MusicBrainz Data License Issues

by Robert Kaye (rob@musicbrainz.org)

The MusicBrainz project currently distributes its dataset using the OpenContent license. However, the Free Software Foundation frowns upon this license, and it really isn't suited for licensing factual data as it turns out. Given these issues, what license should be used for distributing the MusicBrainz dataset?

What is wrong with using the OpenContent license?

MusicBrainz uses the OpenContent license [1] to make the data in the MusicBrainz server available to the public. This license hopes to be a GPL analogue license for general content (music, pictures, stories, etc.) and is not specifically designed to handle data that resides in an SQL server.

Richard Stallman has some objections to the OpenContent license and the GNU web page states:

> "This license does not qualify as free, because there are restrictions on charging money for copies. We recommend you not use this license." [2]

If OpenContent does not solve the problems that MusicBrainz faces, then what license does? I like the spirit of the GPL, and I've considered using the GPL for MusicBrainz, but GPL uses software specific terms like *source code*, o*bject code*, and *executable* all of which do not apply to data. Richard Stallman suggested using the GNU Free Documentation License, but that license is specific to documentation and uses terms like *title pages, appendices, and cover texts*. This license feels like trying to force a square peg into a round hole, and I don't think it would provide adequate protection for a data set in a court of law.

After endless hours of research trying to find a fitting license I've come to the conclusion that none exist that can cover the issues that arise from trying to apply the Copyleft concept to a database. Simply trying to define what constitutes a *derivative work* for a database is a frustrating experience.

Consider the following issues:

- Is a derivative work created when someone takes some data from a Copyleft database, combines it with some data from a proprietary database and then displays the data to a customer? What if that customer is or is not charged for access to that database?
- It would seem that importing one column from a Copyleft database into a proprietary database would constitute a derivative work. But, what if a user inserts row identifiers from a Copyleft database into a proprietary database in order to cross reference them?

- If someone takes some data from a Copyleft database and adds it to their own web page, is their web page now subject to the same license that covers the database?

A license that protects the contents of a database should be able to address the concerns above. This may seem like a tall order already, but the situation gets more complicated. In the 1991 case of *Feist Publications, Inc. v. Rural Telephone Service Company, Inc., the US Supreme Court* ruled that a database (a compilation work) must contain a minimum level of creativity in order to qualify for copyright protection [3]. Merely listing factual data in an alphabetic listing is not enough, and anyone can legally extract portions (short of the whole compilation) of the database for their own use.

Put bluntly, **factual data cannot be copyrighted**.

To date, the data collected in MusicBrainz is 100% factual -- this will change over time as MusicBrainz starts collecting data like music reviews and genre listings. However, given the Feist decision, how can we protect the MusicBrainz data?

Detailed Problem Analysis

Instead of looking for one license that can cover the MusicBrainz data, we must now look for a license to cover the non-factual data and we must come to terms with the reality that factual data cannot be copyrighted.

Factual data cannot be copyrighted in the United States -- in the European Union and in lots of other countries of the world factual data can be copyrighted. But with the US as the weakest link in the chain to protect the data, it will be impossible to protect the data even if the MusicBrainz project were located outside of the US. Consider this scenario:

> *A citizen of a European country travels to the United States and downloads most (but not all) of the portions of the MusicBrainz data. Perhaps this person takes all of the data, except for a few obviously bad metadata entries. This person then exports the dataset by returning to the EU. Finally, said person can then take this dataset and apply a copyright notice and begin using and/or selling this data to other EU customers. While this may not be morally correct, it is perfectly legal.*

Regardless of the license that MusicBrainz applies to the data, the above is legal. Lets assume for a moment that the MusicBrainz community decides to release the data under the GPL. The GPL assumes that copyright law applies to the source code/data and proceeds to give the licensee a set of rights, as long as the licensee abides by terms of the GPL. If the licensee violates any portion of the GPL, the GPL becomes null and void, and the licensee is now in direct violation of copyright law.

The EU citizen in the example above exports the GPL dataset back to the EU and then

begins to sell copies of the data. That action is a clear violation of the GPL, which then causes the GPL to no longer apply, thus falling back to the existing copyright laws. **But in this case, there are no copyright laws to fall back upon, and thus the MusicBrainz community has no legal recourse against the immoral citizen.**

The end result is that there is no way to *protect* factual data. Period.

On the other hand, factual data cannot be copyrighted, and the MusicBrainz community should seek for or define a suitable license for the non-factual metadata. We'll cover this topic in much greater detail later.

A New Approach

If the factual data in MusicBrainz cannot be protected, do I just stop hacking and get a real job?

No. We need to examine our beliefs and our preconceived notions about having to *protect our dataset.* At the first CodeCon Conference [4] Fred von Lohmann, who is a staff attorney for the EFF, suggested that the community stop trying to *protect* the dataset and instead focus on how to ensure that the dataset will always be available to the community.

Fred's comments suggest a simple method for resolving the license issues that is not fraught with paradoxes and complex legal documents. Rather than restricting who and why someone can use the dataset, we simply need to ensure that the data has a permanent home that cannot be compromised by greed or other malicious activities.

The thought of letting go of control of the MusicBrainz data will be a new one for the MusicBrainz community. The concept itself is not new at all -- the most prominent example of letting go of control comes from Linus Torvalds. Linus, the creator of the Linux kernel which makes up the core of the GNU/Linux operating system, made the source code to his kernel available to the public in 1991, and thus relinquishing ultimate control over the code. Letting go of control over the kernel sources has spurred the development for Linux and its consequent rapid adoption in the industry.

The same will be true for MusicBrainz. Letting go of control of the data and explicitly placing the factual metadata into the Public Domain will make the metadata more useful than it is today. I am even willing to argue that we're holding back the MusicBrainz project by trying to *protect* the data.

Data has no intrinsic value. Data with relationships to other data has some value. Data with relationships to lots of other data is more valuable. In essence, the more connected a dataset is, the move valuable it becomes. Consider Metcalfe's law:

> Metcalfe's Law states that the usefulness, or utility, of a network equals the square of the number of users.

I believe that Metcalfe's law applies to data as well:

> The usefulness, or utility, of a dataset equals the square of the number of relationships contained in the dataset.

The more people/projects/companies use and link against the MusicBrainz dataset, the more useful the dataset will become. Each new relationship in the dataset and external relationships to and from the dataset will increase its value. Only by making the dataset freely available, without restrictions on its use, can the MusicBrainz dataset realize its full potential.

Threat Analysis

Lets assume that the MusicBrainz factual metadata will be explicitly released into the Public Domain. What threats exist that may make this move undesirable? Two threats become apparent:

1. The M$ threat: M$ downloads the data and creates a service that sells this data to its customers. M$ makes money off the backs of the MusicBrainz community.
2. The GN threat: GN purchases all of the MusicBrainz intellectual property, data and the souls and bodies of the creators of MusicBrainz. GN shuts down MusicBrainz and offers the same service for a fee to its customers.

These threats may seem serious at first glance, but a detailed breakdown shows that theses threats don't amount to much. In the first threat, M$ would have a hard time selling a product that is available for free to its customers. The free software community would make its voice heard (as it has in the past) and create an overwhelming negative PR nightmare for M$. The uproar from the community would make it clear that the service for which M$ had been charging is available for free at MusicBrainz. The idea of starting a service to charge for what is available for free is not a sound business practice.

M$ could also choose to add extra value to the product offering in order to charge money for the service. This concept is actually endorsed by the GPL, and thus is not really a threat. Furthermore, M$ could refuse to contribute changes back to the community. This means that they need to create the tools to update and maintain the dataset or take the MusicBrainz server source code and adapt it for their own use. Either one of these approaches costs lots of money in engineering time -- it would be cheaper to let the established MusicBrainz community take care of the maintenance of the dataset.

However, we all know that M$ has lots of money, and lets assume that they develop their own tools for data maintenance. It is unlikely that the paying M$ customer base would be willing to help maintain the dataset as the MusicBrainz community does. M$ is not known for having customers with noble goals about open communities. Thus, M$ would be forced to maintain the dataset themselves, and this translates into ongoing expenses, which are harder to justify than one time expenses. The dataset would also suffer in quality, since M$ cannot possibly have the breadth of users that MusicBrainz enjoys. M$ would be better off to have a friendly relationship with MusicBrainz and to send their customers to MusicBrainz for data maintenance.

The GN threat is even less important, if we take a few preventative steps. Fred von Lohmann suggests that we form a non-profit corporation that has the express bylaws that state that the MusicBrainz dataset must always be made available to the public. Furthermore, a US non-profit corporation cannot be bought by another corporation; in order to dissolve a non-profit corporation, its assets must be donated to another non-profit corporation. This will ensure that the assets of MusicBrainz cannot be legally acquired and locked away from the public.

And in the case that GN acquires the souls and bodies of the creators of MusicBrainz, the other members of the MusicBrainz community will take action to isolate compromised elements of MusicBrainz. And as stated earlier -- an isolated data set is worth a lot less than a well connected dataset.

In order for MusicBrainz to *protect* itself it must take a few legal and technical steps to ensure that the availability of the data cannot be compromised and that it is easier to work with MusicBrainz than to work in isolation.

The New Solution in Detail

A comprehensive solution to the MusicBrainz data license dilemma will require separate solutions for the factual and the non-factual metadata. The following steps propose such a solution:

1. All factual metadata should be explicitly placed into the Public Domain, thereby acknowledging that factual metadata cannot be copyrighted.

2. A MusicBrainz non-profit corporation should be created, which should adopt a set of bylaws that state that MusicBrainz will make all factual metadata created by the MusicBrainz community available to anyone who wishes to download the data.
3. The non-profit corporation should establish a number of partnerships with groups like the Creative Commons or the Free Software Foundation, to mirror the MusicBrainz data. Having a neutral third party make the data available ensures fair access to the dataset and acts as a backup in case the MusicBrainz non-profit should cease to exist.
4. The MusicBrainz community should work to identify any issues that arise from applying the Copyleft concept to a database of non-factual metadata. Defining a *derivative work* and to what extent linking to and using non-factual metadata is allowed under the Copyleft concept are only a few of the issues that need to be addressed.
5. MusicBrainz should work with the established free software/open source license experts such as Richard Stallman and Eric Raymond to define a new license that addresses the complexities of applying the Copyleft to non-factual metadata. Creating a license that is derived from the GPL or other licenses that have had public scrutiny would be a great benefit to creating a solid license that can withstand an attack in a court of law.

Conclusion

The OpenContent license is not the right license for MusicBrainz, and as far as I can tell, no appropriate license for data has been created yet. This white paper serves as a call for participation for creating an appropriate license. If you have any thoughts on this matter, please take a moment to share them with me or the MusicBrainz community in general [5].

References:

[1] http://www.opencontent.org
[2] http://www.gnu.org/licenses/license-list.html
[3] http://www.bitlaw.com/copyright/database.html
[4] http://www.codecon.org
[5] http://musicbrainz.org/list.html