Extending Music Notation as a Programming Language for Interactive Music

Juan Carlos Martinez
jcm7@gatech.edu
Alumni, Georgia Institute of Technology
Atlanta, USA

Figure 1: Score of the interactive piece "HOLA MUNDO! ..." [Hello World] for hybrid piano. © 2021 Juan Carlos Martinez

ABSTRACT

This work describes a novel approach for composing and performing interactive music by extending the traditional staff notation to the programming language domain. The proposed syntax aims to describe the interaction between humans and computers in live-electronics music performance. Thus, both performers and machines will understand this new notation, creating a cohesive music representation for performance that is both human-readable and technology-independent. This paper starts by describing some critical issues related to live-electronics that make it challenging to build repertoire around this genre. Next, the proposed approach is detailed, along with some syntax examples. Finally, the last section describes the evaluation of the proposed approach, including a description of the software implementation and a set of short interactive-pieces.

IMX '21, June 21–23, 2021, Virtual Event, NY, USA

ACM ISBN 978-1-4503-8389-9/21/06…$15.00
https://doi.org/10.1145/3452918.3458807

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; **Performing arts**; • **Software and its engineering** → *Domain-specific languages*.

KEYWORDS

interactive music; music notation; domain-specific languages; live electronics

ACM Reference Format:
Juan Carlos Martinez. 2021. Extending Music Notation as a Programming Language for Interactive Music. In *ACM International Conference on Interactive Media Experiences (IMX '21), June 21–23, 2021, Virtual Event, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3452918.3458807

1 INTRODUCTION

Common Western Music Notation (CWMN) is the cornerstone for preserving the musical works known as classics. These musical works are performed by musicians through an interpretation of the music that was encoded on a musical score using the CWMN notational system. CWMN, in essence, is a product of the Middle Ages; it reached its stability at the end of the sixteenth century. Some notational mechanisms were added after but without modifying its core notational basis [9].

CWMN plays a central role in the workflow from composition to performance in traditional Western music practices. First, the composer encodes a musical idea using staff notation. Next, the encoded information is interpreted by musicians that recreate the composer's idea based solely on the musical score. Additionally, the role of traditional music notation transcends the performance domain as it enables high-level abstractions by music theorists, which contribute to the development of musical techniques and styles.

With the technology rise during the twentieth century, a new music genre commonly known as Interactive Music emerged. In this musical genre, electronic sounds from some external inputs are modified in real-time. Among the different approaches to interactive music, one important sub-genre involves acoustic instruments that interact in real-time with a computer that reacts to human performance. This genre receives different names such as Live-Electronics, Real-time music, or *Musique Mixte* (i.e., Mixed Music), although the two first terms have a broader scope.

Musique Mixte has two components: the acoustic parts and the computational parts. However, only the acoustic parts are notated using CWMN, whereas the computational parts are encoded in a language specific to the technological framework that runs the interaction. This fact creates issues when reperforming the piece in the future as technology changes at a fast pace and becomes obsolete. This work analyzes this and other issues that arise in the interactive music's performance and proposes a new computational approach for composing and performing *Musique Mixte* that extends from traditional staff notation and addresses these issues. This work is an extension of my research on this field conducted at Georgia Institute of Technology during my doctoral studies [12].

2 BACKGROUND

2.1 Common Western Music Notation

This section describes some important notational features associated with traditional CWMN that are relevant to this work. CWMN has two inherent and different notational mechanisms, a graphic representation of pitch and duration and a verbal description regarding aspects such as tempo, mood, or articulation. The graphic notation is called diastematic, which means that pitch relationships are represented graphically on the vertical axis. The verbal annotations regarding tempo, loudness, and mood are named phonetic as it uses text in its representation [16].

In the diastematic domain, traditional staff notation excels in representing temporal relationships between events. Traditional music notation abstracts time and solely encodes rational relationships among time durations (e.g., quarter to the whole is 1:4). In this paper, this time representation is referred to as symbolic in contrast to absolute, which means that time is specified in seconds. This abstraction has shown to be very convenient for performance purposes as it allows synchronization while enhancing expressivity during the performance.

From the phonetic dimension standpoint, CWMN has used textual annotations mainly since the seventeenth century, and this phonetic dimension has grown in scope since then. Phonetic annotations are usually called expression marks (or performance marks). However, it is worth noting that this denomination is misleading as its scope is far beyond notating expressivity [15].

2.1.1 *Technology Independence.*

Traditional music notation has always kept its abstract connotation for both domains: diastematic and phonetic, achieving a notation that is independent of instrument technology. It is worth noting that there have been other notations in music history that are dependent on the instrument construction. For example, in the tablature-notation for fretted-string instruments the finger position is notated on the score. The fact that Western Music notation is technology-independent has played a fundamental role in building the music repertoire, for instance, by easing the re-performance of pieces written for instruments that went into disuse centuries ago [22]. For example, musical scores for viola-da-gamba dated from before 1800 still could be played nowadays by cellists. The technology-independence characteristic is very relevant to our discussion, as one crucial issue in interactive music related to the sustainability of interactive pieces.

2.2 Performance of Live Electronics

Although the repertoire of music that combines electronics with acoustic instruments (i.e., *Musique Mixte*) is very recent from the Western Music tradition standpoint (which is measured in centuries), reperforming any live electronics piece after its premiere has been found to be extremely difficult up to the point that is difficult to argue that exists an actual standard repertoire of live-electronics music [26]. Among the barriers that prevent the development of a true performance tradition in live-electronics are (1) technology obsolesce, (2) the need for the composer guidance to be able to reperform the piece, and (3) the complexity and particularities associated with the technology implementation [1] [10]. Technology obsolesce has kept many important contemporary composers away from this genre as technology becomes obsolete very fast; the risk of creating perishable music that does not create traditions around music practices is high in this genre [18]. The last two issues, the need for the composer guidance for recreating any performance and technology implementation complexities, prevent the piece preservation for the future and puts at risk the piece's premiere.

2.2.1 *Music Representations for Interactive Music.*

Developing a Music Representation for live electronics is fundamental to build repertoire around interactive music [20]. However, nowadays, current music representations of interactive music do not suffice for recreating the piece after the premiere as a significant part of the performance information is hidden inside the implementation, so porting that technology to a different technology setting is very hard. Additionally, performers need to understand how the human-computer interaction works to improve their interpretation and that information is usually not available for them [19].

Nowadays, the most used software application for interactive-music is MAX[1], a visual programming environment where users implement the interaction logic by connecting blocks (called patches). MAX follows the classic flow-based programming paradigm. The advantage of a visual programming environment is that coding has a fast learning curve for novice users, but the disadvantage is that a flow-diagram representation gets complex very quickly in a real

[1] https://cycling74.com/products/max

use-case. Additionally, in MAX, the priority of processing depends on the position of the objects, which could bring unexpected results if the blocks (patches) are moved. Moreover, MAX uses absolute time reference (i.e., seconds), but scored-music is based on symbolic references, and this fact brings additional complexities when implementing a computer interaction for *Musique Mixte*. For instance, in Risset's *Extensions* [18], the pianist should play an exact predefined tempo to be able to match the fixed timings in milliseconds used in the MAX patch.

Some musical software tools have tried to make the encoded performance logic more readable for musicians by mixing standard staff notation with their custom programming language syntax. For example, NoteAbilityPro [7] and Ascograph [2]. However, in these tools, the music representation of the performance information is still highly dependent on the technological platform that drives the digital instrument. For instance, the language syntax of NoteAbilityPro has a strong dependency on MAX, the popular visual-flow programming application, and Ascograph with Antescofo[2] , a score-following system where Ascograph acts as a music-oriented user interface. Ascograph interface [3] uses a piano roll user interface where each programming-event is located, the y-axis represents pitch (i.e., piano keyboard), and the x-axis represents symbolic time using a textual description (measure-beat). The Ascograph interface also has a text editor which contains the programming code instructions using Antescofo custom language.

In summary, besides music representation being human-readable, the syntax should be cohesive with the music-performance domain and technology-independent [3].

2.2.2 Existing approaches for reperforming and building repertoire of Musique Mixte. Nowadays, there are four different approaches that are usually combined to reperform a piece of *Musique Mixte*: preservation, emulation, migration, and virtualization [1]. Preservation implies keeping the original software and hardware as long as possible. Emulation is based on new technologies that recreate old ones, for example, software synthesizers that emulate old analog devices. Migration is the most used approach to reperform a live-electronics piece and consists of porting old technology to new updated versions. Finally, virtualization provides abstract documentation of the required performance information; this is the preferred approach as it eliminates the technology dependence. All the first three approaches, preservation, emulation, and migration, face difficulties as they are impacted by technology obsolesce and only provide a temporal solution [1]. All of them require accurate and complete technical documentation.

Additionally, emulation and migration require a good understanding of old and new technologies by computer music designers[4]. Virtualization is the most sustainable approach for documenting a piece. There are some successful examples documenting audio-signal processing chains using flow-diagrams developed by Andrew Gerzso for some of Pierre Boulez's interactive pieces [5]. However, virtualization implies a significant effort with strong collaboration between composers and archivists, which is not always possible as usually, composers do not spend time documenting the piece's electronic side.

Moreover, the accompanying documentation that enables reperforming live-electronics repertoire is often incomplete. There is no standardized approach to notate the required performance information to recreate a piece of *Musique Mixte*. Regarding this aspect, Lemouton and Bonardy wrote the following statement relevant to this research:

> As the electroacustic repertoire has now matured sufficiently to arrive at its 'Age of Reason,' we believe that the time has arrived to propose a documentation model equivalent to traditional musical notation. This documentation should be sustainable, because we have seen that for data preservation over time paper always seems to be a better solution than digital archives. The latter can not always guarantee physical integrity or content intelligibility [11].

2.2.3 Passive Role of Music Representation during Performance. Notating the piece's electronic side (i.e., performance logic) before the premiere is not an absolute prerequisite for playing the piece, and usually, interactive music composers move on to a new project after the premiere. Consequently, the information for reperforming the piece is usually missing, or at best incomplete [11]. However, there are notable exceptions, such as Stockhausen's works [8]. The typical case for electronic pieces is that only acoustic parts are accurately notated using traditional staff notation. In short, if the music representation that encodes the performance logic for the electronic parts does not play an active role during the performance, it is very likely that the composer will not spend time in this high time-consuming task even though this information would help performers to understand their role in the computer interaction during the premiere and would ease the performance of the piece in the future.

3 RE-THINKING THE MUSIC WORKFLOW FOR INTERACTIVE MUSIC

Gerzso writes, "If contemporary music has the ambition to prolong the classical music tradition, it must find the techniques, modalities and practices to ensure its survival despite the instability of digital technology" [5]. This work aims for a cohesive, complete, human-readable, and technology-independent representation of the music performance information for *Musique Mixte*. Additionally, suppose this ideal representation of interactive music is an absolute prerequisite to perform not only the acoustic parts but also the electronic parts. In that case, it will provide complete information to performers for understanding their role during the computer interaction for the premiere and future performances.

This work extends CWMN to the interactive domain by adding abstract-verbal statements on the score phonetic-dimension that enrich the accompanied graphic signs to develop a cohesive and human-readable representation of the performance logic of an interactive piece. Additionally, the newly added score annotations can be understood by a machine from the digital version of the score, so the musical score also becomes the source-code of the

[2]http://antescofo-doc.ircam.fr
[3]https://antescofo-doc.ircam.fr/UserGuide/deprecated_ascograph/
[4]the computer music designer is a position introduced at IRCAM institute which role is translating composers ideas into technical implementations.

piece's performance logic. Thus, easing the musical interpretation and playing an active role during the actual performance.

An interactive piece created with the proposed notation has a workflow from composition to performance that resembles traditional Western music practices. First, the composer writes a musical idea in staff notation using any engraving software (e.g., Finale[5] or Sibelius[6]). This musical score includes annotations in the phonetic domain that describes the logic of the computer interaction parts, allowing performers to understand their role in the computer interaction and theorists to identify music patterns as these annotations are human-readable, abstract, and technology-independent and most important presents complete information of the music performance. Moreover, a computer can also understand the score from its digital version and perform the computer interaction live during the concert. In summary, this work proposes the virtualization of the performance information solely based on the staff notation that is active during the performance. The output of the performance engine that interprets this digital score during a live performance is control signals that could drive an electronic instrument directly or could be routed to an external music software application.

3.1 Computational Counterpoint Marks

The proposed notation syntax was named *Computational Counterpoint Marks* as it extends counterpoint to the computational domain. Since the seventeenth century, the phonetic domain of musical notation has grown in scope, expression marks started notating tempo (e.g., adagio), then dynamic expressions were added (e.g., pp), and later on, articulation marks appeared (e.g., legato). As the proposed notation is encoding motivic transformations that computationally extends traditional counterpoint techniques (e.g., augmentation or diminution) to the interactive domain, in this work, these programming annotations in the score will be referred to as *Computational Counterpoint Marks*; sometimes this name will be abbreviated to just counterpoint-marks. *Counterpoint-marks* are human-readable annotations that function as expression-marks added to a musical score and accurately describe a performer-computer interaction's business logic. It is worth noting that the purpose of this extended staff-notation is to provide a language for the *Musique Mixte* genre that is sustainable, technology-independent, and human-readable but still able to be interpreted by a machine. Its syntax belongs to the CWMN domain; thus, it is not focused on representing all possible use cases in interactive music.

3.2 Conciliating Symbolic and Absolute Time

From the time domain standpoint, the proposed representation is based on traditional score notation, so time representation is symbolic and not absolute. This is a great advantage from the notation perspective as dealing with time representations in the context of Interactive Music is a constant complaint from composers that use music software tools as MAX which are based on absolute time references [25]. In our model for mapping symbolic time to absolute time, during the performance, the engine periodically receives an update about the current symbolic time and based on this information, the engine estimates the correspondent absolute time when the events and instructions should be executed. This input-clock could be generated from a score-following application or being human-triggered depending on the required time resolution between the human and the computer part; score-following is a system that estimates the current score location based on audio or MIDI[7] input from performance [13].

3.3 Control Message Information

When a digital score is executed during a live performance, the output is composed of control messages that drive digital instruments. This subsection details what data is carried by a single message. A traditional musical score records only (with different degrees of accuracy) for a single event: pitch, duration, and dynamics (i.e., loudness), and the time when the event should be triggered (in symbolic time). The proposed syntax operates in the musical-notation domain; thus, each control message carries the three data dimensions associated with a single event: pitch, duration, and dynamics. The symbolic time when a control message should be sent to a digital instrument could be notated directly on the score using standard notation (for simple statements) or could be generated as a result of the computer-interaction logic execution.

4 SYNTAX

This section presents an introduction to the *Counterpoint-marks* syntax. The description here does not intend to be exhaustive but an introduction to the reader about how the language interacts with traditional staff notation when describing a computer interaction. Most examples include a visual score picture; the top staves illustrate the actual notation while the bottom staff shows the statement's output. The output-staff is not part of the actual notation; it was added only for didactic purposes. Moreover, modeling complex computer interactions with the proposed notation produces outputs that cannot be represented graphically as the time of some output events cannot be represented by simple numeric rational relationships as occurs in traditional staff notation (e.g., 4:1 or 3:2).

4.1 Note Expressions

Note expressions are the simplest annotations and could be used for just sending messages at specific symbolic times for event-driven computer interactions. Figure 2 shows in the top staff an example of a note expression; this basic example will help explain the syntax basis. The first important fact to highlight is that *Counterpoint-marks* statements are notated as expression marks in the score, and they are executed from the note (or chord) that is vertically aligned on the same staff. Additionally, these expressions usually use jargon close to the music theory domain. For example, the constants OCTAVE and P8 refer to an interval a perfect octave above. Moreover, the statement is executed on every note-event on the score until a new statement appears on the same staff. In this particular example, there are two expressions on the top staff, the first one starts at the first-measure/first-beat and another on the third-measure/second-beat that replaces the previous one.

[5]https://www.finalemusic.com
[6]https://www.avid.com/sibelius

[7]Musical Instrument Digital Interface (MIDI) is a communication protocol between electronic instruments

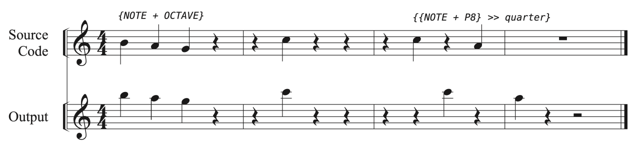

Figure 2: Sample score using *Note* expressions. The first two measures create an octave transposition and the last two measures create an octave transposition shifted by a quarter note.

4.2 Theme Expressions

In theme expressions, operands are musical themes (i.e., motives), so theme expressions perform motivic transformations. Theme expressions could be unary when only involves one theme or binary when two themes participate. Figure 3 shows an example of a unary theme expression. First, note that themes are notated by single upper-case letters between right-angle brackets which reflects the syntax used on contemporary music scores for highlighting motivic material in a musical score [24]. Additionally, note that this example uses the circular-shift operator. This operator performs a left circular shift of the pitch (i.e., default dimension) by the specified index. Moreover, *theme A* consists of the sequence of the following pitches: G-F-C-A-D, so a left circular-shift of size two will start at the third pitch (i.e., zero-indexed), obtaining the pitch sequence: C-A-D-G-F. As shown on the output-staff, each note's duration remains the same as the expression only modifies the pitch dimension.

Figure 3: Sample score using *Theme* expressions, The output is a copy of the motive with the pitches circular-shifted by 2.

One important fact to highlight is that the second staff is notated using the standard one-line staff as pitch information on the second staff is not used on this statement. However, it is worth noting that a statement could potentially use pitch information notated on the staff. Additionally, note that this example uses a symbolic character as the operator, in this case, a circular shift. This character is entered in the engraving software as Unicode text, which is supported in most notation systems and editors. Regarding *Counterpoint-marks* symbols and signs, whenever possible, a graphic sign related to the constructor semantic (e.g., circular-shift) was chosen to improve the statement's readability. Readability in programming languages syntax refers to the ability to be understood by the user [21]; readability in the programming language domain is closely related to the human-readable aspect in the music notation domain.

4.3 Event Expressions

Themes contain ordered single music-events; each event is a note or a chord. This section introduces event expressions which are surrounded by the Unicode double-parenthesis sign. Event expressions allow iterating among music-events (i.e., notes or chords) that belong to a theme. Moreover, the iteration is circular, so the next iteration goes back to the first one after reaching the last event. From a programming standpoint, event expressions behave as array iterators. For improving readability, the syntax includes a redundant empty square bracket after the theme (e.g., *A[]*), emulating the array syntax in general-purpose programming languages. Event operators include the most basic indexing-array operations. From a musical standpoint, event expressions allow mapping motivic material to more complex music-phrases with a time structure not related to the original theme.

Additionally, an event expression allows note-by-note transformations, which are very useful for describing gradual processes (e.g., a crescendo in the dynamics dimension). Figure 4 shows an example of an event expression; index-values were added below each staff to highlight how the iteration maps from the theme to the output phrase. The transformation performs an octave transposition of each event that belongs to *theme A*.

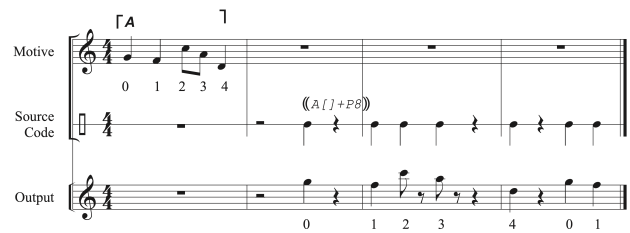

Figure 4: Sample score using *Event* expressions. The output on the bottom staff is the motive mapped in time to the note-durations of the middle staff.

4.4 Branch Expressions

In branch expressions, the program jumps to a different location on the score based on a performance condition. In other words, the computer jumps to the same measure that the performer jumps to. Figure 5 shows an example of a branch expression; in this example, the output is not added as it is non-deterministic; it is based solely on human input performance. A branch expression is notated through the *dal-segno*[8] music notation mechanism, although the semantics is reinterpreted computationally. In traditional Western Music notation, the *dal segno* (i.e., D.S.) textual annotation instructs the performer to jump to the measure where the *segno*[9] graphic character appears. In the proposed syntax, each *segno* graphic symbol has an accompanying numeric tag (e.g.,1 or 2), and the *dal segno* textual annotation (i.e., D.S.) has an accompaniment array that indicates the possible measures where the performer could jump from there. In computational terms, the *segno* graphic symbol acts as a

[8]Dal-segno in English translates "to the sign"
[9]Segno in English translates sign.

label of a program section, and the *dal segno* textual annotation acts as a jump instruction.

Moreover, the control tag (*@control*) indicates that the particular staff is a human-played input, and based on that performance input, the program will jump to a specific location. For example, at the third measure where the 'D.S. [1,2]' phonetic annotation is located, if the pianist plays next C4 and E4, the program will jump to the second measure, but if instead, the pianist plays C4 and A4, the program will jump to the first measure. It is worth noting that if the condition is not meet for any of the tags, the program will continue to the following measure, in this case, the fourth measure.

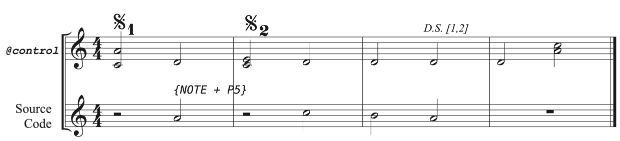

Figure 5: Sample score using *Branch* expressions, The *Dal Segno* (D.S.) marks in the top staff allows the instruction control-flow to move between non-sequential measures based on performer input.

5 WESTERN MUSIC NOTATION FOR INTERACTIVE-MUSIC

The proposed approach supports interactivity as the annotated score allows coordinating the live performance between the performers and the computer. This notion is explained in further detail with the following example using a transcription of Steve Reich's Piano Phase [17]. The original piece is for two pianos, and this transcription is for a duet between a human performer and a computer playing on the same (digital) piano. This transcription uses the proposed approach; it is worth mentioning that the interaction between the human performer and the computer is notated accurately by just adding a few programming annotations in the score, as shown in Figure 6. In contrast, the original score provides a textual description in the accompanying documentation to express the interaction between the two pianists, as the interaction between the two pianists can not be notated using traditional staff notation. The original performance notes begin with the following text:

> The first performer starts at bar 1 and, after about 4 repeats, the second gradually fades in, in unison, at bar 2. About 12 repeats getting into a comfortable and stable unison, the second performer gradually increases his or her tempo very slightly and begins to move very slowly ahead of the first until, after about 4 repeats, he or she is one sixteenth note ahead, as shown at bar 3 ...[17].

In the transcribed score (Figure 6), the top part is played by the pianist and is notated using traditional staff notation. In contrast, the bottom part is played by the computer and is notated using the proposed syntax. The segment highlighted in yellow shows an interaction between the computer and the pianist; the syntax of the *Counterpoint-marks* annotations is explained in the next paragraph. In the interaction, the computer slowly compresses the motive, whereas the pianist keeps playing the same motive (*theme A*). This phasing effect between the two parts is shown in the stem plot in Figure 7. In the figure, the x-axis represents time, black stems represent the piano part, and blue stems represent the computer part; note how the computer part is gradually played faster until it again matches the black stem at the end.

The segment highlighted in Figure 6 starts at the last measure of the first system and contains two statements: *{{X**0.992}->X}* that is above the staff and *{A->X}* that is below the staff, *theme A* is declared at the beginning of the system, and it is the twelve-note musical motif on the top stave. The statements' precedence is based on their position in the staff from bottom to top. Thus, in this case, the statement *{A->X}* is computed first, so *theme A* is copied into *X*. Next, the instruction *{{X**0.992}->X}* is computed. This last statement is surrounded by a music repeat-bar sign (i.e., ||: ... :||), which means that it is repeated sequentially until a music breath mark (i.e., comma) appears. Repeat sequentially implies that the same instruction is computed again after the last note of the previous output. The expression *{{X**0.992}->X}* uses the time compression operator, which is notated by the double star (i.e., **). Thus, theme *X* is compressed by a factor of 0.992 every time, and the output of this compression is reassigned to *X*. In other words, the variable *X* is re-written with each execution, creating what in a programming language is named a recursive-call but translated into the musical domain.

In summary, the highlighted segment in Figure 6 creates a sequence of repetitions of *theme A* where the theme is being slightly compressed every time by a factor of 0.992. The numerical factor of 0.992 was chosen to achieve that the last note of the compressed theme (before the repetition ends) is one-sixteenth (in symbolic duration units) ahead of the human-played counterpart.

This brief example shows how few *Counterpoint-marks* annotations, along with a reinterpretation of traditional staff notation in a computational way, achieves an augmented piano performance by describing a complex computer interaction that is both human-readable and can be understood by a machine.

6 EVALUATION

The proposed approach's evaluation included: first, a software implementation of an engine that executes embedded *Counterpoint-marks* annotations on the musical score during a live performance. Second, a small user study to evaluate syntax readability among musicians familiarized with traditional western music notation and, lastly, the composition and performance of several pieces supported by the proposed notation. Additionally, the piece scores were discussed with two professional musicians well-versed in interactive music composition and performance to obtain qualitative insights.

6.1 Software Implementation

This section explains the software implementation of the proposed approach. The prototype was able to parse and run a digital version of a musical score with *Counterpoint-marks* annotations during a live performance. The digital score format used for this implementation was MusicXML [6], which is supported by most engraving (i.e, music notation) software applications. Additionally, the communication protocol between components is Open Sound Control [27],

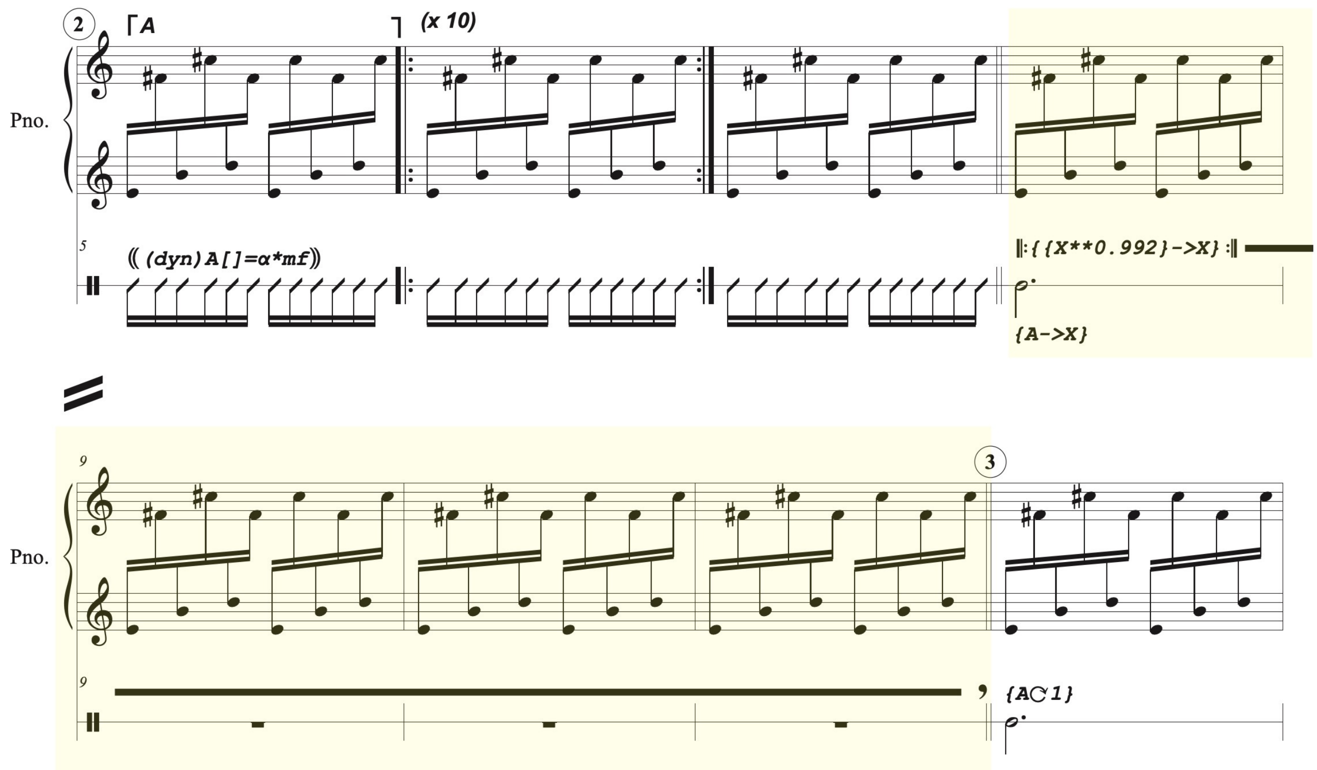

Figure 6: Score segment of an interactive transcription of Steve Reich's Piano Phase piece for solo piano. Computer interactions are notated using *Counterpoint-marks* annotations. © 2021 Juan Carlos Martinez

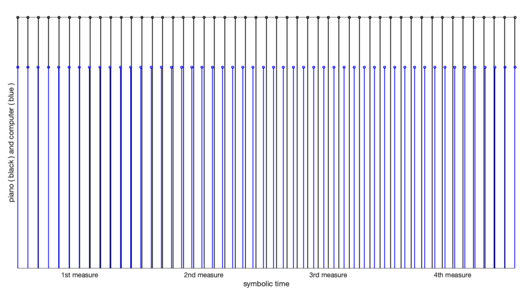

Figure 7: Timing relationships between the human-played piano part and the computer part. Black stems represents the piano events and blue stems represents computer events played in the same digital piano.

which allows communication with a wide range of music systems. Figure 8 shows the basic components and how they are related. The implemented prototype consists of three components: (1) A score-compiler that translates the MusicXML digital representation to a performance-efficient intermediate representation, (2) a performance engine that runs the intermediate representation in real-time during the concert, and (3) a MIDI host that translates the engine output control messages into MIDI messages and routes them to the corresponding digital instrument. Additionally, the MIDI host implementation embeds a basic score-following algorithm that estimates the current location in the score based on the human playback; this information is sent periodically to the performance engine.

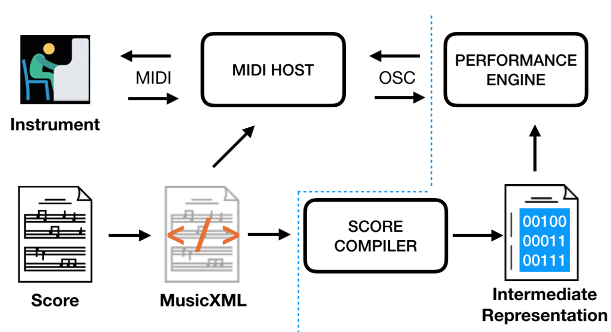

Figure 8: Architecture of the Software Implementation that supported the live-performance of the interactive pieces for Digital Piano that were notated using the proposed extension to Western Music Notation.

6.1.1 Performance Latency. A set of short, interactive pieces for digital piano were composed using *Counterpoint-marks* annotations and performed using the implemented engine; the latency was calculated from the performance data. According to Pennycook [14], in music performance, control information is assumed to change at a rate of around 50 milliseconds. Thus, any introduced latency in the control domain should be below this threshold to be used in a music performance setting. Additionally, some studies suggest that the delay time introduced by any technology framework should be ideally less than 14 ms [4]. In this work, the maximum latency introduced by the software implementation was four milliseconds (ms). Thus, the prototype engine's latency is below the maximum values recommended for music performance, making the proposed approach suitable for interactive music applications.

6.2 Syntax User Study

A small online user study was conducted among five college students to measure how the proposed syntax was understood by novice users familiar with Western Music Notation. The inclusion criterion was the subject understanding of traditional staff notation. This criterion was validated with an initial practice question. The online user study included a brief explanation about the notation at the beginning; then, the subject filled out a blank staff with the expected output of the *Counterpoint-marks.*

The first three questions aim to evaluate the syntax core: theme expressions, chained expressions, and event expressions. The subjects had no previous knowledge regarding the proposed notation. Results in Table 1 show that the subjects understood the core notation just from a brief introduction to the syntax. These findings match the qualitative results from the interviews detailed in section 6.3. It is worth noting that we cannot argue these findings categorically as the number of subjects was small.

Table 1: Correct answers among subjects grouped by expression syntax

Syntax Core	Percentage
Theme Expressions	100 %
Chained Expressions	80 %
Event Expressions	80 %

Additionally, the last question presented the subject with a new advanced programming construct that was not explained in the introduction. Thus, it was expected that nobody would have this question completely correct; this question just intended to measure to what degree novice users could infer the semantic just from the syntax. In short, the evaluation wanted to measure the completeness of the answer. However, two subjects inferred entirely the new construct's semantic and wrote the expected output correctly.

6.3 Composition and Performance

Additionally, a qualitative evaluation of the proposed workflow was conducted by composing, performing, and discussing a set of short-interactive pieces for digital piano. These pieces were created using standard staff notation enriched with *Counterpoint-marks* annotations. From a musical standpoint, this set of pieces explores different counterpoint textures between humans and computers. Both parts are played on the same digital piano, achieving an augmented version of the human performance. Some qualitative insights were obtained from two interviews conducted with two professional musicians well-versed with live-electronics music: the pianist Lisa Leong[10] and the composer Jorge Variego[11]. During the interviews, the scores were discussed where only a brief introduction about the proposed syntax was given, and each interviewee read the scores individually before the discussion. One important result to highlight was that both musicians accurately described the computer interaction from the score for all the discussed segments. This result suggests that the proposed notation for interactive music could potentially ease sharing and understanding of existing works from other authors. This latter fact is a fundamental design key of computational tools for arts according to Shneiderman [23] and, of course, one of the pillars of building musical knowledge that allows abstracting music practices as mentioned in the background section.

6.4 Advantages of the Proposed Music Representation

6.4.1 A Virtualization that is active during the performance. One fundamental advantage of the *Counterpoint Marks* approach is achieving technology independence through virtualization. As mentioned in the background section, virtualization signifies providing abstract documentation for performance. The proposed virtualization has a novel element: the abstract documentation (i.e., musical score) plays an active role during the performance as the musical score is also the source code used by the computer that performs the interaction. This fact simplifies the composition workflow and forces the composer to document the required performance documentation during the composition stage before any rehearsal/concert.

6.4.2 Cohesive Representation for Musique Mixte . Another important advantage of the proposed approach is that the performance documentation is based on an extension of traditional staff notation. Consequently, the music representation is cohesive with respect to acoustic parts, and additionally, it extends from a language syntax already known by musicians.

7 CONCLUSION

This work proposes a workflow for composing and performing interactive music based solely on staff notation to achieve a cohesive and complete source of performance information for both human-interpreters and machines. The proposed approach resembles the traditional approach based on staff notation, although reinterpreted computationally. The new workflow is based on extending traditional Western Music notation by adding programming statements as expression marks in the musical score. Thus, describing computer interactions and creating musical scores that can be understood by musicians and machines with the advantage of achieving a

[10] https://centerfornewmusic.com/event/lisa-leong-pianist-arrell-boulez-harvey-lin-nono-williams/
[11] https://jorgevariego.com/

technology-independent and human-readable music representation for performance in the context of live electronics.

REFERENCES

[1] Alain Bonardi and Jérome Barthélemy. 2008. The preservation, emulation, migration, and virtualization of live electronics for performing arts: An overview of musical and technical issues. *JOCCH* 1 (06 2008). https://doi.org/10.1145/1367080.1367086

[2] Grigore Burloiu, Arshia Cont, and Clement Poncelet. 2016. A Visual Framework For Dynamic Mixed Music Notation. *Journal of New Music Research* 46 (11 2016), 1–20. https://doi.org/10.1080/09298215.2016.1245345

[3] Donald Byrd and Eric Isaacson. 2003. A Music Representation Requirement Specification for Academia. *Lem Music Journal - COMPUT MUSIC J* 27 (12 2003), 43–57. https://doi.org/10.1162/014892603322730497

[4] Chris Chafe, Michael Gurevich, Grace Leslie, and Sean Tyan. 2004. Effect of time delay on ensemble accuracy. In *Proceedings of the international symposium on musical acoustics*, Vol. 31. ISMA Nara, Musical Acoustics Research Group, The Acoustical Society of Japan, Nara, Japan, 46.

[5] Andrew Gerzso. 2015. The Longevity of Musical Works for Instruments and Electronic Music in the Digital Era. *Cahier Louis-Lumière* 9 (2015), 29–34. https://doi.org/10.3406/cllum.2015.962

[6] Michael Good. 2001. MusicXML for notation and analysis. *The Virtual Score: Representation, Retrieval, Restoration* 12 (01 2001).

[7] Keith Hamel. 2006. Integrated Interactive Music Performance Environment. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression*. IRCAM, IRCAM — Centre Pompidou, Paris, 380–383.

[8] Thom Holmes. 2012. *Electronic and experimental music: technology, music, and culture*. Routledge, New York.

[9] Thomas Forrest Kelly. 2014. *Capturing music: The story of notation*. WW Norton & Company, New York.

[10] Serge Lemouton. 2016. Computer Music Interpretation In Practice, In International Computer Music Conference Proceedings. *International Computer Music Conference* 2016.

[11] Serge Lemouton, Alain Bonardi, Laurent Pottier, and Jacques Warnier. 2019. On the Documentation of Electronic Music. *Computer Music Journal* 42 (05 2019), 41–58. https://doi.org/10.1162/comj_a_00486

[12] Juan Carlos Martinez Nieto. 2018. *The algorithmic score language: Extending common western music notation for representing logical behaviors*. Ph.D. Dissertation. Georgia Institute of Technology.

[13] Nicola Orio, Serge Lemouton, and Diemo Schwarz. 2003. Score Following: State of the Art and New Developments. In *Proceedings of the New Interfaces for Musical Expression (NIME)*. McGill University, Montreal , Canada.

[14] Bruce Pennycook. 1985. Computer-Music Interfaces: A Survey. *ACM Comput. Surv.* 17 (06 1985), 267–289. https://doi.org/10.1145/4468.4470

[15] Don Michael Randel (Ed.). 2003. *The Harvard dictionary of music*. Harvard University Press, Cambridge, Massachusetts.

[16] Richard Rastall. 1983. *The Notation of Western Music: An Introduction*. London: JM Dent & Sons, New York.

[17] Steve Reich. 1980. *Piano Phase:(1967)*. Universal Ed., London.

[18] Jean-Claude Risset. 1999. Composing in real-time? *Contemporary Music Review* 18 (01 1999), 31–39. https://doi.org/10.1080/07494469900640331

[19] Margaret Schedel. 1999. The Notation of Interactive Music: Limitations and Solutions, In International Computer Music Conference Proceedings. *International Computer Music Conference* 1999.

[20] Margaret Schedel and Federica Bressan. 2019. Notation for an Electric Stage: Twenty Years of Writing about Notation and a Thought Experiment. With additional commentary about Preservation. *Musica/Tecnologia* 13, 1 (12 2019), 85–101. https://doi.org/10.13128/music_tec-11164

[21] Robert W Sebesta. 2012. *Concepts of programming languages* (10th. ed.). Pearson, Upper Saddle River, New Jersey.

[22] Eleanor Selfridge-Field (Ed.). 1997. *Beyond MIDI: The Handbook of Musical Codes*. The MIT Press, Cambridge, Massachusetts.

[23] Ben Shneiderman. 2007. Creativity support tools - Accelerating discovery and innovation. *Commun. ACM* 50 (12 2007), 20–32. https://doi.org/10.1145/1323688.1323689

[24] Kurt Stone. 1980. *Music notation in the twentieth century: a practical guidebook*. WW Norton & Company, New York.

[25] Mauricio Toro. 2018. Current trends and future research directions for interactive music. *Journal of Theoretical and Applied Information Technology* 96 (08 2018), 5569–5606.

[26] David Wetzel. 2006. A model for the conservation of interactive electroacoustic repertoire: Analysis, reconstruction, and performance in the face of technological obsolescence. *Organised Sound* 11 (12 2006). https://doi.org/10.1017/S1355771806001555

[27] Matthew Wright and Adrian Freed. 1997. Open SoundControl: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference*. International Computer Music Association, Thessaloniki, Greece.