The effects of translation between symbolic music formats: A case study with Humdrum, Lilypond, MEI, and MusicXML

Néstor Nápoles López, Gabriel Vigliensoni, Ichiro Fujinaga

McGill University, CIRMMT

1 Abstract

After encoding or translation, symbolic music files are prompt to contain discrepancies that may lead to distinct interpretations in different software. In this paper, we investigate the effects of translating four collections of symbolic music files that have been distributed in different formats: Humdrum (**kern), Lilypond, Music Encoding Initiative (MEI), and MusicXML. The corpus of symbolic music files used in our experiment corresponds to four collections of music scores from the common practice period of Western music obtained from well-established projects and websites. The sources we selected have made their scores publicly available and distributed them in different symbolic music formats.

We firstly present a survey of the possible ways that files can be translated between these formats, we then apply each translation path to all the symbolic music files in our corpus, and we finalize this paper by searching for the effects and discrepancies that the translations add to the converted files. An initial analysis of our experiment shows that discrepancies were introduced—to a greater or lesser extent—in all of the translation paths that we tested. The discrepancies involve changes in the duration of the notes, articulations, and the offset of a note in the translation with respect to the same note in the original file. Additionally, several files cannot be translated or parsed after the conversion. We present a brief summary of the circumstances related to these problems.

2 Introduction

Digital music scores are handled by different software applications using a variety of symbolic music formats. End users of these applications come from different disciplines and have different backgrounds, and so it is common that they are more familiar to certain software and their native symbolic music formats (Arthur, Condit-Schultz, and Sapp 2017). As a result, the translation of symbolic music formats is a common and inevitable process when working with digital music scores.

In order to extend the compatibility of formats and reach different audiences, a good application should correctly translate between different formats automatically. Examples of such software can be found among editors (Sapp 2017), music engravers (Pugin, Zitellini, and Roland 2014), and music analysis toolkits (Cuthbert and Ariza 2010). In cases like these, the process of translation is not only inevitable but also transparent to the user. In a previous work, it has been found that during encoding or translation the generated or converted symbolic music file can have discrepancies that may affect the way they are interpreted by different software (Nápoles López, Vigliensoni, and Fujinaga 2018). In order to dig deeper into the topic of symbolic music format translation, we investigate the different paths for the translation of four symbolic music formats: Humdrum (**kern), Lilypond, MEI, and MusicXML. We present a summary of these paths in the next section.

3 Survey

The translation of symbolic music formats is usually *asymmetric*. That is, translating symbolic music format A to B can potentially lose more information than translating B to A. Even when translations could be lossless in theory, in practice, the robustness and quality of the software

translating format A to B can be different than the software translating B to A. Therefore, it is not only important to evaluate different software for doing one-directional translations, but also evaluate the inverse process. In Table 1, we summarize the most up-to-date software for translating between symbolic music formats.

	Humdrum	Lilypond	MEI	MusicXML
Humdrum		hum2ly[1]	Verovio[2]	hum2xml[3]
Lilypond	lilypond-export[4]		ly2mei[5]	python-ly[6]
MEI	mei2hum[7]	MEILER[8]		music21[9]
MusicXML	humlib/musicxml2hum[10]	musicxml2ly[11]	Verovio[12]	

Table 1: Software used to translate from one symbolic music format to another. The original file is indicated by the leftmost column and the translation by the topmost row.

4 Experiment

In order to study the effects that these different translations have in the content of the output files, we apply the software in Table 1 to a corpus of four different collections of symbolic music files and compare the translations to the original files. We classify the corpus of symbolic music files based on the format they are distributed in. The selection of the sources for each collection was based on free accessibility to the files. The four 100-piece collections of symbolic files are:

- *Humdrum*: A set of 100 files obtained from the *humdrum-data* repository.[13]
- *Lilypond*: A set of 100 files obtained from the *Mutopia Project*.[14] The selected files have been classified with the *Classical* style tag in the website.
- *MEI*: A set of 100 files obtained from the *Verovio* website,[15] the *Digital Interactive Mozart Edition*[16], and other MEI collections.
- *MusicXML*: A set of 100 files obtained from the *Gutenberg Project*.[17]

5 Results

In order to compare the original scores with their corresponding translation, we have followed a similar approach to the one introduced in Nápoles López, Vigliensoni, and Fujinaga (2018), where each note-rest event in the original music score is matched with a note-rest event that occurs at the same offset in the translation, when such note-rest event is found in the translation, the attributes of the original note-rest event (e.g., articulations, ornaments, and duration) are compared against the attributes of the translated note-rest event.

1. `https://github.com/craigsapp/hum2ly`.An alternative tool has been derivated from this project, libguile_humdrum, which has not been tested in this paper
2. Available translation from the Verovio command-line arguments
3. `https://github.com/craigsapp/humextra` Standalone program part of the Humdrum Extra Tools
4. `https://github.com/openlilylib`. Additionally, allows to write MusicXML files. The MusicXML writer has not been tested in this paper
5. `https://github.com/uliska/mei2ly`. Despite the name of the repository, it also allows Lilypond to MEI translations
6. A python library to manipulate Lilypond files, significant effort has been done by several contributors to translate Lilypond to MusicXML as part of this project
7. `https://github.com/craigsapp/humlib`. Standalone program included in the humlib library
8. `https://github.com/rettinghaus/MEILER`
9. A musicology toolkit that is able to translate MEI files to an internal python representation, which can be translated into MusicXML output files
10. `https://github.com/craigsapp/humlib`. Standalone program included in the humlib library
11. Included in the Frescobaldi Music Notation Editor
12. Available translation from the Verovio command-line arguments
13. `https://github.com/humdrum-tools/humdrum-data`
14. `http://www.mutopiaproject.org/`
15. `https://www.verovio.org/index.xhtml`
16. `https://mozarteum.at/en/digital-interactive-mozart-edition-nma-online/`
17. `https://www.gutenberg.org/wiki/Gutenberg:The_Sheet_Music_Project`

It is common to observe discrepancies involving changes in the duration of the notes, articulations, and the offset of a note in the translation with respect to the same note in the original file. In the worst case of our analysis, close to 30% of the notes in a translated file have a different offset than in the original file and close to 5% of them lose articulations and ornament properties.

6 Conclusion

Whether it is due to software applications using a variety of symbolic music formats or end-users coming from distinct disciplines that are more familiar to certain symbolic music formats, translation between these formats is an inevitable process. We continue to explore discrepancies introduced in symbolic music files studied previously (Nápoles López, Vigliensoni, and Fujinaga 2018), focusing particularly in those related to translation between symbolic music formats. We believe that introducing new methodologies for comparing symbolic music formats and putting them in practice with real collections is a valuable process to gain insight in the current state of symbolic music translation software, which could be useful for digital music libraries, researchers, performers, and users in general. It could also help to facilitate the integration of incoming symbolic music formats that will eventually arrive to the—already wide—ecosystem of symbolic formats for digital music scores.

References

Arthur, Claire, Nathaniel Condit-Schultz, and Craig Sapp. 2017. "MEI, Humdrum, and music21: A Comparison of Music Encoding Systems and Toolkits." In *Presentation of the Music Encoding Conference 2017.* Tours, France.

Cuthbert, Michael Scott, and Christopher Ariza. 2010. "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data." In *Proceedings of the 11th International Society for Music Information Retrieval Conference,* 637–642. ISMIR 2010. Utrecht, Netherlands.

Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. "Encoding Matters." In *Proceedings of the 5th International Conference on Digital Libraries for Musicology,* 69–73. DLfM '18. Paris, France: ACM. ISBN: 978-1-4503-6522-2. doi:10.1145/3273024.3273027. http://doi.acm.org/10.1145/3273024.3273027.

Pugin, Laurent, Rodolfo Zitellini, and Perry Roland. 2014. "Verovio: A Library for Engraving MEI Music Notation into SVG." In *Proceedings of the 11th International Society for Music Information Retrieval Conference,* 107–112. ISMIR 2014. Taipei, Taiwan.

Sapp, Craig. 2017. "Verovio Humdrum Viewer." In *Presentation of the Music Encoding Conference 2017.* Tours, France.