Initial Survey Results from The LilyPond Consortium

Mike Solomon  
Ensemble 101  
mike@ensemble101.fr

Urs Liska  
openlilylib  
ul@openlilylib.org

Trevor Baˇca  
Harvard University  
trevorbaca@gmail.com

ABSTRACT

The present poster introduces The LilyPond Consortium (lilypondconsortium.org), a newly formed international federation of companies, institutions, associations, professionals and hobbyists that use the GNU LilyPond score writer. The poster presents the broad goals and missions of the LilyPond Consortium as well as specific recommendations formulated by the consortium’s working groups regarding the evolution of LilyPond in the near future.

1. INTRODUCTION

Created in October 2014, The LilyPond Consortium is a federation of companies, institutions, and individuals using the GNU LilyPond [1] music typesetter. It is founded upon the conviction that beautiful music engraving is an evolving art that requires constant exchange around an open set of ideas. The LilyPond Consortium seeks to break down institutional, geographical, economical, cultural and educational barriers that prevent collaboration between LilyPond users, developers and educators.

The primary mission of The LilyPond Consortium is to help its members effectively collaborate on projects that use LilyPond. This collaboration, in turn, contributes to the evolution and perfection of LilyPond’s open-source code base. By advocating LilyPond in musical and academic communities and matching members’ interests with logistical, financial and human resources, The LilyPond Consortium aims to increase LilyPond’s presence in the music engraving market, which will in turn contribute to the evolution and perfection of LilyPond’s open-source code base.

In December 2014, The LilyPond Consortium conducted a survey of its membership in order to elaborate a strategic plan for collaborative development effort in 2015 and 2016. Certain members’ suggestions have sparked projects focusing on infrastructural improvements to LilyPond’s code base so that it can evolve with a rapidly changing technological landscape. Other projects significantly extend LilyPond’s functionality and usability from the perspective of our members’ requirements.

Table 1. Members of the LilyPond Consortium as of February, 2015.

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization/Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike Solomon</td>
<td>Ensemble 101</td>
</tr>
<tr>
<td>Urs Liska</td>
<td>openlilylib</td>
</tr>
<tr>
<td>Trevor Baˇca</td>
<td>Harvard University</td>
</tr>
</tbody>
</table>

Copyright: ©2015 Mike Solomon et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
also provides a limited number of travel grants for members to present their work.

2. Downloads of GNU LilyPond binaries as well as LilyPond-related software and libraries. It also offers a continuous-integration service for its internal projects via Jenkins.


3. SURVEY RESULTS

In December 2014, members of The LilyPond consortium participated in a voluntary survey (presented below in its entirety) in order to determine the consortium’s sections and projects for 2015 and 2016. This section presents salient findings from the survey. In cases where a member wished to remain anonymous, they are referred to by an uppercase letter (i.e. Member A).

To kick things off, our goal is to submit a poster for consideration to the International Computer Music Conference (http://icmc2015.unt.edu). The poster will act as a roadmap for the Consortium’s activities by defining the following two core concepts:

1) sections: themed clusters into which members can be grouped based on how they use LilyPond
2) projects: development initiatives on the LilyPond code base that are needed by members

To help define these concepts, we’ve come up with the two questions below, which I would like everyone on this list to answer (either to me personally or reply-to-all):

1) In what ways does your project use LilyPond? 2) How, if at all, can LilyPond be improved to better integrate with your project?

The answers to these questions can be as specific or general as you’d like.

3.1 Sections

3.1.1 Scholarly editing

This section represents members whose primary interest is the generation of critical editions and other forms of high-quality scores suitable for both performance and academic study. It represents multiple forms of engraving know-how codified into LilyPond syntax and workflows. As OpenLilyLib states, “The original idea behind OpenLilyLib was to make library material publicly available that was gathered during an edition project” (OpenLilyLib, personal communication, December 12, 2014). ensemble 101 reports, “we are part of an emerging trend of performer-composer-engravers for whom the generation of Urtext documents and artistic creation is increasingly inseparable. As such, the LilyPond libraries we have developed over the years likely reveal as much about typographical intent as our scores themselves” (ensemble 101, personal communication, January 3 2015). This section also includes issues of incorporating LilyPond into other platforms in which scholarly editions can be compiled. “lilyglyphs is a TikZpackage that allows to integrate notational elements into TeXdocuments by either accessing the glyphs from LilyPond’s Emmentaler font or using small PDF files generated with LilyPond (to use arbitrary notational constructs not available through the font itself)” (lilyglyphs, personal communication, January 1, 2015).

The scholarly editing section is currently comprised of Scores of Beauty, lilyglyphs, Music Notation Project, The University of Alicante, SCORA, sound-rep, ensemble 101 and OpenLilyLib.

3.1.2 Contemporary composition

Consortium members dealing with issues in contemporary musical composition are disproportionately highly represented in The LilyPond Consortium with respect to other musics for which LilyPond is used as a typesetter. OpenMusic “is a Computer Assisted Compositional environment using a Visual Programming Language. ... [OpenMusic] exports Om Score Objects into [sic] LilyPond .ly format” (OpenMusic, personal communication, December 22, 2014). abjad uses LilyPond as a backend to display scores written in Python via its libraries. Clairnote displays notation via “chromatic staff, use of solid and hollow notes for inciting pitch, vertically compressed staff, alternative key signature and accidental signs, etc.” (The Music Notation Project and Clairnote, personal communication, December 22, 2014).

The contemporary composition section is currently comprised of abjad, OuMuPo, Clairnote, OpenMusic, PWGL, ensemble 101, Music Notation Project and OpenLilyLib.

3.1.3 Graphical user interfaces

Members creating graphical user interfaces with LilyPond fall into three categories. denemo an Scorio provide graphical front-ends that are translated into LilyPond code and render to LilyPond in realtime. Frescobaldi is a “text editor to edit LilyPond files. It provides syntax highlighting and tries to understand at a very high level the contents of the LilyPond file” (Frescobaldi, personal communication, January 6, 2015). OpenMusic and PWGL provide no direct visualization of LilyPond scores, but export to the LilyPond format.

The graphical user interfaces section is currently comprised of Frescobaldi, denemo, Scorio, OpenMusic and PWGL.

3.2 Projects

3.2.1 Modularity

Modularity has surfaced as the principal concern in survey results. As summarized by Member B, “What makes it hard for newbies to understand the LilyPond source code is the lack of modularization. Although the code is pretty well-structured, it is very hard to identify components (such as Parsing, Music interpretation, Grob processing, ...) and how they interact. Another problem is source debugging due to the intermingling of the C++ with the Scheme interpreted code. It might
therefore be helpful to split the code into components that expose their public functions through interfaces” (Member B, personal communication, January 8, 2015).

Member A also expresses a need for modularity in LilyPond spacing engine, writing “it would be useful if the task of spacing (horizontal layout of notes and paging) could be separated as a stand-alone library” (Member A, personal communication, January 29, 2015). The OpenLilyLib project generalizes this concern, stating “we want to think about a general/generic way to design, use and deploy libraries to extend LilyPond’s functionality…. The overall idea is to have an easy way to distribute and use libraries for specific purposes” (OpenLilyLib, personal communication, December 12, 2014).

Perhaps the biggest need for modularity comes in parsers, expressed as needs by OpenMusic, PWGL, abjad, Frescobaldi and Member B. All of these tools have implemented custom parsers for LilyPond. As member B writes, “the internal representation is given by the LilyPond grammar and its AST representation. It is difficult for programmers writing converters from or to LilyPond format and therefore keeps LilyPond somewhat isolated from the rest of the music notation world” (Member B, personal communication, January 8, 2015).

3.2.2 Alternative notation systems

Several members of The LilyPond Consortium work with notation systems that differ in several respects from the engraving norms established by publishing houses such as Edition Peters and Bärenreiter. Member C writes, “LilyPond’s default values often favour [sic] older notation standards which aren’t suitable for contemporary works” (Member C, personal communication December 19, 2014). The Music Notation Project and Clairnote, who responded as a single entity for the survey, call for the “‘de-hardwiring’ of LilyPond from the assumptions of standard notation” in several areas, including staves with alternative line spacing and support for vertical pitch distribution systems such as the chromatic staff (The Music Notation Project and Clairnote, personal communication, December 22, 2014). In order to maintain relationships between these changes and improvements in traditional engraving, ensemble 101 suggests “a mapping structure that allows for tweaks to be linked to internal behavior so that hacks do not fall apart due to variations between versions of LilyPond” (ensemble 101, personal communication, January 3, 2015). This also includes notation systems of varying quality – Member B suggests “a ‘light’ typesetting mode replacing the typesetting algorithms which are not linear in time by simpler linear algorithms accepting a slight decrease of the score quality” (Member B, personal communication, January 8, 2015).

3.2.3 Standards

Most tools that deal with parsing LilyPond input and output have also called for rigorous, informal standards of structuring LilyPond input and output so that tools can operate using common assumptions. Frescobaldi writes “It is helpful...if the LilyPond interface (i.e. its input file format, its log output and command line structure) is cleanly structured and well described and doesn’t change too often. ... It would be nice if the LilyPond log output explicitly names the created files [from certain batch processing jobs], so Frescobaldi needs not to guess which output files were generated by a LilyPond job” (Frescobaldi, personal communication, January 6, 2015). The Music Notation Project calls for standards to be aligned with “improved MusicXML import/export” (The Music Notation Project and Clairnote, personal communication, December 22, 2014).

4. CONCLUSION

In 2015 and 2016, The LilyPond Consortium will act upon the results of this survey in order to bring improvements to the LilyPond code base by making it more modular, adapting it to alternative forms of notation and advancing a set of specifications for LilyPond input and output. Before June 2015, a development roadmap with concrete targets will be laid out by The LilyPond Consortium and effort will be undertaken by members of The LilyPond Consortium and freelance developers. By attending TENOR, MEI and ICMC, The LilyPond Consortium hopes to raise awareness about this initiative and grow its member base.

Acknowledgments

The LilyPond Consortium would like to thank its volunteer board for helping to prepare this document.

5. REFERENCES