

ステータス:	終了	開始日:	2013/03/10	
優先度:	通常	作業時間の記録:	0.00時間	
担当者:				
カテゴリ:				
対象バージョン:				
説明				
連符内連符および和音の逆アルペジオをサポート。				
関連するチケット:				
関連している Bug # 79: (V6.40)譜面モニタ音部領域のテンポ属性表示で%記号が2つ表示される場合がある			終了	2013/05/10 2013/05/11
関連している Bug # 80: (V6.40)初期化パラメータVGSをゼロにするとXG音源で音が鳴り止まない場合がある			終了	2013/05/11 2013/05/11

履歷

#1 - 2014/01/08 08:46 - Redmine Admin

; 《Ver6.4 開発後記》 2013.04.25

Muse開発の残課題を書き溜めた検討リストの中で私が最も実現したい、にも関わらず10年以上、手付かずのままになっており、そして技術的に最高レベルで困難を極めるだろうと考えていたテーマ。その開発は今までのMuseコンパイル処理と内部データ構造を一から見直す必要があるものであり、作業量としても膨大な規模になると予想していたテーマ。それは、・・・「連符内連符」です。正直言って、時間がたっぷり出来る定年後の老後の楽しみでやるしかないだろうと考えていたぐらいです。既にマクロが再帰的な階層構造で展開されており、しかも繰返し表記や連符、和音、コードなどの再現表記までが交錯する状態で、その和音や連符を、連符内の音長比率を継承して無限の階層に仕立て上げるなんて、本当に自分にその力量があるのかさえ疑うほどの高い壁でした。

2ヶ月ほど前に、とりあえず老後の備えとしてコンパイルの1パス目、分析関数を整備しておこうと考えたのが切っ掛けでした。無限階層の連符や和音を処理するには括弧記述に対するスタック構造が必要です。2パス目でこの構造の構築もやってみたくなりました。こうして徐々に深みに嵌まっていったのです。いや、高みに昇っていった、と言ったほうがいいのかな。その先にLilyPondの対応やらアルペジオやらアクセントやらゼロ音長やら、多くのトラップが待ち構えていることは薄々勘付いていたのですが、まあ行けるとこまで行ってみよう。收拾がつかなくなったらいつでも引き返せばいいんだ、という気持ちで突き進みました。開発を進めながら気づいた課題をメモっておき、それを順次潰していくのが私の開発スタイルなのですが、一時は1つの課題を潰すと5つの課題が見つかるといった凄まじい事態になりました。少しずつ山頂を目指して登っていくのですが、峰を越えるたびに新しい本当の山頂が現れるといった具合です。データ互換も確保しなければならないので、現存する8,300強のデータを逐一MIDIバイナリで出力し差異が無いか確認しながら慎重に進めました。開発トライアルリビジョンはなんと100を越えました。更に、これまた骨のある改造だと感じていた和音の逆アルペジオにまで食指を伸ばしてしまい、いわば地獄の階層構造です(苦笑)。

「もう登頂は断念して引き返えそう」そんな悪魔の声がいつも聞こえてました。新しい文法は従来の記述を無理なく拡張しているので記法としては何の問題も無いのですが、プログラム自体のデータ構造と解析シーケンスは大きく異なります。今まで連符と和音で1階層の括りしか許していなかったため、厳密な定義をしていなかった部分があり、それを明確にする過程で、どうしてもデリケートな部分

； の互換を崩さざるを得ないことになりました。アルペジオの連符内スケーリング
； に関する仕様は、この機会に積極的に変更させてもらいました。微妙な差異です
； が、8千曲中、数百のデータに非互換を与えてしまうことになり、本当に申し訳
； ありません。Museの理想的姿への脱皮のためにご理解頂き、どうかご容赦下さい。
； 開発の後半戦は、LilyPondへの対応でした。これまた凄まじい作業となりました。
； 当時、LilyPondの文法や制約もよく判っていなかった頃に四苦八苦しながら作り
； 込んだ部分なので、後付感満載のプログラム状態だったのです。今回コツを充分
； 理解していたので、ここもデータ構造も含めて一から作り直すことにしました。
； こんなことならLilyPond対応をする前に、連符内連符をやっつけておけば良かった(苦笑)。でも、ものは考えようで、お陰ですっかり垢が落ちました。トリルや
； トレモロの記譜能力も高まり、出力を断念するデータもかなり減ったと思います。
； 下手な制約条件の上に例外だらけのプログラムを組むより、制約を取り払い一段
； 高いレベルのデータ構造とシンプルなルールで処理を組んだ方が、格段に優れた
； ソフトウェアが完成することがよくわかりました。確かに抽象度が一段上がるの
； でデータ構造の設計には苦労しますが「練りぬかれた設計思想こそ、美しい作品
； を生み出す」という、常日頃考えているこの教訓を再認した次第です。複雑な和
； 音や連符、アクセントやアルペジオの階層定義、そしてゼロ音長や修飾音の扱い
； 等、苦労の連続でしたが、結果として処理性能を向上させ、更にメモリ効率も高
； めた形で登頂することが出来ました。そして驚くべきことに、文法表現能力が向
； 上しているにもかかわらず、実行モジュールはむしろ小さくなっているのです！
； Readme.txtは相変わらずですが(笑)。
； 前回バージョンアップでは外装であるUI系を刷新しました。今回は内部のエン
； ジンをオーバーホールし、その殆どを作り変えました。Museはこうして少しずつ
； 入れ替わり、多分一昔前のMuseは既に存在しなくなっていると思います。まるで
； 新陳代謝を繰り返している生命体のようです。もしかしたらこれこそ、Museが「生
； きている」証しなのかもしれません。そしてその原動力は、皆さんから頂く助言
； やデータ、障害対応などの数々のご支援であることは、疑う余地がありません。