GitHub: DayToScore\Originator\Vertex\Memos\

Tempo_FunctionalSpecifications.rtf

Target date: Monday 9/3/2023 or sooner

Art Lenskold

327-4 Evans St.

Williamsville NY 14221

dloksnelATgmailDOTcom

St. Gregory the Great Catholic parish

Problem:

We need to honor Pope St. Gregory more on and after his feast day.

Solution:

Sing the Gregorian Missal chant, in the spirit of Abbey Solesmes, France.

This DaytoScore computer programming project intends to use a development platform for crowdsourcing, such as wwwDOTgithubDOTcom, and wwwDOTskypeDOTcom for communications.

In order to keep it simple, and to promote self-documenting, I recommend using a programming language that has classes and methods, such as the cuurent version of the Python language that is 100% object-oriented, together with object-oriented databases.

I recommend quarterly communications, each lasting a microcentury of time (53 minutes), liberally interpreting The Standard Code of Parliamentary Procedure by Alice Sturgis. Begin with a prayer for the intersession of the saints: Saint Isidore of Seville, patron saint of the end-users/beta-testers, Saint Peter Regalado, the patron saint of teams of developers, parallel progamming, quantum computing, and contemplatives, and Blessed Carlo Acutis, the database developer.

Quarterly highlight reports tell what each beta-tester and developer has learned, if anything. Art Lenskold is in the running, as a volunteer, for the office of a recording secretary, not necessarily also corrresponding secretary, ending when the liturgical year ends on 1SundayAdvent of 2021.

Functional Specifications:

In structured English:

A system called Day To Score takes:

Input:

year, month, day, hour,

that comes from a smart phone,

and produces:

Output:

a link to a Proper of the Mass sequence of about 6 musical scores, written in melismatic (Solesmes) Gregorian chant, in a modern musical staff, with lyrics in English that has the approval of the Catholic Church,

that goes to the smart phone.

Schedule for the primary solution: N/A

The Gregorian Missal chant music that the whole congregation sings is prepared in 6 stages.

Consider, for instance, the Introit and Offertory for the 22nd Sunday in ordinary time. (I don't know where to find them in soft copy). Somewhere, its Gregorian Missal chant has already been written in Latin using square notes.

The first step is to write the melismatic chant in Latin using round notes. One could hope that a computer program is already in use to accomplish this. Otherwise do the transcription manually using either Finale or MuseScore3.

The second step is to translate Latin words to English words.

The third step is the interplay of steps 1 and 2.

The musical interplay of step 3 is as follows:

Use an easy-to-use cloud medium for communication, such as Google Drive, preferably without encryption.

The fourth step is to add a tempo.

The fifth step is to add dynamics of softening the singing.

The sixth step is to add security.

The seventh step is to add cookies.

Detail:

In a phrase of one breath of Psalm verse, make wise use of a leading half-measure rest.

The goal is to maximize the placement of whole notes on the downbeat.

This avoids some complications of jumbled half and whole notes.

Use a triple of (2, 1, 2) whole notes to represent a 3-syllable monotone on the downbeat.

For a 4-tuple of (2, 1, 2, 2.5) whole notes, append a monotone half-note to the final slur.

Add appropriate dynamics for a strong sudden accent, then softening the singing of the groupings between rests (decrescendo).

Scenario:

A traveling communicant enters a Catholic church 15 minutes before the celebration of Mass. Five minutes is spent adoring God. The communicant next opens a tablet or smartphone containing the DayToScore app and inputs a day as YYYYMMDDHH. The app outputs a set of musical scores:

the entrance antiphon

the responsorial psalm

the offertory antiphon

the communion antiphon.

In the final 5 minutes before Mass, the communicant practices humming or softly singing from the musical scores that are displayed on the media.

Background:

The author is aware of a competing project.

It is a project that merely uses a somewhat private translation of Sacred Scriptures into English.

And it uses merely plainchant, not the melismatic chant of the Abbey Solesmes.

Input sources for the Day To Score project backend:

Primary source of the lyrics (95%):

The Bible translation (on-line) by the United States Conference of Catholic Bishops (USCCB) in the idiomatic English of the present day.

Secondary source of the lyrics (5%):

The translation by Abbey Solesmes, perhaps as a JPEG image, of selected pages of the book, Gregorian Missal (the 2012 edition), for non-scriptural syllables, such as Alleluia.

Source of the musical notes (100%):

Abbey Solesmes square notes pre-processed as musicxml for use as an input digital database of musical scores to the DaytoScore server. The input to this pre-processor is to originate exclusively from the Gregorian Missal, and it is of monasterial vintage approximately 150 years or several centuries prior to that, going back to the Middle Ages.

I think that this computer program, the square note pre-processor, has already been developed as a GitHub open-source project by someone else, to whom I am grateful.

Source of the finished product, a sample set (3%):

Ideally, machine learning from the 3% sample can be processed as a batch outputting to the remaining 97% of the musicxml scores, and the output database can be accessible in real-time on the server.

The sample size is 32 completed musical scores, and it was generated manually by the author of Day To Score over the course of 1.7 years, whose age was just short of his 80th birthday.

The author is praying that the GitHub workers, skilled in artificial intelligence, can achieve 100% completion with a limited diminution of quality.

Quality:

Here is the author's attempt to define the quality of design:

Client performance:

a turnaround time not to exceed 10 seconds.

Client ease of use:

Default the YYYYMMDDHH input to the smartphone's GPS time of pressing the submit button.

Ease of scrolling a 2-dimensional musical score while the user is singing it.

Some syllables of slurs exceed 50 musical notes, with groups of notes separated by half-measure rests in groups of 12 notes or less. But, generally, keep a slurred syllable on at most a single screen of the smartphone.

Client audio:

There is no audio, so Day To Score is only suitable for human singing.

Preliminary education for all developers:

Read "The Gregorian Missal for Sundays and Solemnities Notated in Gregorian Chant by the Monks of Solesmes", Solesmes, 2012. (hardcover, with an imprimatur, $26), wwwDOTsolesmesDOTcom - editionsATsolesmesDOTcom. Only you, as a team of developers, will need to understand the frequent use of podatus and porrectus in the

chants of this book.

Your check memo with the following wording written in French and English:

Pour aider en priorité l'Abbaye Saint-Pierre de Solesmes/ 'To help the Abbey of Saint-Pierre de Solesmes as a priority'.

La Fondation des Monastères
14 rue Brunel
75017 PARIS
FRANCE

Read "Catechism of the Catholic Church", Libreria Editrice Vaticano, Citta del Vaticano.

(paperback, with an imprimatur, $30). This book will enhance your understanding of the Holy Sacrifice of the Mass.

Education for those developers who use machine learning:

1. The words of both Sacred Scripture and the Gregorian Missal are more important than the musical notes that are to be sung.

2. Verify that the words from the author's sample set are the same as the words online of the current wwwDOTusccbDOTorg version of the Sacred Scripture (United States Conference of Catholic Bishops). Therefore, there is no private version of the Sacred Scripture.

3. Notice that the word "Alleluia" is the most notable of the exceptions. Its musical notes, together with its words, have been copied directly into the author's sample set. Another exception is some antiphons on the days of solemnities. I'll just refer to both of these as the "alleluia exceptions."

4. I think that someone has a tool for making 2-dimensional Gregorian chant (square notes) into another musicxml that can be "more easily machine-readable," with its own open-source syntax.

In particular the new syntax would portray podatus and porrectus with accompanying lyrics.

This is useful for the alleluia exceptions.

5. I think that a similar tool can transcribe the author's sample set into the same "more easily machine-readable" syntax.

6. See the "Tips for Developers" section below for a rule-based set of specifications for

producing a musical score that is to be open-source for editing.

7. By means of crowdsourcing on the GitHub platform allow a team of developers to refine their rules, in order to approximate the author's sample set of size 32.

8. Optionally, discard the end-result of the set of approximations of size 32, but not the rules.

Keep them in a Rules Database on the server.

9. "Randomly", choose, without repetition, a few old square-note scores from Solesmes's Gregorian Missal, and through machine learning, teach the server, using the Rules Database, to generate a few round-note scores with lyrics. The quality of the lyrics dominates the quality of the musical notes.

10. Machine learning produces the completed result for a population size of about 900 musical scores with lyrics. There are about 900 old musical scores in the Roman Missal. The generation of about 868 new scores electronically saves a lot of manual work.

11. Save the 900 or so scores in a pair of databases on the server: an Uncompressed Final Database and a Compressed Final Database.

12. The end-user needs to have a human-readable subset of about 6 musical scores with lyrics within 10 seconds of requesting them on a smart phone.

Tips for Developers

Tips on selecting a random sample:

It can be from a uniform distribution, without replacement, on the beginning and ending pages of the Roman Missal. The beginning page contains an entrance antiphon for the First Sunday of Advent. The ending page contains the communion antiphon for whatever is the last solemnity Mass in the book. If there are multiple scores on a page, choose the first one.

Tips for identifying the header information on a musicxml score:

The Title:

The title serves two purposes. It uniquely specifies the name of the musicxml file in the population of about 900 Solesmes Gregorian Missal chants, and it is easy for the end-user to understand.

The title is shortened, in the Solesmes Gregorian Missal, from the heading on the top of either the current page, next to the page number, if non-blank, or the following page of chant.

Some examples:

2February,PresentationOfTheLordAntiphon3 is derived from 2 February, Presentation of the Lord Antiphon 3.

The only punctuation mark carried over is the comma. All spaces are deleted. The lower case words, of and the, are capitalized. The end-user sees both brevity and the absence of confusion caused by the hyphen or the underscore, that is, - or _.

TheBodyAndBloodOfChristEntranceAntiphon is derived from The Body and Blood of Christ Introit.

See the sample set of size 32 for other modernized idioms, because the word Introit isn't used much anymore.

The server will need at least one lookup table named TitleIdioms.

The server will also need to make sense out of the end user's input of YYYYMMDDHH.

On Christmas, there are 3 separate Mass liturgies, depending on the time of day, so that the rarely-used HH designation, and am or pm, comes into play.

The client code would resolve am or pm into HH military time, and would resolve whether the end-user chose to take the default YYYYMMDDHH.

Furthermore, the Catholic Church has a Liturgical Cycle that repeats every third year.

And the cycle years are labeled A, B, and C, corresponding to 0 mod3, 1 mod3, and 2 mod3 that accommodates an instance of YYYY.

Therefore a chant title sometimes needs a suffix, and, for the sake of brevity, the Abbey of Solesmes has only printed some alias title, such as, one alias from the following list:

...CommunionAntiphonB

...CommunionAntiphonA&B

...CommunionAntiphonB&C.

There is no ...CommunionAntiphonA&B&C alias. You simply leave the suffix blank.

One complication is that the Liturgical Year begins on the First Sunday of Advent, and not on January 1. This may be a Saturday evening or a Sunday in November or

December of the year prior to the Liturgical Year. And it necessitates a reinterpretation of YYYYMM.

Here is a work-around: Reinterpret YYYY12 as ZZZZ00, where ZZZZ is YYYY incremented by one year. This ought to work. Although the First Sunday of Advent might accur in late November, notice that, in the current edition of the Gregorian Missal, all the chants for that Sunday do not vary from year to year.

Another anomaly is that sometimes the Gregorian Missal gives the singer a choice of chants. For example, you should use the following 2 titles:

Mary,MotherOfGodAlleluia1, and

Mary,MotherOfGodAlleluia2.

The worst case scenario is that the server presents both of these to the client, and the end-user picks one of these to sing, following trial and error.

Also, on certain pages of the Gregorian Missal, there may be a link to another page for the sake of brevity. Often, there is a Latin keyword that precedes the page number. So there may be an alias title with a fixed subtitle in the round-note musicxml output. Since the outputs to the client are usually about 6 at a time, the correct alias title should be clear from the context.

The Subtitle:

There is no subtitle if the lyrics follow no book of the Bible. (Neither is there a lyricist).

Otherwise, the Solesmes Gregorian Missal abbreviates the book(s) of the Bible in the upper right corner introducing the square-note score. If there are 2 books of the bible that are abbreviated, then choose only the first abbreviation and expand it into a full word using the SubtitleLookup1

table on the server. Thus, "Thess." becomes Thessalonians.

And "Ps." becomes Psalm in the case that the psalm is listed alone, by itself. In the latter case, an orphaned psalm, which occurs often, you use the SubtitleLookup2 table (5 rows) to append to the subtitle and its verses one blank space, and then one of 5 psalm types. In the author's opinion, the psalm type is a guide to the emotional state in the life of the psalmist that allows for better singing of the score. And there is room available for at most 5 more words on the subtitle. See the author's sample set of musicxml scores for many examples of the psalm type.

For the 150 psalms, there is a SubtitleLookup3 table, because the numbering of the psalms is inconsistent, whether or not the psalm is orphaned. Usually psalm # (n) in the Solesmes Gregorian Missal numbering system corresponds to psalm # (n + 1) in the United States numbering system used by the United States Conference of Catholic Bishops, but not always. In reference to the last paragraph, in the United States numbering, check the Appendix A to these functional specifications for details on the range of numbers in SubtitleLookup2 (5 rows). The word "Psalm" and its verses are omitted from the subtitle if the psalm is not orphaned.

The psalm types, with an interval expressed in United States numbering, are:

A Psalm Of Individual Lament	1-41
A Zion Psalm	42-72
A Psalm Of Communal Lament	73-89
A Historical Psalm	90-106
A Psalm Of Ascent	107-150

Composer:

the page number (nnn), the page suffix, one blank, followed by Solesmes.

Insert nnn, where nnn is the page number in the 2012 edition of the Solesmes Gregorian Missal, for example, nnn is 214 in the case of:

Exsulta, p. 214.

This information is for verifying a link, because brevity is a feature of the Roman Missal.

In most cases the page suffix is one blank, but it is important to mark the suffix a or b in the rare cases when 2 Solesmes square-note scores begin on the same page.

This means that the server must verify each backward target page of a link, assuming that the server generates round-note scores in the order of ascending page numbers.

When viewed by the end-user, it is best not to emphasize the page suffix technicality, in general.

The absence of a page suffix is the default, equivalent to the page suffix a.

Lyricist: United States Conference of Catholic Bishops (97% of the time).

For a few scores, the author is aware of no online source of the lyrics, so leave the lyricist blank. The Abbey of Solesmes has both the lyrics and the square-notes in printed form, e.g., the Palm Sunday Responsory (one of the "alleluia exceptions").

Copyright: leave it blank

This completes the identification of the header information.

The staff information:

You use only the treble clef.

You omit the tempo.

The tempo is calculated at a much later stage.

The mechanism for reinserting the tempo, will most likely be this: The server exports the transcribed round-note score, saves a copy of it, starts over with a blank score, inserts both the newly-calculated tempo and the saved number of flats, and imports the saved copy.

You use the most common signature. That is 4 over 4, so as not to disturb the end-user. It would be more of a disturbance to use the uncommon 2 over 2, representing 2 half-notes per measure, even though the author's sample set of musicxml scores contains no quarter-notes.

The number of flats:

Prefer to use only a small number of flats, if at all. No sharps are allowed.

Choose to use at least one flat in the round-note score at the start of each line if the square-note score contains one flat. The square-note score never has flats on 2 different tones, unless they are an octave apart. And, if a natural seemed to be intended, make it a flat, for ease-of-use.

For usability, avoid using any flats imbedded in the round-note score.

Human voices have evolved along a 1-dimensional non-linear scale: bass, contralto,

tenor, and soprano. The author's favorite singing octave begins at B-flat. The central note of my choice is E-flat. And the note, the A below middle C, is tuned at 220 cycles per second.

Each Solesmes score has a spread of notes. I call this spread the domain of the square-note score. Seek to make E-flat the central note of the round-note range that is the output of the domain for just about every score.

Already we are approaching machine-learning.

Solesmes seems to be providing a vehicle for both sopranos and basses to sing, but I think that this is an illusion. Since the square-note staff has only 4 horizontal lines, rather than 5 lines, it is merely necessary to introduce the 2 clefs, the do-clef and the fa-clef, in order for the resulting score not to wander too far up or down from the staff. The sample set of size 32 mostly ignores, whether the usual position of C (for the do-clef) or the unusual position of F (for the fa-clef) on one of the 4 horizontal lines of the staff, may seem to favor sopranos or basses.

Nevertheless, when the fa-clef is specified (always to the exclusion of a do-clef in the same score), it is good practice to include at least one flat in the output musicxml score, because both F major and F minor use flats.

The goal is for the singer to meditate on the meaning of the words of the score, rather than the necessity either to sometimes sing the notes transposed an octave down, or not to sing.

There is a complication in calculating the domain and the placement of its central note in the range. Some square-note scores include one shift of the position of the do-clef indicator down to one of the 4 horizontal lines that is below the original position. This happens occasionally in the middle of a 2-part score, for example, the Book of Wisdom shifts to the Book of Psalms.

The calculation of the central note of the domain becomes more critical if the ideal target range is still to be the octave beginning at B-flat. And the end-user wants no change in the number of flats throughout the score, for ease-of-use.

Concerning minor key or major key, Solesmes provides a Roman numeral just to the left of the top staff. It is from I to VIII. For simplicity, let the even Roman numerals (0 mod 2) be minor keys and the odd numerals (1 mod 2) be major keys. And try not to introduce too many flats in the key signature, if any.

The syllables of some chants follow along a single recitative note, perhaps for a half dozen of Latin syllables. The do-clef's position on one of the 4 horizontal lines of the

staff is often used to mark the note C for the recitative note of the entire score, not from just the second portion that may be the Book of Psalms. The note C is appropriate because the do-clef for the C major has no flats, a usability advantage for both minor and major keys.

In summary, the choice of the number of flats is complicated and is important.

The leading tone:

A leading tone is infrequently used in the musicxml output. It is for usability. It is advisable to examine the entire square-note score before deciding to use it.

The number of measures:

Choose 120 empty measures.

Composing the music:

The design methodology is partly top-down, partly bottom-up, and having a conclusion somewhere in the middle.

The implementation methodology for the music perhaps can be bottom-up.

The round-notes are primarily whole notes, and secondarily half notes, for usability.

The final notes of a square-note score are rarely a porrectus or even a podatus. Yet I think that the first notes of the score may be a podatus.

Hence, by putting down the predictable ending whole notes right away, the machine-learning proceeds better in a bottom-up methodology.

Concerning the ending alleluia, the word is common to both English and Latin.

The majority of Solesmes square-note scores do not conclude with one or more alleluias.

These are not simply alleluias in plain chant notation.

And the syllables of the alleluias are not listed by the Book of Psalms or any other.

It seems to be a manual work item to insert rest half-measures into the alleluias of the Solesmes Gregorian Missal and to rewrite the square-notes as round-notes after the principal draft of the output musicxml has been completed by machine-learning. Do likewise for many more of the "alleluia exceptions" that may or may not contain prolonged melismas.

This "alleluia exceptions" issue is a big one. It would be wonderful if a team of artificial intelligence programmers can automate a system called Alleluia to take a random page of the Solesmes Gregorian Missal as a jpeg photocopy, identify alleluia exceptions, insert half-measures of rests into them, and output a few segments of the notes and associated syllables as round-note inserts to the associated round-note musical composition's unfinished database entry.

And Solesmes Abbey is due at least a note of thanks for the selective use of photocopies.

I don't know of another innovative automated solution to the "alleluia exceptions" issue.

Suppose now that a randomly selected Solesmes square-note chant contains no "alleluia exception," and is like the majority of elements in the sample set of size 32. This supposition is equivalent to total reliance on the online Bible of the United States Conference of Catholic Bishops for he lyrics of each of those elements.

Suppose, furthermore, that the square-note score contains no psalm.

And suppose it has no A-B-C suffix on the title.

Call a square-note score "simple" if it satisfies the 3 conditions above. One of the few examples of a simple score is the 2February,PresentationOfTheLordAntiphon3 occurrence in the sample set of size 32. I call it the "prototype" score, for short. Other simple scores from the sample set are:

3AdventCommunionAntiphon,

HolyThursdayAntiphon6,

TheSacredHeartCommunionAntiphon.

The Prototype Score's Development:

In summary,

1) Develop the round-notes and rest measures.
2) Develop the stress levels and decrescendos.
3) Develop the slurs.
4) Develop the English lyrics.

The engine that drives the development is the syllables of the English lyrics.

And the fuel for the engine is how long a congregation of people can hold their breath.

In detail,

1) Obtain a page database. These are a sequence of JPEG images of the 2012 edition of the Solesmes Gregorian Missal from page 177 to page 781.

2) Model the square-notes of the page database as a tree with 4 levels. The MAGIC criteria ought to be applied to this hypothesis:

M - magnitude, how big is the effect?

- A - articulation, how precise is it?
- G - generality, how widely does it apply?
- I - interestingness.
- C - credibility. Is it backed by theory? Is it possible?

3) The levels of the page tree have 4 levels: 0, 1, 2, and 3.

The specification of these levels is dependent on certain software, as in the following introduction to the page levels:

4) I think that someone has a tool for making 2-dimensional Gregorian chant (square notes) into another musicxml that can be "more easily machine-readable," with its own open-source syntax.

In particular the new syntax would portray podatus and porrectus with accompanying lyrics.

5) In order to get some representation of Gregorian Missal chant, there is a need for the server to scan its JPEG photocopy image, primarily to get the square-notes, and secondarily to get the matching Latin lyrics, with or without some assistance from the Abbey of Solesmes. I recall that the monks began their own collections and transcriptions in the nineteenth century.

"There is no contradiction between the two editions of the Roman Missal. In the history of the liturgy there is growth and progress, but no rupture. What earlier generations held as sacred, remains sacred and great for us too, and it cannot be all of a sudden entirely forbidden or even considered harmful. It behooves all of us to preserve the riches which have developed in the Church's faith and prayer, and to give them their proper place. Needless to say, in order to experience

full communion, the priests of the communities adhering to the former usage cannot, as a matter of principle, exclude celebrating according to the new books. The total exclusion of the new rite would not in fact be consistent with the recognition of its value and holiness."

wwwDOTvaticanDOTva/content/benedict-xvi/en/letters/2007/documents/hf_ben-xvi_let_20070707_lettera-vescoviDOThtml

6) Get the corresponding English Bible passage from the website of the United States Conference of Catholic Bishops, and count the syllables in the passage.

7) Compare the syllable count, s, to the "note count," g, of the Gregorian Missal chant. The "note count" is a count without repetition of consecutive tones And so the value of g will be less than the actual count of notes.

8) For example, consider our prototype score, 2February,PresentationOfTheLordAntiphon3. Its actual count of notes is 241. However, the value of g, the "note count", without repetition of consecutive tones, is only 196. This value is easily verified by counting the notes in the finished round-note musicxml from the sample set of size 32. The syllable count, s, is 97, from Luke, chapter 2 on the United States Conference of Catholic Bishops website.

9) The page tree introduction is complete when the server has calculated the ratio, g/s, of the "note count" to the syllable count of the entire score.

10) For our prototype, the ratio g/s = 196/97 = 2.02 notes per syllable. Now, if you count the the number of slurs in the completed round-note score as s (regarding an unslurred note as a slur of length 1), then the average length of each slur is a little over 2 notes (for the prototype).

Tree Level 0

1) The singers of Solesmes chant, unlike the singers of plain chant, need to rest in order to take a breath of air. This functional specification is that a rest be inserted into the score after no more than 12 notes that are counted without repetition of consecutive notes. An exception is that the limit is 14 notes if the server encounters a word of 4 or more syllables.

2) Since the server has not yet determined the key signature, for the sake of draft #0, and for all scores, make it initially the key of D minor, with just 1 flat. Note that the number of musicxml scores in the sample set of size 32 is 15 minor keys and 17 major keys. Therefore, the choice of D minor, rather than F major, is a random choice. And D minor just happens to be same key signature as the prototype score in its final musicxml

representation.

3) Using exclusively 1 whole note to each round-note measure, it is therefore necessary that the

score begin on the downbeat without any leading tone. Draft #0 will contain the actual count of notes, not the "note count." Every podatus will become 2 whole notes and every porrectus will become 3 or more whole notes. And both of these will appear in the correct sequence of their internal notes. For example, draft #0 of the prototype consists of 241 whole notes.

4) The server may accomplish draft #0 in this manner. First, ensure that the horizontal lines of the square-note score's staff are truly horizontal. (I am assuming that a square-note score comes from a JPEG photocopy). Second, I think that someone has a tool for making 2-dimensional Gregorian chant (square notes) into another musicxml that can be "more easily machine-readable," with its own open-source syntax. Third, use this machine-readable syntax to produce draft #0, with its key signature of 1 flat, in the traditional syntax of a round-note musicxml score.

5) Identify, for each D-minor score, both the highest note to be sung and the lowest note. Compute the central note.

6) Use existing musicxml software to transpose the score, if necessary, so that the new central note of the score for draft #0 is E flat. The resulting number of flats will be in the output to the end-user. See the heading above, "Number of Flats."

Tree Level 1

1) The prototype lyrics contains 4 verses of St. Luke's Gospel. If you count the 2 vertical lines ending the score as just 1 full vertical line, the prototype square-note score contains 4 full vertical lines.

Because of the 4-for-4 correspondence, this is the ideal pair of scores.

A full vertical line in the square-note score extends all the way from the top line of the Gregorian staff to the perpendicular bottom line.

2) In the prototype round-note score, insert 3 "primary rests," that is, a full measure rest at each

of the first 3 occurrences of full vertical lines in the corresponding prototype square-note score.

3) These insertions of primary rests partitions the page level 0 tree, creating 4 new ordered edges, called "limbs," that define the tree level #1 ordered segments of the prototype's tree-level structure.

4) Let the sequence of whole notes in each of the ordered segments of the new page-tree end with a primary rest, except for the last segment.

5) I am making no separate recommendation yet for the server to pay attention to the partial vertical lines in the square-note score. These partial vertical lines are centered. And they extend only 70% of the distance between the bottom horizontal line of the staff and the top line.

6) Draft #1, for the server software, is now complete. The tree trunk has grown 4 limbs.

Tree Level 2

1) Consider Limb #1 of the tree. The objective of the succeeding tree level is to determine whether the round-note score is to begin on a downbeat or on an upbeat (a leading tone).

2) If the first syllable is the English word "the", then the score should usually begin on an upbeat. The only possible exception is to let the round-note score begin on a downbeat, if the first square-note neume is a podatus. The server is not yet adding slurs to the round-note score, but, for this exception, the slur would extend for no more than 2 round-notes. In singing, you don't want to prolong the definite article, "the."

3) Referring to the round-note score for the prototype musicxml, the number of measures is 56 measures. But Limb #1 contains a lot more measures, 89 of them. The final 3 syllables are "of the Lord." The server places the definite article, "the," on the shortest duration as possible. And this duration is to be the half-note, that has yet to be placed properly.

4) The next considerations are the Deity words, "Lord, God, Yahweh, Father, Jesus, Christ, Paraclete," and, when capitalized, "Spirit, Consoler." (In the sample set of size 32 there might be a few other Deity words). There should be a full measure rest after the final slurred note of each Deity word in order to allow the singer a pause to meditate. I call this the "Deity rest", possibly to be distinguished from the "primary full measure rest" at the end of each limb of the tree (except for the last limb). The server need not

insert a "Deity rest" if it coincides with the "primary full measure rest."

5) Let us redefine the "limb note count," g/s, to apply only to a limb of the tree and not to the trunk. For Limb #1 of the prototype, g/s = 73/30 = 2.43 notes per syllable. Henceforth, the value of g refers to a counting without repetition of consecutive tones. In the usual case, when g/s is large, a sequence of consecutive tones is represented as just one English syllable, and contains as many whole notes in the round-note score as is possible.

6) The last neume of Limb #1 of the prototype is the double-dotted podatus. Because of the dots, the durations of the 2 notes are extended. So the last syllable is "Lord." And it appears with the correct English punctuation. So I have represented this neume as 2 whole notes, for ease of use. In fact, it turns out that all 4 limbs of the prototype conclude with 2 whole notes.

7) Compute g/12, where 12 is the maximum number of different notes that I can comfortably sing on one breath of air. For Limb #1 of the prototype, g/12 = 73/12 = 6.08, which rounds up to 7. The first tree limb of our tree trunk has grown 7 new ordered edges, called "branches," that will hopefully define the tree-level #2 ordered segments of the prototype's tree-level structure. Unlike the straightforward 4-4 correspondence of tree-level #1, we have now encountered a stranger 73-7 correspondence for the branches of Limb #1 of the tree for the prototype. It turns out that the server ought to allocate the value of g among the 7 branches as: 12 + 12 + 11 + 8 + 9 + 11 + 10 = 73. Allocations such as these may or may not require the server to resort to machine learning from the sample set of size 32.

8) There are some tips for the developer of the server software in order to achieve the ordered set of branch sizes, in this case (12,12,11,8,9,11,10) for 7 branches. Although each branch terminates with some type of rest, it can now be either a full measure rest or only a half-measure rest. The exception is to have no rest at all, when the succeeding branch begins with a half-measure rest. Therefore, the server ought to allocate the branch sizes in the reverse order,

(10,11,9,8,11,12,12).

9) Having used machine learning already to assist finding the allocations of the branch sizes, machine learning may or may not also be useful for transcribing syllables to round-notes in the general case, without definite articles or Deity words and syllables. I hope that the machine learning will relieve the team developers from studying the Latin language. It seems enough for the team to understand the square-notes.

10) Draft #2, for the server software, is now complete. Each tree limb has grown approximately 7 branches.

Tree Level 3

1) Consider, in the order that a server would develop them, Branch #2 and Branch #1 of Limb #1 of the prototype tree.

2) For Branch #2, the only syllable of interest here is the first, because it is the word, "the."

The definite article is assigned the minimum number of round-notes, that is, just a half-note on the upbeat. The downbeat before the upbeat is a half-measure rest. Therefore, this information is significant to the server when transcribing Branch #1, because the singer is already assured of being able to breathe.

3) For Branch #1, the only the words, not the syllables, are provided by the United States Conference of Catholic Bishops: "It had been re-vealed to them by."

The server needs to have an online English dictionary in order to discriminate "re-vealed," as opposed to "rev-ealed."

4) There are already at least 3 anomalies with Branch #1: Anomaly #1: It is the top branch of a limb. Anomaly #2: There is anomalous information passed back from the succeeding branch. Anomaly #3: The lyrics contain one or more hyphens.

5) Anomaly #3, concerning lyrics, may or may not require that the server consult an online English language parser, as well as machine learning, in order that the corresponding round-notes acquire "artistic license." In the word "re-vealed", my opinion is that an artistic preference is for the accented syllable of the branch ought to appear on the downbeat of a round-note measure. And place the first syllable, "re-" on the upbeat of the previous measure, in order to minimize the number of slurs in the branch. The online English parser has determined that, in this branch, the verb, "revealed" is the most important word.

6) Suppose, instead, that the server was transcribing an "ideal branch," that is, none of the last 2 anomalies: no hyphens, and no anomalous information to be passed back.

7) For the ideal Branch #1, compress all its whole notes into half-notes.

8) The server will, at a later time, re-expand some subset of these half-notes into whole notes, for 2 purposes:

The whole notes tend to minimize the number of slurs, and

the whole notes need to be maximized, so that the singer will have ease-of-use.

9) The ideal Branch #1 of Limb #1 does not have a leading half-note to introduce the score on an upbeat. Neither does our prototype have anomalous information to be passed back to "Branch #0", that is, to introduce the score, undesirably, on an upbeat, it turns out. It is desirable to begin most scores on the downbeat, so that he singer has ease-of-use.

10) We are done with this particular task. The entire prototype final musicmxl score can be started on the downbeat in one of the existing software products for musical composition. Therefore the server may bypass checking a box for a leading tone of 2 quarter notes. And this action completes the last step in specifying the header information of the score.

The Tempo

Waves vary.

The usual duration of the celebrant, as he prays only the Offertory of the Mass at a Low Mass, satisfies the inequality:

1.5 minutes < Low Mass duration < 1.9 minutes. The tempo of the congregation's singing is in beats per minute. The tempo is inversely proportional to both the duration and the storage size.

The unusual duration of the celebrant, as he prays only the Offertory of the Mass at a High Mass, satisfies the inequality:

4.6 minutes < High Mass duration < 5 minutes,

The Low Mass duration is therefore 36% of the High Mass duration.

A book, The General Information on the Roman Missal (GIRM), or in Latin IGMR, governs the relative prevalence of low masses to high masses in a parish. The following reference may or may not be up to date:

wwwDOTvaticanDOTva/roman_curia/congregations/ccdds/documents/rc_con_ccdds_doc_20030317_ordinamento-messale_enDOThtml#C._The_Liturgy_of_the_Eucharist

At a Low Mass, a gold chalice contains an offering of wine.

Besides the gold offering, at a High Mass, a vessel containing burning frankincense is an

additional offering, requiring more time.

Since all masses are unbloody sacrifices of Jesus Christ, there is no offering of a tomb vessel, containing myrrh.

The first tip for developers was this:

"Tips on selecting a random sample."

The originator's sample was of size 32.

I estimate that anyone else's random sample of the same size, chosen without replacement, would turn up about 10 offertories, just as my sample did.

We need to record into the header of each final musicxml score its storage size in kilobytes, just as we did for the page number and suffix of the Gregorian Missal.

My sample of size-32 scores shows that many Solesmes Offertory scores can be chanted, at a Low Mass, in a duration of 1.7 minutes, in spite of the frequent encounters with relatively long sequences of slurs. The tempo is again in beats per minute.

We have 3 "beat choices" to measure a "beat."

1) beats per quarter note.
2) beats per half note.
3) beats per whole note.

Match up the storage size in kilobytes next to a suitable choice of which of the 3 "beat choices" measures the tempo of each of the 10 Offertory score's headers.

And then, use the same "beat choice" for all of the 900 headers in the population of musicxml scores, for ease-of-use at a Low Mass.

Many of the Offertories in the statistical population ought to be sung to completion at a Low Mass in 1.7 minutes. If some Solesmes score cannot be sung to completion at a Low Mass, then try scheduling a time slot for one of those unusual High Masses. Or else, at Low Mass, try singing the Offertory of the Missa Luba in its place, or some other hymn.

Never allow the audible ticking of a metronome during Mass, but include the metronome number in the score for practice.

Here is the selection of verbiages for the tempo that are to be included in each header:

1) Dignified
2) Slow
3) Ambulatory
4) Moderate
5) Lively
6) Presto (suppressed)
7) Prestissimo (suppressed)

Except for suppressed verbiages, the corresponding metronome setting that is also to be placed in the header of each score, in beats per minute, with the uniform 4/4 time signature is:

1) < 60
2) 60 < metronome < 79
3) 79 < metronome < 96
4) 96 < metronome < 122
5) 122 < metronome < 160
6) 160 < metronome < 190
7) > 190

I prefer #4, "Moderate."

We suppress Presto and Prestissimo for ease of use by the end user, and simply replace them with Lively.

As an alternative to the tip described above, there is sometimes a statistical approach of approximating the binomial distribution by a Gaussian distribution. Here are 2 reasons for not doing it this way:

1) A sub-sample size of 10 is too small.

2) We don't reject anything, like outliers, from "The Word", which is the inspired word of God, as indexed by The Sacred Tradition of the Catholic Church. On the wwwDOTusccbDOTorg web site, we use a public interpretation of the Holy Scriptures, with an Imprimatur. We don't pick and choose whatever is convenient for us to believe, through private interpretation.

We have targeted the Offertory, because the Solesmes Offertory chant seems to be the only extension of the Liturgy of the Word into the Liturgy of the Eucharist. There on the altar sits an expensive vessel containing a simple mingling of wine and water. Its aroma has a tendency to be tempting to some of the worshippers, especially those who may have observed the Eucharistic fast since midnight. And for priests who have talents other than that of being musically inclined, we must be steadfast to avoid the devil's workshop, until we beg God with a prayer from the ordinary of the Mass, "Pray brethren...".

For the 10 examples of tempo-related verbiage, search the output headers of the musicxml transcriptions for titles having the suffix "Offertory," in order to observe the 3 additions, that are about to be inserted.

For the other 5 parts of the Mass, exclusive of Offertory, sing them to completion, singing with emotion, occasionally without paying attention merely to their duration.

With a Lively tempo, sing a Psalm of Ascent. With an Ambulatory tempo, sing a Psalm of either individual lament or communal lament. Otherwise sing a Psalm at a Moderate tempo. And this paragraph applies only to non-Offertory Psalms.

Here is a failed attempt to assign a Low Mass Offertory Tempo:

Let f = f(x, t),

Dsquared(z) = a second partial derivative with respect to z,

v = wave speed,

and A = amplitude.

The partial differential equation,

Dsquared(x) – (1 / v^2)(Dsquared(t)) = 0,

has a doubleton solution set, of which one element is:

f(x, t) = (A)exp{2(pi)i[kx – (loweromega)t + phi]},

where:

i^2 = -1,

k = the wave number,

lambda = 2(pi) / k = the wave length,

loweromega = the angular frequency,

phi = the phase ,

and

v = loweromega / k = the wave speed.

When expressed as wave mechanics in the field of complex numbers,

e^iT(upperomega) = e^it(loweromega)

implies that

e^iu[S(upperomega) - sigma] = e^iu[(s(loweromega) - sigma]

We define the symbols in a case-sensitive manner, as follows:

The omegas are tempos, sometimes measured as a percent of 120 beats per minute. However, it is impossible to measure 0% with the software that the originator uses. The minimum percent is 10%. Therefore, the notation “Moderate” effectively reduces 120 beats per minute to only 108 beats per minute. And the same reduction of 12 beats per minute also needs to be made for all the other verbiages, depending on the software used by the server.

The constant, sigma, measured in kilobytes, is the storage size of the TitleSigmaNull.musicxml file. This file is a transcription of a score that is sung as just one whole note, the concert "A" note, held for 56 seconds of duration, corresponding to a tempo of 10%, or 20 beats per minute, the minimum tempo. The duration is 14 seconds per quarter note.

1 beat = 1 quarter note

sigma = 4 kilobytes.

S and s are storage sizes of musicxml files, measured in kilobytes. T and t are the corresponding durations, measured in minutes, as a decimal. For example, six seconds is equal to 0.1 minutes.

In order that exponents of e come out dimensionless, we define the unit constant, u, as:

u = 56 seconds per whole note

u = (56 seconds per 4 quarter notes)(1 beat per quarter note)

u = (14 seconds per beat) / (60 seconds per minute)

u = 0.233 minutes per beat

u = 0.932 minutes per whole note

(1) / (u) = 4.29 beats per minute

(1) / (u) = 1.093 whole notes per minute

(u) / (sigma) = 56 seconds per 4 kilobytes per beat

(u) / (sigma) = 0.933 minutes per 4 kilobytes per beat

(u) / (sigma) = (0.233 minutes per beat) per kilobyte = x / y

(sigma) / (u) = (x beats per minute) per y kilobytes

(sigma) / (u) = 4.29 beats per minute per 1 kilobyte

(sigma) / (u) = 4.29 beats per kilobyte per minute

(sigma) / (u) = 1.093 whole notes per kilobyte per minute

u = [(4 kilobytes)(56 seconds)] / [(60 seconds per minute)(4 beats per whole note)]

u = 0.933 kilobytes per whole notes per beats per seconds per minutes per second

u = 0.933 kilobytes per whole notes per beats per minute

u = 0.933 kilobytes per minute per 0.25 whole notes per beat

u = 3.732 (kilobytes per whole note) per (1 beat per minute)

The server revisits the 900 Solesmes scores upon their complete transcriptions into round notes, rests, and English lyrics. The server places the appropriate tempo for the quarter note after the header of each score, as a tempo marking, proportionately.

As an example, consider the ChristmasVigilOffertory.musicxml.

sigma = 4 kilobytes,

u = 0.933 minutes per kilobyte

that corresponds to :

u(sigma) = 4 beats per 56 seconds = 0.0714 beats per second = 4.29 beats per minute

S = 50 kilobytes

[S(loweromega) – (sigma)(loweromega)]u = [50(loweromega) – 4(loweromega)]u = [(46 kilobytes)(1 minute per kilobyte)] (loweromega) = (46 minutes)(loweromega)

For a Low Mass, we have:

T(loweromega) = (1.7 minutes)(loweromega) = t(loweromega).

We need to solve the proportion (after cross-multiplication):

{T(upperomega)]{u[(s(loweromega) – sigma]}

= [t(loweromega)]{u[S(upperomega) – sigma]}

The tempo that we seek is a factor of upperomega, the angular frequency.

It is determined by:

sigma / upperomega = S + (T/t)[(sigma / loweromega) - s].

However, (T/t) = 1. Therefore,

sigma / upperomega = (sigma / loweromega) + S – s.

= (4 kilobytes / loweromega) + 50 kilobytes – 4 kilobytes

= (4 kilobytes / loweromega) + 46 kilobytes.

Hence,

(4 kilobytes / upperomega) = (4 kilobytes / loweromega) + 46 kilobytes,

and

(1 / upperomega) = (1 / loweromega) + (46 kilobytes / 4 kilobytes),

and

(1 / upperomega) = (1 / loweromega) + 11.50

and

(1 / upperomega) = [1 / (20 beats per minute)] + 11.50

and

(1 / upperomega) = 0.05 minutes per beat + 11.50

and

(1 / upperomega) = 11.55

Hence, upperomega = 1 / 11.55 = 0.0866 beats per minute, an extremely slow tempo.
So I must have made a mistake somewhere.

The 0.05 is measured in minutes per beat, but the next term, 11.50, has no measurement.

Here is a workaround to the failed attempt to assign a Low Mass Offertory Tempo:

The brute way of thinking goes that

S / s = (1 / loweromega) / (1 / upperomega)(0.933 minutes / 1.7 minutes),

which yields

upperomega

= (50 kilobytes) (20 kilobytes / 4 kilobytes)(0.933 minutes / 1.7 minutes),

= (2.94)(50 kilobytes)

that is:

upperomega = 147 beats per minute, a Lively tempo.

An end-user who is accustomed to seeing "Moderate" tempo, ought to take notice of the speed-up.

Colloquially, the Offertory tempo is found by multiplying 2.94 times the number of kilobytes in the score, at a Low Mass.

The name, Low Mass Offertory Multiplier, is given to:

2.94 beats per minute per kilobyte.

The server puts the numerical Offertory tempo in the score.

The server makes another tempo marking, indicating Ambulatory, Moderate, or Lively after the header, as fits the tempo, and exports the tempo-adjusted musicxml file.

The Dynamics

There are at least 2 dynamics:

A) Add a sforzando accent to each Deity word grouping, at the appropriate syllable. For example, place sfz at the syllable "Ho-" in the Deity word grouping, "Holy Spirit."

B) Use decrescendo sliders with moderate frequency, to relieve what some worshippers decry as boredom in any type of chant.

This means that the computer server needs to understand the meanings of:

i) decrescendo placement,

ii) decrescendo duration,

iii) keeping the decrescendo contained on one line of the screen of a smart phone, held horizontally,

iv) a moderate frequency of decrescendos.

Perhaps a public opinion poll on decrescendo usage, measured in stars, with a randomly-chosen sample size of 32 worshippers in each of 32 Catholic parishes, will either allow for the statistical calculations needed by the server's software or allow for more artificial intelligence to be used.

1 = poor

2 = fair

3 = average

4 = good

5 = excellent.

Of course, there would be an optional open-ended quiz on the worshipper's retention of the Solesmes song's message, its call to action, and its call to accept God's grace. The originator does not know how to specify the server is to quantify such an optional quiz objectively.

But this would be a volunteer effort, also. And it would honor St.Gregory the Great. In Biblical work, the meanings of phrases are the most important concern.

And words carry emotions that are represented with dynamics in a score.

Quality software that can parse an English sentence ought to be currently available on GitHub.

Security:

Use Google products:

Use Google Drive, preferably without encryption, for communication,

and use Google Analytics, for measuring the “Unsubscribes.”

Use the General Data Protection Regulation and the similar California Consumer Privacy Act.

Cookies:

The end user needs to be presented with the following prompt:

Please subscribe to Google cookies, for keeping our religious out of jail.

Subscribe

Unsubscribe

Learn More

Here is the text for “Learn More”:

The reasons for our entreaty to you are the General Data Protection Regulation and the similar California Consumer Privacy Act.

And the Google cookies pertain to Google Analytics.

Your free choice of “Subscribe” sets up a legal contract with you, the end user.

The Day To Score app seems to be selling you something.

We apologize that Google Analytics will not be useful to you, but only to your team of humble developers.