on the CHARM specification [6], which was intended to be a general purpose representation system for capable of capturing these kinds of high-level concept at an abstract level. This work aims to characterise the original specification in a Semantic Web context and demonstrate the advantages of abstract representation.

2. CHARM

CHARM (Common Hierarchical Abstract Representation of Music) [3, 5, 6] is a specification for a flexible music knowledge representation language built from Abstract Data Types (ADTs) [1]. This approach aims to decouple the specific encoding format from the mathematical properties required of it. Abstract representation of this kind allows processing applications to operate on musical data independent of the underlying encoding format.

At the abstract level CHARM consists of basic entities described by ADTs (Pitch, Time, Duration, for example), as well as a mechanism for creating groupings of entities in multiple-hierarchical structures. Each grouping can be fitted with formal and textual definitions of the structure it denotes and all entities are given a unique identifier over which strict equality is defined. At the implementation level, functions must be provided which satisfy the algebraic properties of the ADTs. Sophisticated manipulation and creation of musical structures can be achieved using a relatively small set of atomic functions defined for the ADTs Pitch and Time [5]. An example of a function definition for Pitch is given in Eqn (1).

$$interval : Pitch \times Pitch \rightarrow PitchInterval \qquad (1)$$

Whilst primarily intended for representing sonic events with pitch and time, the CHARM data structure lends itself just as well to other hierarchical arrangements, such as audio segments or score notation. It is this flexible data structure that the CHARM Ontology seeks to capture.

3. THE CHARM ONTOLOGY

The core of the CHARM Ontology is shown in (Figure 1). The *Constituent* class encompasses the distinct entities of the representation which are nodes in the hierarchical structure. The object property *hasPart* connects a *Constituent* to its sub-Constituents (or particles), with inverse defined as *isPartOf*. The *Attribute* class encompasses the ADTs of the system. Specific ADTs such as *Pitch*, *Time* and *Duration* are defined as subclasses of *Attribute* while the object property *hasAttribute* connects a *Constituent* to an *Attribute*. Two more properties are defined for *Attribute*: *hasValue* connects to a literal containing the concrete value, and *cdt* connects to a definition of the concrete implementation of the data type (e.g. MidiPitch: Integer number of semitones).

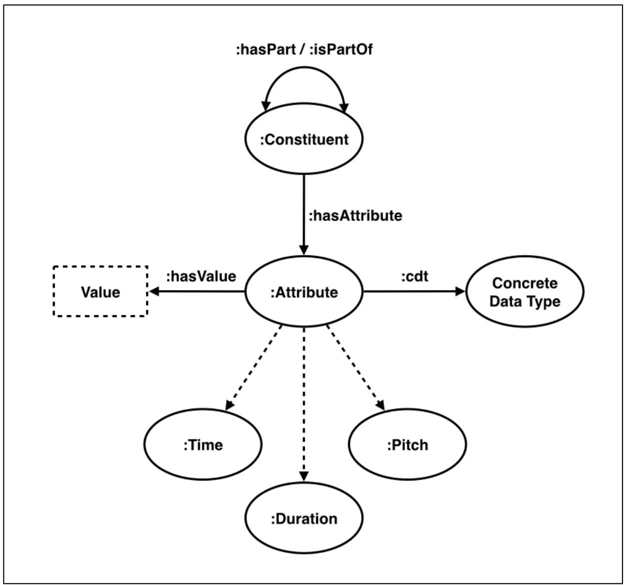

Figure 1. The core classes and relations of the CHARM Ontology. Solid arrows correspond to object and data-type properties while dotted arrows correspond to subclass relations

This basic architecture provides the fundamental components needed to realise the CHARM data structure. By extending the Attribute and Constituent classes, more domain specific ontologies can be created. Currently, additional knowledge about the intrinsic and extrinsic properties of Constituents can be provided by linking to external recourses, however work is proceeding to extend the CHARM Ontology with classes and logical axioms that permit more sophisticated automated reasoning. A brief mention of future work is provided in section 5.

4. DEMONSTRATOR

The demonstrator is a simple `Node.js` web application that retrieves and processes linked data documents. The application includes an implementation of CHARM which operates on JSON data, in which the keys correspond to the classes and relations of the CHARM Ontology. Compatible JSON objects can be made using the JSON-LD API by suppling a `@context` object, which maps JSON keys to URIs and vice-versa. The CHARM module acts as an interface between music processing tools and the data itself. An additional module performs analysis and visualisation of data via the CHARM interface. Whenever an analysis program attempts to perform an operation on an ADT, the CHARM interface checks the *cdt* property which stores the name of a module containing the concrete implementation of the *Attribute*. The module can then be required and used to carry out the operation. (The implementation is such that a concrete implementations can be passed in when a CHARM object is created. This avoids unnecessary checking should the concrete data type be known beforehand.)

To demonstrate the advantages of this kind abstract representation an example from the literature is partially recreated [5]. A common analysis procedure is performed on a simple musical example, a MIDI score of Debussy's Syrinx, and the results are visualised. With the methods described above, the same analysis program is able to operate on the on two different representations of the music. The first representation encodes pitches as integers and the second as strings (e.g. 'c#5'). The results are identical as each of the concrete implementations satisfies the pitch contract defined by the ADT in the CHARM interface.

The demonstrator runs locally on a laptop and has no special requirements.

5. FUTURE WORK

Further work is currently underway to extend to the ontology so as to provide a mechanism for defining Attributes and their required properties. Abstract specification of Attributes within the ontology will give a signature to the ADT and will constitute a blueprint for any concrete implementation of CHARM. A key goal in developing this mechanism is that it be extensible, enabling more complex type definitions to be derived from simple ones. In this way, the intrinsic properties of constituents can be abstractly defined in a logical language compliant with the original CHARM specification.

6. REFERENCES

[1] Nell Dale and Henry M. Walker. *Abstract Data Types: Specifications, Implementations, and Applications*. Jones & Bartlett Learning, 1996.

[2] Ben Fields, Kevin Page, David De Roure, and Tim Crawford. The segment ontology: Bridging music-generic and domain-specific. In *Proceedings - IEEE International Conference on Multimedia and Expo*, pages 1–6, 2011.

[3] Mitch Harris, Alan Smaill, and Geraint A. Wiggins. Representing Music Symbolically. In A. Camurri and C. Canepa, editors, *IX Colloquio di Informatica Musicale*, pages 1–20. Genova, Italy: Universita di Genova, 1991.

[4] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The Music Ontology. In *ISMIR 2007: 8th International Conference on Music Information Retrieval*, volume 8, pages 417–422, 2007.

[5] Alan Smaill, Geraint A. Wiggins, and Mitch Harris. Hierarchical Music Representation for Composition and Analysis. *Journal of Computing and the Humanities*, 27:7–17, 1993.

[6] Geraint A. Wiggins, Mitch Harris, and Alan Smaill. Representing Music for Analysis and Composition. In M. Balaban, K. Ebcioglu, O. Laske, C. Lischka, and L. Soriso, editors, *Proceedings of the Second Workshop on AI and Music*, number 504, page 10. AAAI, Menlo Park, CA, 1989.