AN ECOSYSTEM FOR TRANSPARENT MUSIC SIMILARITY IN AN OPEN WORLD

Kurt Jacobson
Centre for Digital Music
Queen Mary
University of London, UK
kurtjx@gmail.com

Yves Raimond
BBC
London, UK
yves.raimond@bbc.co.uk

Mark Sandler
Centre for Digital Music
Queen Mary
University of London, UK

ABSTRACT

There exist many methods for deriving music similarity associations and additional variations are likely to be seen in the future. In this work we introduce the Similarity Ontology for describing associations between items. Using a combination of RDF/OWL and N3, our ontology allows for transparency and provenance tracking in a distributed and open system. We describe a similarity ecosystem where agents assert and aggregate similarity statements on the Web of Data allowing a client application to make queries for recommendation, playlisting, or other tasks. In this ecosystem any number of similarity derivation methods can exist side-by-side, specifying similarity relationships as well as the processes used to derive these statements. The data consumer can then select which similarity statements to trust based on knowledge of the similarity derivation processes or a list of trusted assertion agents.

1. INTRODUCTION

The process of music recommendation in a general sense involves drawing associations between music-related items - i.e. artist a is *similar* to artist b so recommend b if the user expresses interest in artist a. We believe that similarity is the underlying "currency" for recommendation. This realization drives our interest in developing a formal model for similarity.

Similarity is a difficult concept. The exact nature of similarity has been discussed extensively in cognition [26, 28], philosophy [22, 14], and computer science [27, 17]. In the field of music information retrieval we have been less concerned with the nature of similarity and more concerned with finding ways of calculating it [18, 20, 5]. This pragmatic approach has led to a wealth of methods for deriving music similarity statements from audio analysis and contextual metadata.

But if we want to develop a generalized model for music similarity, it becomes more complicated. As Wittgenstein puts it in his seminal work *Philosophical Investigations* "Some things share a complicated network of similarities overlapping and criss-crossing: sometimes overall similarities, sometimes similarities of detail." Music would definitely be such a thing. Discussing a pair of songs, we can have a dizzying array of similarity options: the audio could have timbral similarity, rhythmic similarity, or melodic similarity; the contexts of the songs could make them similar in terms of lyrical content, cultural meaning, or shared listenership; or an authoritative source such as a music critic or website could judge the songs to be similar without providing any additional justification. Further complicating matters, similarity is *subjective* - what one individual or agent considers similar another may not.

Because similarity can be so nebulous and contentious we purpose a model for expressing similarity that foregoes hierarchical classifications and instead focuses on provenance and transparency. Instead of focusing on how a particular similarity statement is related to another similarity statement, we focus on *who* made the similarity statement and *why*.

Our approach is based on the Resource Description Framework (RDF) [4, 9] and the Web Ontology Language [3]. While these technologies provide an impressive amount of expressiveness and form the foundation of the Semantic Web, we augment their expressiveness with N3 [7]. The facilities for quoting formulae provided by N3 allows us to use the N3-Tr framework [23] for defining similarity derivation workflows.

In Section 2 we develop our model in the form of a Web ontology, briefly discussing some of the supporting technologies and previous work. In Section 3 we describe our vision of a similarity ecosystem where a number of agents aggregate and publish similarity statements in the Web of Data while music applications query these statements for recommendation or playlist generation. In Section 4 we provide a cursory evaluation of our ontology. In Section 5 we review some related work and finally provide some conclusions and directions for future work in Section 6.

2. AN ONTOLOGY FOR SIMILARITY

Because of its decentralized nature, wide deployment base, and robust technological underpinnings we use the RDF/OWL framework [4, 3, 9] for defining our Similarity Ontology.

This allows us to use the concepts, practices, and resources of *Linked Data* [8]. In the Linked Data paradigm, every resource and concept is given a Unique Resource Identifier (URI). These URIs can be dereferenced using HTTP to provide additional information and links to other relevant URIs.

2.1 Previous Ontologies

RDF [4] allows us to express information in the form of *triples*: subject, predicate, object statements. Generally the subject will be an instance of a *class concept* while the predicate will be an instance of a *property*. The object will also be an instance of a class concept but not necessarily the same class as the subject. Classes and properties are defined in an ontology document using the Web Ontology Language (OWL) [3] or the RDF Schema (RDFS) [9] or a combination of both. These technologies together enable what is commonly referred to as the Semantic Web or Web of Data.

These concepts have been successfully applied to the domain of music with the Music Ontology [24, 23]. The Music Ontology allows us to express a wide variety of music-related information as structured data in a decentralized fashion. It has been adopted by the Linked Data community and is used extensively throughout the Web of Data as a means of describing tracks, artists, performances, and related data.

The Music Ontology provides a basic facility for dealing with music similarity. The `mo:similar_to` property allows one to assert a similarity relationship between two items. However, this property relation does not provide any further information - How was the similarity derived? Who derived it? How similar are the two items?

2.2 Association as a Concept

Instead of treating similarity or, to use a broader term, association as a *property*, we treat association as a *class concept*. This allows us to reify the association in order to provide additional information about it. We introduce the class `sim:Association` and a sub-class `sim:Similarity` as the key concepts in our ontology. A simple similarity example is presented in the following listing [1]:

```
:track01 a mo:Track .
:track02 a mo:Track .
:me a foaf:Person .
:mySimilarity a sim:Similarity ;
    sim:element :track01 ;
    sim:element :track02 ;
    sim:weight "0.90" ;
    foaf:maker :me .
```

We introduce the namespace `sim` to refer to our Similarity Ontology. First we define two tracks using the cor-

responding Music Ontology concept `mo:Track`. The identifiers of these tracks can give entry points to additional information in other data sets (i.e. linking to dbpedia.org [2] URIs or Musicbrainz [3] identifiers). We define `:mySimilarity` to actually make the similarity statement. The `sim:element` property is used to refer to the tracks involved in this similarity and the `foaf:maker` property refers to the agent which asserted this similarity. Also note we can assign a numerical weight value to the similarity using the `sim:weight` property.

Now we have a method for asserting a similarity statement and reifying that statement to some extent. However, in the above example we only know *who* is making the similarity statement, we do not know *how* or *why*.

2.3 Provenance and Transparency

We introduce the `sim:AssocationMethod` concept to identify the process used to derive a similarity statement. This enables some interesting functionality when consuming the associations data - a consumer application can elect to include only similarity statements that are tied to a particular `sim:AssociationMethod`. This is discussed further in section 3.1. For now let us consider the following N3 listing:

```
:timbreSimilarityStatement
  a sim:Similarity ;
  sim:element :track01 ;
  sim:element :track02 ;
  sim:weight "0.9" ;
  sim:method :timbreBasedSimilarity .

:timbreBasedSimilarity
  a sim:AssociationMethod ;
  foaf:maker :me ;
  sim:description :algorithm .

:algorithm = {
{ { ?signal1 mo:published_as ?track01 .
    ?signal1 sig:mfcc ?mfcc1 .
    ?mfcc1 sig:gaussian ?model1 }
      ctr:cc
  { ?signal2 mo:published_as ?track02 .
    ?signal2 sig:mfcc ?mfcc2 .
    ?mfcc2 sig:gaussian ?model2 } .
  (?model1 ?model2) sig:emd ?div .
  ?div math:lessThan 0.2 } =>
  { _:timbreSimilarityStatement
      a sim:Similarity ;
      sim:element ?track01 ;
      sim:element ?track02 }
}
```

Here `:timbreBasedSimilarity` is the entity that describes our process for deriving similarity statements. Note that this entity is only described by three triples - its class type, a property for the creator and the description.

N3 extends the semantics and syntax of RDF in a useful and intuitive way. It allows for the existence of RDF graphs (a set of triple statements) as quoted formulæ. We can then make statements about the entire RDF graph providing metadata about that graph. In this way N3 is similar to Named Graphs [10], the main difference being that N3 considers RDF graphs as literals (their identity is their value), whereas Named Graphs consider graphs as entities named by a web identifier.

---

[1] We use N3 [6] in all our code listings. Each block corresponds to a set of statements (subject, predicate, object) about one subject. Web identifiers are either between angle brackets or in a prefix:name notation (with the namespaces defined at the end of the paper). Universally quantified variables start with `?`. Existentially quantified variables start with `_:`. Curly brackets denote a literal resource corresponding to a particular RDF graph. The keyword `a` correspond to the identifier `rdf:type`. The keyword `=>` correspond to the identifier `log:implies`.

[2] http://dbpedia.org
[3] http://musicbrainz.org/

In the above example, when we follow the `sim:descripti` property we see an RDF graph `:algorithm` denoted by the { and } characters. This RDF graph provides a disclosure of the algorithm used in the similarity derivation process. In this case, MFCCs are extracted and Gaussian mixture models are created concurrently for the two signals, and an earth mover's distance is calculated between models. Depending on that distance, we output a similarity statement. If more details are needed about a particular computational step, e.g. if we want to gather more information about the MFCC extraction step, we can look-up the corresponding web identifier, in this case `sig:mfcc`.

The algorithm is specified using the N3-Tr framework which uses transaction logic and N3 to describe signal processing workflows. Additional details on N3-Tr are available in [23].

Here, the N3-Tr formulæ describe the workflow supporting the similarity statement. We could forego the use of the `sim:AssociationMethod` concept and use the `log:supports` built-in predicate [4] in the N3 framework. However, as we will discuss in section 3.1, binding similarity workflows to the `sim:AssociationMethod` concept allows us to make simple, useful queries (i.e. "show me all similarity derivation methods available in the system").

Finally, note that we bind the `foaf:maker` property to the association method rather than directly to the association itself. As in the above example we can make our association method transparent, or we can provide a minimum amount of information when dealing with a "black box" similarity derivation processes. In either case it is a matter of best practice to create an association method, even if we do not desire full transparency because this allows data consumers to make simple queries.

As indicated in Figure 2.3, our framework also supports the grounding of similarity statements directly through the property `sim:grounding`. This property associates a similarity statement with the instantiated N3-Tr formulæ which enabled its derivation. In the above example, we would link our timbre similarity statement directly to a specific workflow with references to the calculated values at each step.

3. A SIMILARITY ECOSYSTEM

The data model provided by the Similarity Ontology allows for lots of flexibility in specifying similarity statements. This flexibility is balanced by the built-in mechanisms for provenance tracking. By following the `method` property in a similarity statement we know *who* made the statement and *why*. When consuming similarity data, we select statements by deciding which agents and algorithms to trust. While it is entirely possible to make a similarity statement within this framework completely anonymously, such statements are likely to be ignored by data consumers. Instead the statements from trusted agents or transparent algorithmic processes are likely to be selected by data con-
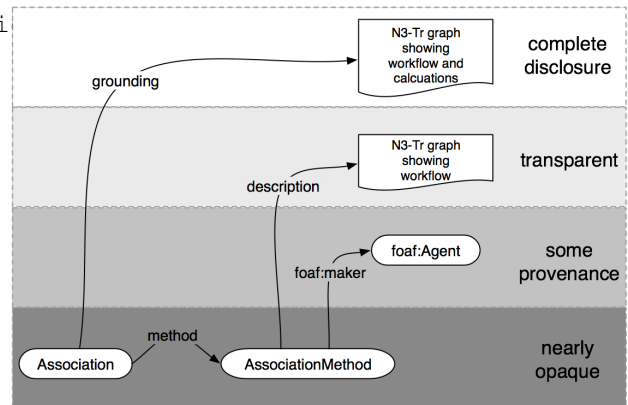
Figure 1. Using the Similarity Ontology. As additional properties are bound to our association and association method statements, we achieve greater transparency.

sumers. In a music recommendation application, this allows for more transparent recommendations - providing the end user with the source or process used to make the recommendation. Intuition as well as recommender system research suggest users are more likely to trust transparent recommendation processes [11].

Beyond the specification of the Similarity Ontology, we envision a broader ecosystem where autonomous, semi-autonomous, and human agents operate in parallel, making similarity statements about music tracks and artists while providing provenance and justification for these statements. A simple diagram illustrating how this ecosystem might be structured is provided in Figure 2.

An enabled client music application publishes the end user's listening habits to the Web of Data. Similarity agents operate on the Web of Data and publish their own music similarity statements - perhaps consuming the listening habits of end users as well as other data. These statements refer to specific URIs for each track and artist. Similarly, the client music application links the content in the user's personal collection to URIs using methods such as those detailed in [25]. This avoids ambiguity - we can be sure that the similarity statements are referring to the specific resource in which we are interested. The similarity statements made by various agents are aggregated into one or more data stores for querying. The client music application, perhaps responding to a user request, can query the data store for similarity statements from trusted agents involving the target resource (i.e a track or artist). The query returns similarity information that can be used for content recommendations or playlist generation.

3.1 Similarity Queries

Queries in this similarity ecosystem would be made using the SPARQL query language [1]. The SPARQL specification is a W3C recommendation and the preferred method for querying RDF graphs. As mentioned before, the design of the Similarity Ontology allows for the construction of simple queries to retrieve similarity information. The following query retrieves artists similar to a target artist as
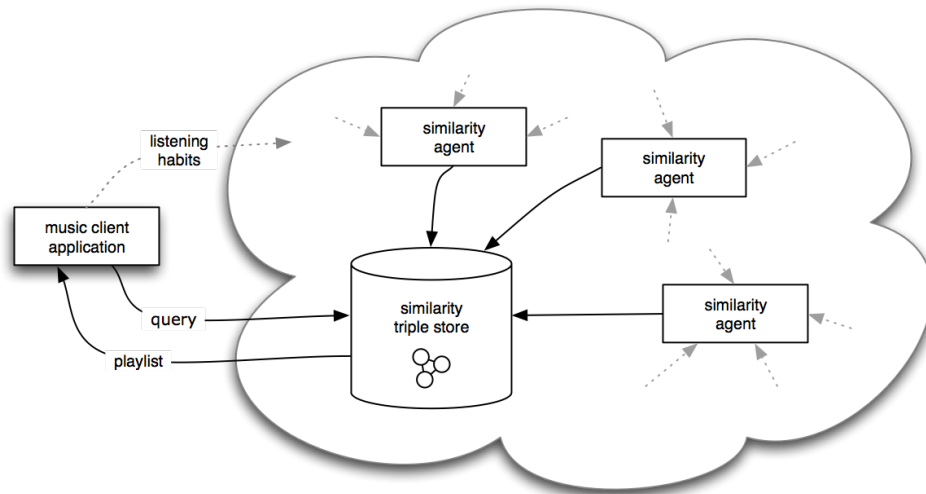
Figure 2. The music similarity ecosystem. Similarity agents operate on structured data to create similarity statements. Such statements are aggregated in a data store and queried by a client music application to provide recommendations, playlists, and other functionality.

stated by a specific trusted method:

```
SELECT ?artists WHERE {
  ?statement sim:method <http://trusted.method/uri> .
  ?statement sim:element <http://target.artist/uri> .
  ?statement sim:element ?artists . }
```

Notice we only have to include a triple pattern for our target resource, a triple pattern for our trusted agent, and a triple pattern to select the similar artists. Of course this is a very simple example and in real-world applications we include additional optional patterns and conjunctions for a more expressive query.

In an initialization step, an application could query available data sources to determine exactly what association methods and asserting agents are available. The application would use the following query:

```
SELECT DISTINCT ?method WHERE{
  ?method a sim:AssociationMethod . }
```

The application could then filter through the results and, perhaps with some input from the end-user, decide which similarity agents to trust.

3.2 Similarity and Recommendation

While we hold that similarity is the basis of recommendation, we also acknowledge that similarity and recommendation are not identical. By no means does the ecosystem proposed here solve the problems of recommender systems - rather it provides a new distributed cross-domain platform on which future recommender systems might be built.

While an item-to-item recommendation system fits quite naturally into this similarity ecosystem, we can also imagine a collaborative filtering-style user-item recommendation system. Each user in the system is treated as an `sim:AssociationMethod` instance. Each user's method makes a set of statements asserting that the tracks

found in that user's personal collection are similar to each other. Then an additional `sim:AssocationMethod` instance is used to match users with each other based on the contents of their respective music libraries. Finally, for a given user, the recommendations for that user are an aggregation of the similarity statements derived from the association methods bound to the most similar users.

Also note that the similarity ecosystem fosters hybrid recommendation approaches. Because the similarity statements are made using common semantics and syntax, we can easily combine and compare these statements to derive recommendations or new similarity statements.

4. ONTOLOGY EVALUATION

While our Similarity Ontology is very flexible and potentially very expressive, there is one import limit to its expressiveness - there is no mechanism for expressing *dissimilarity*. This is an intentional design decision that follows from the open world assumption - we cannot know *all* instantiations of similarity, and what we consider dissimilar, another agent may consider similar.

As a cursory evaluation of our Similarity Ontology we present several real-world similarity scenarios and show how our ontology can accommodate these examples.

4.1 Directed Similarity

As often noted in psychology and cognition [28], similarity is not always symmetric. For example in the domain of music we may wish to express an influence relationship or we may simply have a similarity derivation algorithm that is non-symmetric. This leads to a *directed* similarity relationship. To accommodate such scenarios we introduce `sim:subject` and `sim:object` as sub-properties of the `sim:element` property. This allows us to specify a directed similarity statement where the subject is similar

to the object, accepting that the reverse is not necessarily true.

4.2 Contextual Similarity

Because music is a complex construct deeply ingrained in culture and society, we often want to make music similarity statements that relate to the context of musical works rather than the content of the musical works themselves. Let us consider an example from popular rap music. In the mid to late 1980s a series of songs were released disputing the place of origin of the musical genre hip hop launching a multi-faceted feud that became colloquially referred to as *The Bridge Wars*[5] . By simply creating an association method that asserts similarities between artists and tracks related to this feud we can accommodate this scenario.

4.3 Personal Associations

The emotional affect of music can be highly personal. A set of associations between music artists or tracks might be unique for one particular individual. Consider the following statement, "When a first year student at college, I dated a girl who listened to Bob Marley and David Bowie" - while this association between David Bowie and Bob Marley might hold weight for the narrator, it is likely that few other individuals would share this association. However, the narrator, for any number of reasons, may wish to express this association anyway. This is entirely possible in our ontological framework. The narrator can simply create an `sim:AssociationMethod` that asserts similarity statements based on the musical taste of his ex-girlfriend.

5. RELATED WORK

Semantic Web technologies have been applied to music recommendation in previous works [12, 21] although, to the best knowledge of the authors, the present work is the first effort to develop a comprehensive framework for expressing music similarity on the Web of Data.

The Sim-Dl framework provides a basis for deriving similarities from semantic information within a description logic paradigm, although no formal syntax for expressing similarity results is provided [15]. Similarly, the iSPARQL framework extends SPARQL to include customized similarity functions [16] but fails to provide a formal method of expressing the resulting similarities.

Although the N3-Tr framework provides a clean and extensible syntax for describing similarity derivation workflows, alternative frameworks can be used as well. The Proof Markup Language provides a flexible means for justifying the results of a Semantic Web query [13].

The vast body of work on music similarity and music recommendation [18, 5, 20, 11] provides a set of templates for designing music similarity agents that might operate in our purposed ecosystem.

Knowledge management systems for music-related data such as Pachet's work [19] and more specifically the ontology engineering of Raimond [24, 23] and Abdallah et. al [2] provide the basis for the similarity ecosystem. Without the Music Ontology framework for describing music metadata and the technology and infrastructure provided by the Linked Data community - including Muscibrainz URIs for songs and artists and data publishing guidelines - the Similarity Ontology would be unusable.

6. CONCLUSIONS AND FUTURE WORK

We have presented an ontological framework for describing similarity statements on the Web of Data. This ontology is extremely flexible and capable of expressing a similarity between any set of resources. This expressiveness is balanced by transparency and provenance, allowing the data consumer to decide what similarity statements to trust. We have shown hows this framework could exist as the foundation for a broader music similarity ecosystem where autonomous, semi-autonomous, and human agents publish a wealth of similarity statements which are combined, consumed, and re-used based on provenance, trust, and application appropriateness.

We have suggested how similarity algorithms can be made transparent. We have adopted the N3-Tr syntax for describing similarity derivation workflows. In future work we plan to extend this syntax and the supporting ontologies to better enable the publication of similarity derivation workflows. Furthermore we hope to develop a series of recommendations for best practice when publishing such workflows to maximize their usefulness and query-ability.

We also plan to adopt a method of digitally signing similarity statements in our ecosystem using terms available in the WOT RDF vocabulary[6] . This would allow agents to sign similarity statements using Public Key Cryptography to avoid "spam" similarity statements.

While our Similarity Ontology was designed with music similarity in mind, it is by no means limited to the domain of music. As we have shown, the framework is both flexible and extensible. We leave it to future work to explore how this framework might be applied in different domains and across domains.

7. NAMESPACES

The following namespaces are used throughout this work:

```
@prefix mo: <http://purl.org/ontology/mo/>.
@prefix sim: <http://purl.org/ontology/similarity/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix sig: <http://purl.org/ontology/signal/>.
@prefix ctr: <http://purl.org/ontology/ctr/>.
```

8. REFERENCES

[1] SPARQL query language for RDF, W3C recommendation, 2008.

---

[5] `http://en.wikipedia.org/wiki/The_Bridge_Wars`

[6] `http://xmlns.com/wot/0.1/`

[2] Samer Abdallah, Yves Raimond, and Mark Sandler. An ontology-based approach to information management for music analysis systems. In *120th Audio Engineering Society Convention*, 2006.

[3] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneijder, and Lynn Andrea Stein. OWL Web Ontology Language Reference. Recommendation, World Wide Web Consortium (W3C), February10 2004. See `http://www.w3.org/TR/owl-ref/`.

[4] D. Beckett. RDF/XML Syntax Specification (Revised). Recommendation, World Wide Web Consortium (W3C), 2004. Internet: `http://www.w3.org/TR/rdf-syntax/`.

[5] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. P. W. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music J.*, 28(2):63–76, 2004.

[6] Tim Berners-Lee. Notation 3, 1998. See `http://www.w3.org/DesignIssues/Notation3.html`.

[7] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3logic: A logical framework for the world wide web. *Theory and Practice of Logic Programming*, 2007.

[8] Chris Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web.

[9] Dan Brickley and Ramanatgan V. Guha. Rdf vocabulary description language 1.0: Rdf schema. W3c recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[10] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2005. ACM.

[11] O. Celma. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[12] Òscar Celma, Miquel Ramírez, and Perfecto Herrera-Boyer. Foafing the music: A music recommendation system based on rss feeds and user preferences. Proceedings of the 6th ISMIR, 2005.

[13] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Richard Fikes. A proof markup language for semantic web services. *Inf. Syst.*, 31(4):381–395, 2006.

[14] Keith J. Holyoak. *The Cambridge Handbook of Thinking and Reasoning*. Cambridge University Press, Cambridge, UK, April 2005.

[15] Krzysztof Janowicz. Sim-dl: Towards a semantic similarity measurement theory for the description logic in geographic information retrieval. *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1681–1692, 2006.

[16] Christoph Kiefer, Abraham Bernstein, and Markus Stocker. The fundamentals of isparql - a virtual triple approach for similarity-based semantic web tasks. In Karl Aberer and Key-Sun Choi, editors, *Proceedings of the 6th International Semantic Web Conference*, LNCS, pages 295–308, Berlin, 2007. Springer.

[17] Dekang Lin. An information-theoretic definition of similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[18] B. Logan and A. Salomon. A music similarity function based on signal analysis. *Multimedia and Expo ICME*, pages 745–748, 2001.

[19] Francois Pachet. Knowledge management and musical metadata. *Encyclopedia of Knowledge Management*, 2005.

[20] E. Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrival*. PhD thesis, Technischen Universität Wien, May 2006.

[21] Alexandre Passant and Yves Raimond. Combining social music and semantic web for music-related recommender systems. In *Semantic Web Workshop*, 2008.

[22] W. V. Quine. *Ontological relativity and other essays*. Columbia University Press, New York, NY, USA, 1969.

[23] Yves Raimond. *A distributed music information system*. PhD thesis, Queen Mary University of London, 2009.

[24] Yves Raimond, Samer Abdallah, Mark Sandler, and Frédérick Giasson. The music ontology. Proceedings of the 8th ISMIR, 2007.

[25] Yves Raimond, Christopher Sutton, and Mark Sandler. Automatic interlinking of music datasets on the semantic web, 2008.

[26] Roger N. Shepard. Geometrical approximations to the structure of musical pitch. *Physchological Review*, 89:305–333, 1982.

[27] Joshua B. Tenenbaum. Learning the structure of similarity. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA, USA, 1996.

[28] Amos Tversky and Itamar Gati. Similarity, separability, and the triangle inequality. *Physchological Review*, 89:123–154, 1982.