ABJAD:

AN OPEN-SOURCE SOFTWARE SYSTEM FOR FORMALIZED SCORE CONTROL

Introductory Workshop

Trevor Bača ¹ Josiah Wolf Oberholtzer ¹ Jeffrey Treviño ² Study Day on Computer Simulation of Musical Creativity (Saturday 27 June 2015)

¹Department of Music, Harvard University

²Department of Music, Colorado College

TABLE OF CONTENTS

- 1. An example: rhythmic construction
- 2. Abjad?
- 3. Music
- 4. Companion projects
- 5. Conclusion

INTRODUCTION

The Abjad API for Formalized Score Control extends the Python programming language with an open-source, object-oriented model of common-practice music notation that enables composers to build scores through the aggregation of elemental notation objects.

AN EXAMPLE: RHYTHMIC CONSTRUC-

TION

ABJAD?

HISTORY

- · C into Finale via MIDI (1997)
- · Mathematica into Sibelius via MIDI (2001)
- · Mathematica into SCORE (2003)
- · Mathematica into LilyPond (2004)
- · Python into Adobe Illustrator (2004)
- Python into LilyPond (2005)
- Max/MSP into MS Access into Adobe Illustrator (2008)¹
- · Public release on GoogleCode (2008)
- Migration to GitHub (2011)
- Abjad 2.16 released (2015)

¹An attempt by Josiah before discovering Abjad.

Table 1: Abjad's Software Stack

Python				
Abjad				
SCORE	LilyPond	Steinberg?		

OBJECT MODEL

Abjad models musical score as a tree of components

Containers, leaves, spanners & indicators

Relationships between objects are modeled explicitly

Parentage, lineage, logical tie, logical voice

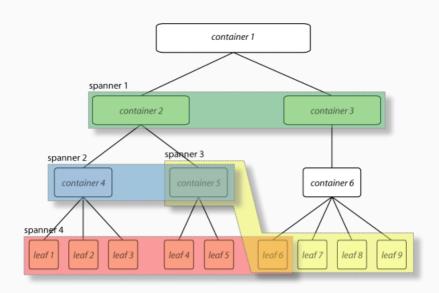
Primitive objects are also modeled explicitly

Duration, Offset, Pitch, PitchClass, Interval, Octave, Accidental

Top-level functions expose higher-level interfaces

Inspection, iteration, selection, mutation, persistence

CONTAINERS, LEAVES & SPANNERS



ABOUT THE CODE BASE

- · 496 public classes
- 387 public functions
- · 186,963 lines of code
- · 9399 unit tests
- · 10190 documentation tests
- · 100% free & open source
- · platform independent
- · runs under both Python 2.7 and 3.3+

MUSIC

JOSIAH'S MUSIC

2015 Invisible Cities (iii): Ersilia for chamber orchestra 2015 Invisible Cities (ii): Armilla for viola duet 2014 Invisible Cities (i): Zaira for eight players 2014 Plague Water for bari sax, e-guitar, piano and percussion 2011 Aurora for string orchestra 2010 Lagartija for mixed quartet

TREVOR'S MUSIC

JEFF'S MUSIC

2015 On the Behavior of Climbing Plants for chamber orchestra

2013 The World All Around for Eb clarinet, prepared piano, and harp

2013 +/for twenty french horns

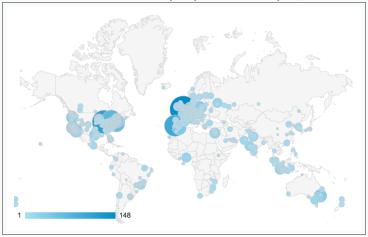
2013 Enfilade, Moses All, and Future Calendars for carillon

2011 **Being Pollen** for solo percussion

OTHER COMPOSERS

Mike Solomon Fredrik Wallberg Oscar Dub ???

Documentation visits by city since January 1st, 2015





IPYTHON

Abjad integrates with **IPython** (*ipython.org*/):

IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers enhanced introspection, rich media, additional shell syntax, tab completion, and rich history.

IPython integration was spearheaded by **Prof. George K. Thiruvathukal** (http://thiruvathukal.com), Loyola University Chicago.

Sasha provides a database of saxophone multiphonic recordings and their associated fingerings, allowing users to search for related multiphonics by timbral, harmonic and idiomatic descriptors.

- http://sasha.mbrsi.org
- http://github.com/josiah-wolf-oberholtzer/sasha
- Multiphonics performed by **Eliot Gattegno** (http://eliotgattegno.com)

Abjad acts together with many other Python libraries to handle programmatic notational output, and to perform validation on musical queries.



CONCLUSION

The Abjad API for Formalized Score Control extends the Python programming language with an open-source, object-oriented model of common-practice music notation that enables composers to build scores through the aggregation of elemental notation objects.

ONLINE PRESENCE

Documentation

http://projectabjad.org

GitHub Repository

http://github.com/Abjad/abjad

User Mailing List

http://groups.google.com/group/abjad-user

TENOR 2015 (GITHUB.COM/ABJAD/TENOR2015)

ABJAD: AN OPEN-SOURCE SOFTWARE SYSTEM FOR FORMALIZED SCORE CONTROL

Trever Bala
Rerord University
Trever Indexparil comJeffrey Trendin
Carlenn College
Self-rey, trevel conjugation.

ABSTRACT Formalized Score Cor

The Aspir APT for Formation Scote Control controls the Python programming language with an open-conce, objecoriented model of common-practice music notation than enables component to build scores through the aggregation of identical notation objects. A summary of widely used notation system: Introduction an advancion of the programming of the programming of the programming notation system: Introduction and security of the python of the programming of the programming programming the programming of the programming programming the programming programming and programming pr

L INTRODUCTION

Night I is no speciment offeren open designed to Might I in a region of the speciment of th

http://www.projectolopid.org http://www.proben.org http://www.bhysomil.org

Cappright 60/61 Brear Balls et al. This is an agent-access article durch under the sense of the Country Common deviluation 10/Enganed Livers, at position accessional use, distribution and reproduction in any medium, position original analysis and converse or credited.

2. A TAXONOMY Many software crossess intolement models of music but few

of the is implement a model of mention. "Maric untrasses of systems that model is marined on and be calculated according to their generation tanks." Many mentions on your consecution of their production tanks. "Many mentions on your consecution of their production o

Comparison to make a managed by any other properties of the contract of the co

What compares we trieg buildings of our often institute to substitute of the monthly defined principles, a compared simulated the compared to the formally defined monthly [10]. Excepting in state processor to also are deplicated former and its bounday would be believed as emission content and the compared of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the substitute of the compared of the compared of the compared of the compared of the committee of the compared of tion. Such systems might go so far as to enable scripting, as in the case of Sibelian's ManaGeriety [39] scripting language or Litypenal's embedded Scheme code; although these systems enable the automation of notational elements; it remains difficult to model compositional processes and

Other species are related to the contraction of the

uniforcisjes. Many model of marical nourism have been designed for purpose of corpus based computational maciology. Ferrame man in 1900. 2003. House, limitation and share-ferrame man in 1900. 2003. House, limitation and share-ferrame many of the share and though a lange amount of data [68]. Communical notions of lange amount of data [68]. Communical notions of lange amount of data [68]. Communical notions of the lange amount of data [68]. Communical notions of lange and lange and

3. ARJAD BASICS

Abjal is not a smale slowe application. Nor is Abjal a programming language. Abjal instead adds a computational model of music notation to the Python programming langgauge. By designing Abjal as an enuncion to one of the most widely used programming languages in the words, we hope to make a considerable collection of programming both processors are all the collection of programming and the processor are all the collection of programming and "An attempt to survey more comprehensively the likeway of depicsively and the collection of the collection of the collection of "An attempt to survey more comprehensively the likeway of depicsively and the collection of the collection of the collection of the "An attempt to survey more comprehensively the likeway of depicsively all the collections." way. Abjud is implemented as a Python package. ^{7 8 9} Composers work with Abjud exactly the same way developers work with all the other packages available for the language. In the most common case this means opening a file writing code and content to file.

- of main proceed region (region processes, manner, region properties
 - opin s Topin from dension and ratio inc dension, anter supies properties) and dension and supies properties) and supies substitution mann s (1) at seater supies substitution (1) (1) at a star supies (int, ind) apinal size s temperor yet ignised size)

The content of the file can then be used in other Python files or in an interactive session: or righted and a staff (context, name intertwinted of the

is paper demonstrates most examples in P

This paper domonerates more examples in Python's interactive controls because the console helps distinguish incifront compare. The however, component work with Abjud primarily by typing nonzionally-enabled Python code into a collection of intervitant files and managing those files as a project grows to encompose the composition of an entire

4. THE ARJAD ORJECT MODEL

and indicators. Every materianal administra in Alpha Monley, and The Commission of t

content and one of open to open to the content as greater as ever or "The inseparation of the content and the content as greater as the content and c

PERSONAL CONTACTS

Trevor Bača

- trevor.baca@gmail.com
- trevorbaca.com
- · github.com/trevorbaca

Jeffrey Treviño

- · jeffrey.trevino@gmail.com
- · jeffreytrevino.com
- · github.com/jefftrevino

Josiah Wolf Oberholtzer

- josiah.oberholtzer@gmail.com
- · josiahwolfoberholtzer.com
- github.com/josiah-wolf-oberholtzer