Αρμονία (Harmonia): A System for Collaborative Music Composition

Mark Goldstein, David Wihl
Harvard University

{*markgoldstein,davidwihl*}*@g.harvard.edu*

Abstract

Increasing productivity of music composition has many positive benefits. Listeners would appreciate music individually tailored to their emotional needs and context. Composers would be facilitated by greater and more diverse cooperation yielding more innovative music. Composition agents could assist in the generation of repetitive or experimental musical forms. Therapists can use music as part of a treatment plan for autism and many other disorders. The system we propose attempts to address these myriad needs by offering two key innovations: a SharedPlan with versioning to mediate the workflow of a composition for a group of musicians and an algorithmic evaluation of a composition against the intention of the SharedPlan to provide guidance to both human and agent composers.

Introduction

Music composition is often an individual endeavor, which is counterintuitive when music is mostly performed, improvised, and experienced in a group. Part of this is due to the singular nature of creative expression, but a large part is also due to a dearth of viable tools to enable collaborative efforts between composers. Most modern composition is facilitated through the use of digital audio workstation (DAW) tools, yet composers do not generally have access to tools designed for collaboration, version control, and annotation that are available to software engineering teams.

There are many additional issues that surround collaboration over structured, shared objects such as a music composition. Each composer has a unique artistic vision, and a composer working on one section of a piece may ruin the plans of another working on a later section. It is crucial in such settings that a group specify an artistic goal clearly and communicate their intentions for material as it is added. However, it is also valuable that new ideas can surface as a consequence of the collaborative process, leading to material that could not be created by any one artist. How can composers stay connected with an original goal and also make room for spontaneity?

Collaborative Ideation (CI) investigates the effect of various idea-sharing tactics on the creativity, productivity and scope of individual idea-generating participants. Such techniques may include showing each participant in a pool of brainstormers the collection of all participants' brainstormed ideas. In CI that focuses on the creation of shared objects such as a composition or story by several people, certain interfaces for collaboration, automated idea-sharing mechanisms, and per-participant system feedback may help facilitate group creativity and productivity.

SharedPlans provide a framework to establish and maintain context and metadata about group artistic work. This framework ensures that the collaborative process and ideation stay within reasonably agreed parameters, provides a formalized notion of common beliefs and defines a means of independent measurement that the work-in-progress is navigating toward the original intention.

We propose Harmonia, a system that addresses collaborative music composition inspired by tools used for software teamwork, collaborative ideation, and musical analysis. The system defines a set of editing procedures and an interface for communication and collaboration over a shared musical artifact.

In §1 we examine previous work related to shared intentionality, collaboration and group ideation, intelligent music systems, and modeling the structure of music. In §2, we describe the details of the overall usage workflow, as well as the individual components of our system. A proposed user interface for users and composers is described in §3. The details of the automated analysis of the work-in-progress is described in detail in §4. We present four use cases for our system in §5, where collaborating composers create music ranging in purpose from casual listening to clinical therapy. In §6, we discuss limitations that we have identified in our system, and a few initial plans for system validation. We discuss potential future work in §7.

1. Related Work

Our work builds on several areas of research related to music, computer science, and creative cognition. First, we discuss related work regarding shared intentionality in multi-agent settings. Then we discuss collaborative ideation, both in general and specifically in music. Next we discuss intelligent music systems that facilitate human composition and improvisation, and related work in Music Information Retrieval (MIR). Finally, we describe previous work that applies information theory to the analysis of musical structure.

1.1. Shared Plans

SharedPlans (Grosz and Kraus, 1996; Hunsberger, 1998) define a robust framework for multi-agent, multi-level collaboration. A SharedPlan consists of a set of intentions, mutual beliefs, constraints, and hierarchies of actions and plans. All of these elements are applicable in this context, although only a subset of the general definition is needed for music composition purposes.

A SharedPlan is the basic unit of work for a collaborative group in our system. It serves as the container for the initial definition of the intent of the work, the musical score representing the work-in-progress, the revision history, the subplans used to communicate between composers, and the final approval given by the user who defined the SharedPlan initially.

Intentions are expressed by a minimum of two criteria: genre and mood. Mutual beliefs are expressed implicitly in the trust of the automated analysis system that guides the collaborative process (See §4) and explicitly through sub-goals. There is one complex action that characterizes the full SharedPlan in this scenario: production of a score that meets the intention. Each composer may divide the complex action into a series of sub-actions (See §2.4 MIDI and Edit Actions). If a composer is unable to complete all actions necessary to achieve the intention, a subplan is created stating the Intention-To for a future composer. This is the primary means by which composers communicate with each other. When a subgroup of composers is working on a subplan, they share a restricted set of mutual beliefs about the intention of the subplan. We do not define a means of using *recipes* in this context.

We use the term composer in the singular form, however a single composer may work as part of several collaborating humans to produce an edit. Any such informal communication is beyond the scope of our proposal. After collaborating, a single human would execute edits on composition toward the intended goal. For example, a composer may improvise with colleagues to explore several alternatives of a subplan. Once a consensus is achieved that the section meets artistic and SharedPlan goals, one composer commits an edit.

1.2. Collaborative Ideation

Collaborative Ideation (CI) improves the productivity of individuals and groups in generating ideas through collaboration. People intend to create related objects (e.g., brainstorm solutions to social problems) and seek either feedback or examples of others' work to enhance their individual process. Collaboration is centered in a shared workspace, physical or virtual, that allows for communication and sharing of ideas. The ideas produced may be for individual use, or ideators may work on shared artifacts such as an essay or piece of art. The dynamics of collaboration may be real-time or not, though increasingly, today's settings are real-time and virtual [1][2][3] (Harhoff and Lakhani, 2016). A simple CI setting is one where each ideator brainstorms solutions to a problem common to all participants, and each participant can see all other's ideas. The design of intelligent computer systems today aims to facilitate these activities to allow for increased creativity and productivity.

Only a small subset of an idea pool may be relevant and inspiring to a single ideator, and it is overwhelming for each ideator to view all participant's ideas (Siangliulue et al., 2015). Ideahound (Siangliulue et al., 2016) prompts each user to interact with a personal "whiteboard" where they can cluster their ideas and separate them by semantic distance. A global semantic map is computed by compiling the whiteboards, allowing each ideator to view their work in the context of the entire solution space. IdeaHound recommends diverse suggestions to each ideator, eliminating the cognitive load of idea search.

When exposed to the work of others, composers may come up with things they would not make on their own. In our work, CI is made explicit when a composer addresses a subplan instantiated by another (See §2.3). Generated objects are structured, rather than unordered collections of ideas, and ideators

[1] `https://www.openideo.com/`
[2] `http://www.ideastorm.com/`
[3] `https://www.quirky.com/`

need to build over each other's ideas rather than only seek inspiration from examples by others. These details present new challenges.

CI has surfaced in the space of online blogs and services designed for sharing visual art and music. Ideas range from small, unfinished efforts seeking direction to finished pieces seeking critique. Artists improve upon their ideas using large-scale feedback. SoundCloud is a hybrid music streaming service and CI platform. Though music is often presented in finished form, people also post incomplete projects. Artists sometimes share "stems" to their music, which are individual sound files that feature isolated instrumental tracks, with the intention that others seeking inspiration remix their pieces into new work.

The Blend platform[4] makes the sharing of source files explicit. By default, artists share their works-in-progress in the format of music production software source files. This setting encourages building on each other's ideas. However, people mostly work toward an ultimately individual direction. What changes when several ideators intend to create a shared piece? In our work, a composition is associated with a specified intent through the duration of its existence. It is up to the composers and the system to keep a piece of music close to its SharedPlan.

1.3. Computer Facilitated Composition and Improvisation

Intelligent Music Systems ranging from composition and improvisation facilitating agents to recommendation systems have a common requirement: to "understand" music at multiple levels, including low-level acoustic signal, mid-level theoretical constructs such as harmony and rhythm, and high-level level aspects such as mood, genre, and style. For example, music recommendation systems such as Spotify [5] seek to analyze music and extract a measure of relevance for a function such as "study music." These issues constitute the research area of Music Information Retrieval (MIR).

Human composers are assisted, not replaced, by creative agents. A composer may have good "seed" ideas, but the system may recommend variations or re-orderings of ideas to make them more conveying. Such system knowledge often comes from large-scale corpus analysis that mines patterns common to a collection of music. ChordRipple (Huang et al., 2016) takes as input a progression of musical chords from a composer, and suggests substitutions of intermediate chords that preserve the original semantics of the input while serving to replace conventional choices with more interesting ones. If the composer agrees to make one of the recommended changes, the system assists the composer in interpolating between original and substituted material, resulting in a mix of human and system generated music. Our system makes suggestions to composers, but it focuses on structural changes to support collaboration rather than substitutions of material.

While the current system seeks to assist teams of human composers to enrich and organize their work, our design permits automated agent composers to fully participate without requiring human intervention. Google Brain's Magenta project explores the limits of machine creativity in art and music[6]. Magenta is an integrated environment of software tools and music-related datasets. Built on Magenta, AI Duet[7], reacts to human improvised gestures. Improvisation is an important part of composition. When a single composer is brainstorming, it may be beneficial to improvise on ideas with an automated agent, much like two musicians iteratively vary and refine their ideas in live rehearsals. In settings where a piece is defined by a specific enough set of guidelines (See §5.3), a listener may need music at a certain tempo and with a simple beat. Powerful information retrieval systems may make effective machine composition agents possible. Human composers may be placed at later steps of collaboration to ensure that the piece meets requirements in humanly perceptible ways.

1.4. Information Theory and Music Analysis

Our systems relies on the ability to analyze musical structure to support automated feedback for collaborating composers. Recent approaches have represented musical form in the context of listener perception, modeling the attention dynamics of the listener as they experience the toggle of surprise and redundancy. The Information Dynamics Approach (Abdallah et al., 2012) uses *predictive information rate*, an entropy-based measurement of how a listener's distribution over future musical events is continually revised as new information is presented. Our work assumes that musical structure can be effectively summarized by this criteria. We assume that pieces from a particular combination of genres and moods are defined by characteristic balances of surprise and redundancy over time, with peaks of information content communicated by the

[4] https://blend.io
[5] https://www.spotify.com
[6] https://magenta.tensorflow.org/welcome-to-magenta
[7] https://aiexperiments.withgoogle.com/ai-duet

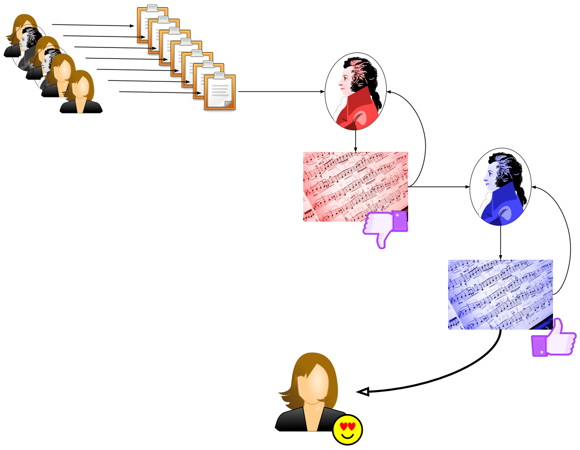

Figure 1: Overall workflow: a non-musical user or a composer define a SharedPlan. Composer 1 (red) starts iterating and commits a work-in-progress. Composer 2 (blue) continues iterating until the original user is satisfied.

composer(s) in genre-specific locations. We use the Information Dynamics Approach for automated analysis in our system, which compares a collaborative work-in-progress to the characteristic curves for the genre and mood specified by the work's SharedPlan.

2. System Design

The system design addresses two aspects of collaborative composition. The first includes the composition interface, the representation of music, the editing capabilities of composers, and the version control system used to track musical revisions. These all focus on issues of communication and coordination in collaboration over a shared artifact. The second aspect of design is the automated music analysis used for giving feedback and suggestions to composers. This facilitates the collaborative composition process by helping composers assess the effect of their edits on the movement of the work-in-progress toward a goal. We discuss how these two system components combine to help composers by providing an interface that makes revisions convenient, modular and amenable to collaboration.

2.1. Workflow Overview

The generic use case that we consider is that a listener (musician or not) requests a new piece of music of a specified genre and mood, and other related keywords. A new, blank score is associated with a SharedPlan that stores the genre and mood. Additional descriptive keywords may be included in the form of tags, like those on YouTube or Soundcloud [8] (`#USA`, `#lo-fi`, `#chill`, `#electronic`, `#120`*BPM*). These tags power an automated comparison between work-in-progress and goal, and helps composers pick edits that may bring a composition closer to a sound characterized by the goal keywords. We assume that an information retrieval system for querying MIDI music scores by keyword exists.

The system computes a summary of characteristic musical structure of the retrieved body of work. Composers iteratively add and edit material while receiving system feedback, helping composers choose which edits are most relevant to the SharedPlan. The system also suggests edits, such as to repeat, delete, or the switch the order of blocks of material.

Composers can follow suggestions or make their own decisions. During this process, composers create subgoals that describe the intentions of local edits, such that others can pick up on their material and preserve these intentions. Composers collaborate to bring the composition to a state of completion such that it matches the originally specified goals and the SharedPlan originator is satisfied. The finished product is a MIDI score.

2.2. Version Control

Version control has become an essential part of team-based software development. However, the tools for managing revisions of other creative work, including composition, are nonexistent or limited in comparison. There are two commercial version control systems for music: Blend and Splice[9]. `git` (Torvalds and Hamano, 2010) has been referenced elsewhere (Oberholtzer, 2015) as a viable means for music score version control, especially in the MIDI format. Software based version control systems are excellent for basic operations such as branching, committing and viewing history, but they lack several primitives necessary for integration in our proposed workflow. We propose the following additional commands:

For users and composers:

EditPlan - create, revise or delete a SharedPlan that includes intentionality and other metadata.

Approve - the SharedPlan creator denotes a SharedPlan as satisfactorily executed

Reject - the SharedPlan creator denotes a SharedPlan as unsatisfactorily executed

[8] `https://on.soundcloud.com/creator-guide/tracks`
[9] `https://splice.com/`

Note that Approval or Rejection by a user of a composition may be implicit based on their listening behavior (See §3.1 SharedPlan Workflow User Interface).

For composers only:

Evaluate - prior to committing a revision, perform an analysis of the current score against intentionality.

EditSubplan - create, revise or delete a subplan. A composer defines a subplan to fragment the SharedPlan into a tractable unit of work or express the call for assistance for another composer to provide input.

Release - after a commit, mark a work as ready for review by the creator of the subplan. This closes the subplan. If a plan is rejected, it may be re-opened.

Versioning is particularly relevant in the music context, as intellectual property and originality is often the source of extensive lawsuits. By maintaining version control, an examination of the history would assist in the identification of derivative works and originality of authorship.

2.3. Communication through SharedPlan and Subplans

When a composition task is created, it exists as a SharedPlan that contains an intent specification, as well as a `NULL` score. In addition to describing global goals, the SharedPlan mediates any inter-composer communication that takes place aside from the edits to the composition. A composer may wish to express the intention of a particular edit to justify its presence or to elicit specific future directions for their material. For example, a composer may add a block of music containing an exposed melody, but may also want to communicate that another composer should add a harmonic accompaniment. This is done by issuing a subgoal. Another composer who acts on an open subgoal *releases* it when they commit their work.

It is ideal for communication to be concise and structured. It may be detrimental to collaborative work for one composer to expect others to read long, unstructured goal descriptions. It may require too high of a cognitive load for the others to carefully read and understand, and the intending composer may not have their ideas respected. We have not yet designed a structured language for describing intent in subgoals, but we recommend composers to be concise. A structured language may help support automated agent collaborators in the future. This is discussed further in the Therapy use case (See §5.3).

Intention sharing may help distinguish between two kinds of collaboration issues. One scenario is where a composer agrees with another composer's intention about an edit, but disagrees with the implementation. This may lead to the second composer editing the first composer's material with care to preserve the original sound, or replacing it with something of similar effect. The other scenario is where one fundamentally disagrees with the artistic intention of another. An effective SharedPlan communication system may disambiguate these cases.

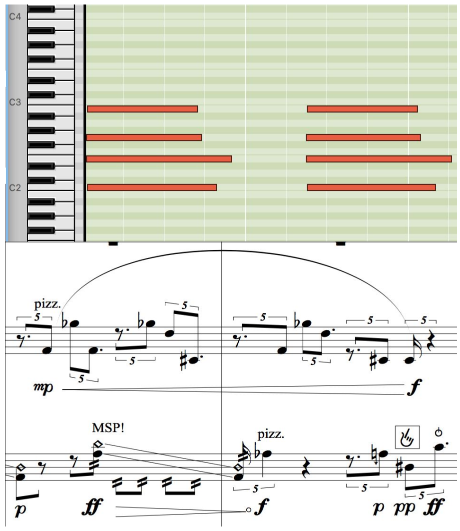

Figure 2: Music represented by MIDI (top) and MusicXML (bottom)

2.4. MIDI and Edit Actions

D: should connect to Collaborative Ideation work here. We don't mention it elsewhere other than Related Work Our system represents music in MIDI format. MIDI is a protocol for communicating discrete information about the pitch, duration, and dynamics of individual notes. A musical work is described by specifying the vertical arrangement of individual notes as chords and their horizontal arrangement over time. There is a General MIDI Sound bank that maps integers to specific electronic instrument sounds such as *grand piano* and *Ocarina*. Most familiar software for music notation and music production build user interfaces on top of basic MIDI editors. Composers are able to segment MIDI files into blocks corresponding to phrases, themes, or sections. Segmentation may represent intentions about musical form. For example, one may segment a composition into exposition and development section.

During an interaction with the system, a composer is able to change a composition by editing the MIDI in several low-level or high-level ways. Composers can add a new block, edit or remove an existing block, swap the order of two blocks, merge two blocks, or split a block into two. At each iteration, the

Action	Description
Insert,Delete,Replace	Add, remove or replace a block
Move	Rearrange order of a block
Split, Merge	Divide or combine a block

Table 1: Summary of Atomic Edit Actions on Score

system suggests an option that may bring the work-in-progress closer to what is specified by the SharedPlan (See §4.2).

2.5. Design Failure Modes

There are multiple potential failure circumstances of the proposed design:

Failure to compose No composer may elect to work on a given SharedPlan.

Failure to satisfy The creator of the original SharedPlan may never approve or reject the produced work, leaving it in an undefined state.

Subplan not released A subplan may be defined but no composer ever completes the subplan, leaving the work in an incomplete state.

Evaluation not converging The evaluation process may continue to offer suggestions that are rejected by the composer. Composers may abandon using evaluation.

Failure to evaluate there may not be sufficient training data for a given genre / mood combination so the evaluation may fail to provide sufficiently useful recommendations.

Failures result in a specific SharedPlan not advancing, but none of these failures affect other SharedPlans in the system. A SharedPlan requester may still accept a piece under some of these conditions.

3. User Interface

There are three aspects to the user interface: for SharedPlan requesters, for composers in workflow, and for composers actively editing the musical score.

3.1. SharedPlan Workflow User Interface

There are three possible commands in this context: **EditPlan** (including create/edit/delete), **Approve, Reject**. Since the user may not have any musical background, this user interface must be accessible to as large a population as possible. Ideally, this would be a simple looking, reactive web-based interface that would work equally well on mobile devices as well as desktops. Music apps such as Spotify may wish to integrate this functionality so a web-based API must also be accessible.

For an unsophisticated user, **Approve** and **Reject** may be considered implicit. If the user listens to the composition past a certain point, such as 80% of the total length of the work, it may be deemed implicitly approved. Similarly, if the user advances to another song within the first 20% of the work, the composition is implicitly rejected. Implicit rejection does not provide any causal reason for the rejection, so trial and error analogous to reinforcement learning must be used to discover the hidden reasons why a work was implicitly rejected. Other user actions (such as stopping all music in the middle of play) do not provide enough signal to determine implicit actions so the work's acceptance status remains unknown. Requiring explicit approval / rejection is unlikely to be used in practice by users with short attention spans. The system will learn user preferences more accurately by empirical observation.

3.2. Subplan Workflow User Interface

The market reality is the most modern composers use an existing Digital Audio Workstation (DAW) such as Ableton or LogicPro. Rather than replacing the composer's primary user interface, integration should be performed within the menu system of the most common DAWs. There are only two commands required in this context: **EditSubplan** (including create/edit/delete) and **Release**.

3.3. Composer Edit / Evaluation User Interface

This is the most complex and demanding user interface as it requires the highest degree of interactivity. Rather than reimplementing the multitude of music editing features of a mature DAW, integration to existing DAWs is again preferable over creating a *de novo* user interface. The most novel aspect would be the integration of the evaluation to the editing task. Ideally the evaluation result would decorate and annotate the score; enabling the composer to address results of the evaluation while remaining in a known editing context, analogous to seeing and resolving comments in Microsoft Word or a Google Doc.

4. Automated Analysis of Musical Structure

D: as mentioned in email, should be consistent about Evaluation vs. Analysis terminology

Automated analysis lets composers know how close their work-in-progress is to their goal. Additionally, it suggests structural edits that could further improve the piece. Crucial to our system, our computational approach models the structure of a piece of music in relation to the expected trajectory of surprise and redundancy that a listener experiences. Our work follows directly from The Information Dynamics Approach. (Abdallah et al., 2012) We first discuss the nature of the musical analysis used in our system, and then discuss how this method supports goal checking and edit suggestion.

4.1. Entropy of Musical Events and Divergence

Let X be a discrete random variable that takes on values from the set $\mathcal{X}$. For example, X may represent the next chord that a listener hears in a piece of music. The event $X = x$ indicates that the listener heard X take on a specific value x. Let $p_X(x) = p(x)$ denote the probability that X will take on value x, *before* the listener hears the event, as estimated by a distribution that the listener brings with them from prior musical experiences, as well as from what they have heard in the piece so far. $-\log\ p(x)$ then corresponds to the *surprise* of the event, because the more the listener expects the event, the lower the surprise, where the log is taken for convenience. Since X represents the event that the listener is about to hear, we can represent the expected surprise of X averaged over all possible values it may take on as:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

which corresponds to the Entropy of X, $H(X)$. Intuitively, this means that the listener does not know what the next event will be (e.g., which chord will be played next), but from context they expect a certain extent of surprise from the next event in general. The current state of listening is maintained as a distribution over future events.

Since we choose to represent musical structure in terms of the surprise dynamics of the listener, it is necessary to describe the way in which the listener's distribution over future events changes as they hear each new event. Upon hearing $X = A\flat^{\Delta 7}$, how does a listener's distribution over future events differ from how it was before they heard that chord? The Kullback-Leibler Divergence of one distribution from another captures this notion of distance between distributions. Avoiding subscripting X with timesteps, let X' be the revised distribution over the next event *after* hearing $X = x$ in the context of the existing distribution.

$$D_{KL}(X' \parallel X) = \sum_{x \in \mathcal{X}} p_{X'}(x) log \frac{p_{X'}(x)}{p_X(x)}$$

This is read as "the divergence from X' of X", and is the average over the ratio of point-wise log probabilities between the two distributions, weighted by $p_{X'}(x)$. For an accessible yet informative discussion of the significance of entropy as a measure of information and KL Divergence [10], see Christopher Olah's post on Visual Information Theory [11].

This divergence describes the amount of revision to a listener's distribution over the future that happens as they hear each event. Let this be called the instantaneous *predictive information* of the event $X = x$ as the listener hears it. When a surprising event occurs and causes the listener to drastically revise their distribution (i.e., this same event will be less surprising in the future), this corresponds to high predicative information. On the other hand, if thirty strikes of the same chord have just happened, hearing a thirty-first articulation does not communicate much predictive information. Our system measures the predicative information *rate* (PIR) over the duration of the piece (or work-in-progress), and uses the trajectory of this rate to summarize the structure of a piece of music as it is expected to be perceived by the listener. In the running example, entropy and divergence are discussed with respect to chord sequences heard by the listener, and their expectation over next chords in context. In even a simple piece of music, the listener tracks multiple interacting parameters: evolving harmony, rhythm, timbre, and more (See §7 Future Work). Our system calculates predictive information by averaging the quantity across trigram harmony and rhythm at each time step.

4.2. Current Design: Analyze, Suggest, and Edit

Harmonia uses PIR to calculate the proximity of a work-in-progress to the SharedPlan goal. Using PIR, the system gives feedback to a composer with respect to the composer's editing decisions, and provides suggestions that may bring a piece closer to the goal. We calculate the PIR for the most popular β% of pieces matching SharedPlan keywords (where β is a parameter to be specified), and average the PIR curves to create a "characteristic curve" that represents the typical structure of a piece of music fitting the criteria.

[10] Another interpretation of entropy is the average number of bits required to send a message from a distribution p under an optimal variable-length coding scheme. The KL Divergence of q from p is the increase in the average bits per message when one communicates items from p using a code optimized for q. This is the difference between the cross entropy $H(p, q)$ and entropy $H(p)$.

[11] `http://colah.github.io/posts/2015-09-Visual-Information`

Comparing the characteristic curve with the PIR curve of the work-in-progress, our system can estimate a notion of distance from the musical goal specified in the SharedPlan. Let the difference be denoted as Δ, where small Δ indicates that two pieces of music are similar in structure with respect to surprise and redundancy over time. Even without edit suggestions made by the system, a composer may simply see whether their latest edit brings the piece of music closer to (lower Δ) or further from (higher Δ) the SharedPlan. Composers may prefer to go with edits that decrease Δ, or may choose to stick with their edit even if Δ increases. Reasons for going with a "worsening" action include laying down material that further edits will re-contextualize, whether by the same composer or by others. In this case, it is important to issue a new subplan.

As specified in §2.4 , composers may edit a composition in several defined ways. Excluding options that require advanced machine composition, the system may give actionable edit suggestions according based on PIR scores. It may recommend repeating or deleting any existing block, or swapping any pair of existing blocks. For any work-in-progress of reasonable length, there is a tractably enumerable set of such choices. The system can just try each choice of deleting, repeating, and swapping, and suggest to the user the choice that minimizes Δ.

5. Use Cases

5.1. Use Case 1: Individual User, Individual Composer

A listener, who may be a non-musician, would like a new piece of music for a very specific function such as study or exercise music (**?**). We consider the case that the listener specifies a new project defined by a mood and genre. In this simple case, we consider a single composer who iterates over the piece with assistance from our system until the requester is satisfied.

5.2. Use Case 2: Multiple Composers

Consider multiple composers who create a SharedPlan together, and then collaboratively create the specified piece. During the process, a composer may wish to take the piece into a new direction that doesn't correspond to the SharedPlan. They are able to create a new branch of revision history to experiment on before attempting to re-integrate material with the original group or creating a new SharedPlan with the material. On the main branch, composers issue subplans to give someone else an opportunity to complete their idea when they are stuck. Even an insufficient attempt by another composer to release the subplan

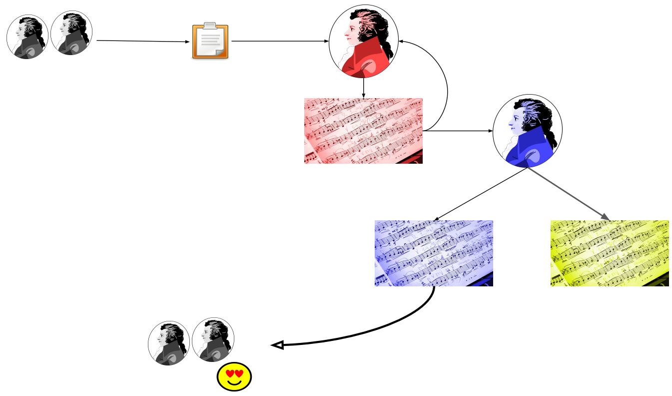

Figure 3: Use Case 2: Multiple composers define a SharedPlan. Composer 1 (red) starts iterating, commits a work-in-progress and defines a subplan. Composer 2 (blue) retrieves the subplan and continues iterating until the original composers are satisfied. Composer 2 maintains a separate branch for personal investigation.

may inspire the issuer. See §6.1 for a discussion of composer-system interaction success criteria.

5.3. Use Case 3: Therapist with Agent & Human Composers

Consider a music therapist who treats their patients using newly composed music, specific to a given patient's needs. Music therapy is used in many contexts including children and adults with mood disorders (e.g. depression and PTSD), developmental disorders (e.g. autism and ADHD), and neurological conditions (e.g. dementia) (Hole et al., 2015). This SharedPlan may have a more highly refined specification than music for casual listening, such as a specific tempo or special therapeutic timbres (sound qualities).

Due to high volume of personal treatment plans, the initial SharedPlan worked on by an agent, which performs the bulk of the composition. Consider an agent in the reinforcement learning setting, exploring the space of musical edits while in a feedback loop with the automated evaluation system. The agent's composition is then reviewed by a human who may make only small modifications to make the music more warm or less mechanical. The clinician makes the final approval and then provides the music to the patient for treatment.

5.4. Use Case 4: Pseudo Music Therapist

For those who do not have access to a music therapist, Harmonia can be used as a platform to assist patients in achieving certain goals and outcomes, such as creating a song that will express sadness or grief, or to express acceptance and hope for the future (Dalton and Krout, 2006). The recommendations generated by the evaluation process can assist a non-musically

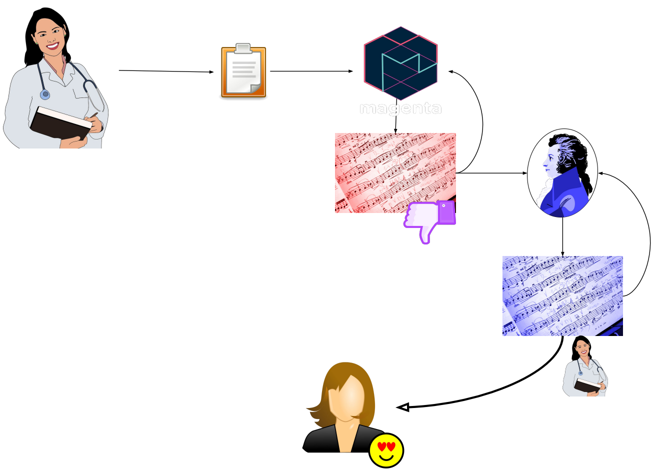

Figure 4: Use Case 3: A clinician defines a SharedPlan with more complex metadata for therapeutic use. An agent performs the initial composition which consists of the bulk of the work. A human composer reviews the work and makes minor adjustments. The clinician approves the work and provides the result to the patient for treatment.

trained composer to create reasonably sounding music for emotional expression. The act of creative composition is itself therapeutic employing various regions in the brain, including the region responsible for autobiographical memory in the case of improvisation. (Hilliard, 2001; Bensimon et al., 2008; Limb and Braun, 2008; Carr et al., 2012)

6. Discussion

6.1. Validation Methodology

We intend to validate our system with several studies on small groups of composers that specify a musical goal and collaborate on a piece of music facilitated by our system. We also intend to validate our system for non-musician requesters seeks for their musical desires to be addressed. The Therapy use cases should be included in studies once these two less-complex cases are assessed. Testing will include the validation of several system aspects:

1. Do composers understand and benefit from system feedback? If the composer proposes edits A and B, and the system scores B higher than A, which does the composer pick? What percentage of preferred edits do composers stick with? For system-suggested edits, what percentage of suggestions do composers accept?
2. Do composers feel that communication through subplans is efficient for collaborating on ideas requiring the work of several composers?
3. Do composers and requesters feel as if finished pieces respect the SharedPlan?

To address point 1, we can compare composer experience using a) our revision and metadata system without automated analysis, b) system with score feedback for composer-specified edits, but no system-suggested edits, c) system with score feedback to composer-specified edits and system suggestions. To address point 2, we should explore other methods of inter-composer communication in place of the subplan issue/release procedure. Two approaches are a) traditional email to complement version-controlled music with no subplan options b) additional text documents to describe the equivalent of subplans but external to the SharedPlan and in longer form, committed along with the SharedPlan and musical score. Point 3 will come from discussion with the requesters and composers.

The percentage of edit suggestions that each composer accepts does not only describe the success of the system's suggestions, but also the personality of each composer: are they generally open to suggestions or do they tend to stick with their own ideas? This will help to not blame the system for poor performance when dealing with a composer that tends not to take suggestions in general. Aspects of a composer's collaboration profile may also be collected orthogonally to system usage through simple collaboration games (to be designed).

Several obstacles remain to setup the studies. This involves implementation of the system, including information retrieval aspects. Compositions may take some time to work on, so these studies may not be very quick. We may design smaller-scale composition games for these purposes.

6.2. Enhancing or Stifling Creativity

The question arises whether algorithmic evaluation of a composition is a potential means of stifling music creativity. Is it possible that a corpus based evaluation will nudge composers into making music that resembles existing work rather free creative expression? Firstly, the evaluations and suggestions are purely advisory and can be ignored by composer. Secondly, the entropy-based evaluation does not comment on material as much as organization of material. The evaluation could be further enhanced with semantic understanding of musical material to show the composer that the work-in-progress resembles certain existing works too closely, avoiding potential future intellectual property infringement. No composer can possibly know all existing work in the corpus. A composer can now have an independent measure of the originality of the new work, which will encourage greater creativity.

6.3. Limitations

There are several limitations in this proposal that are inherent in the design. This is purposeful, as addressing these limitations may result in undo expansion of scope or problem tractability. Nevertheless, it is important to be explicit about known limitations:

No explicit support for improvisation. The collaborative design is inspired from software engineering teamwork, which is a solitary activity. Unlike software, in music very fruitful results may occur in real-time improvisation by several musicians. The design does not preclude improvisation but does not account for it either. There may be interesting paths to explore, discovered during improvisation that would not be captured by this design. A composer would have to create separate branches and record improvisation sections in each branch. The design does not account for partial interchange between branches.

Information Theory. We have presented the way in which PIR may be calculated over single musical parameters such as chord progressions or rhythms, but composers create many complex relationships across parameters. Cohen (1962) demanded 1) a theory of interactions for multiple streams of musical information, for example, to explain how rhythmic information may make harmonic events more or less certain 2) a theory for multiple levels of hierarchy coming at once from a single stream of musical information. Rhythms constitute a local time feel but also accelerate a piece toward new sections. These are still complicated aspects of information today. The first point requires a generalization of mutual information to multiple random variables, which has been met with confusion through several coexistent approaches (Van de Cruys, 2011). The second point has not yet been successfully modeled (Widmer, 2016).

Model may not generalize. While we are confident that SharedPlans, revision control, and algorithmic evaluation are applicable to music composition, it is uncertain that the overall framework can be generalized to other domains. Version control for visual artwork seems promising, but simple text-based revision may not support editing capabilities for sufficently rich representations of visual art in an intuitive or useful way. Algorithmic evaluation of intent may be more difficult to design for a painting. Musical form, as treated by our system, is modeled at an information-general level also applicable to a story, movie, or play, but this model is limited to sequential forms. The static experience of a painting is less related and may require a different set of domain-specific criteria for capturing intent. To this end, generalizable techniques for automated analysis that bypass the necessity to encode domain-specific information are desirable, and may be approached with today's deep learning techniques (See §7 Future Work).

Representation is MIDI, not recorded acoustic signal. Vocals are not supported. MIDI is far easier to generate and maintain under version control, but has two limitations. The first is that many interesting sounds such as vocals and other samples of recorded sound cannot be expressed in the current design. MIDI placeholders may be made for these sounds, such that someone substitutes them into the piece post-collaboration, but collaboration over these sounds themselves is not supported. Given this setting of *written*, rather than *performed* music, the second limitation is that MIDI encodes only a subset of written musical expression. MusicXML is another text-based format that is more flexible than MIDI, while still being amenable to revision control (See Figure 2). It allows musicians to visually communicate performance techniques (e.g. pizzicato, slur, sul ponticello) but would complicate SharedPlan communication protocol and automated analysis.

Corpus based shared beliefs may not be robust. One of the key assumptions is that genre / mood permutations generate a reasonable representation of the intent of the music. This representation is corpus-based. This assumption presumes that genre and mood classification solutions exist and that the representation is robust enough for the evaluation function to provide useful guidance. Since the system has not yet been built, certainty of this assumption remains a source of further investigation.

7. Future Work

Harmonia is intended to serve as a base platform for multiple aspects of collaborative composition, and is modular enough such that individual functionality within the framework may be replaced with newer techniques. The area of creative agents is undergoing rapid evolution. Providing a framework to ensure shared intentionality, artistic consistency, and collaboration with humans should provide further acceleration. Multi-agent online improvisation such as AI Duet is another ripe area for exploration both for artistic purposes as well as for 'call and response' treatment of autism and other disorders.

The current automated analysis system is based on handcrafted entropic measures specific to music information. Given the rapid advance of Recurrent Neural Nets in natural language understanding and other sequential data streams, further investigation is warranted for applying deep learning to musical evaluation. In general, a deep learning approach to *analysis* of music (Jeong and Lee, 2016) is a huge (and exciting) obstacle in the training of automated, musically interactive agents, which could have roles in our system. If Harmonia were used by a large set of users, the resulting listening dataset would provide important training data for applying Reinforcement Learning and other methods to further enhance agent composition. A larger dataset would also be of significant benefit to music therapists who could share empirical data on effective treatments tailored to specific demographic and psychographic profiles.

There are significant limits when using MIDI for musical expression. Incorporating more complex musical encoding formats such as recorded audio, or at least MusicXML, would allow for greater range of expression but would complicate analysis. Adding vocals is an interesting challenge. Given recent developments in speech generation, is it possible to extend the idea to vocals?[12] Can a neural net be programmed to 'sing' like Alicia Keys? Commercial applications of Harmonia include the aforementioned intellectual property similarity evaluation. The analysis module may also incorporate commercial goals such as potential popularity (Pham et al., 2015).

Conclusion

Our proposal contributes two novel areas of collaborative music composition. First, we present a workflow incorporating SharedPlans with explicit intentionality, subplans, Collaborative Ideation, revision control and automated agents. Second, we present algorithmic evaluation of a composition against intention, to ensure that human and agent composers cooperate in reaching the shared objective. These contributions extend the state of the art in collaborative composition frameworks, beyond the ideation of SoundCloud and the revision control without intentionality of Blend. We see this initial proposal as a start of several interesting areas to explore leading to improved creativity, diversity and applicability of music composition as a tool for composers, music therapists and listener pleasure.

Acknowledgements

We are honored to be part of Professor Grosz's final class. We deeply appreciate the guidance that she and Peter Krafft have provided all semester. We wish them all the best in their future endeavors and SharedPlans.

Special thanks to Dr. David Greenberg, who provided important insights and suggestions regarding the application of Harmonia for music therapy, to composers Daniel Pencer and Ariel Herbert-Voss, whose discussions about the proposed user interface of our system were highly valuable, and to Pao Siangliulue, for the discussion about composer communication.

[12] `https://lyrebird.ai/`

References

S. A. Abdallah, H. Ekeus, P. Foster, A. Robertson, M. D. Plumbley, Cognitive music modelling: An information dynamics approach, in: Cognitive Information Processing (CIP), 2012 3rd International Workshop on, IEEE, 1–8, 2012.

M. Bensimon, D. Amir, Y. Wolf, Drumming through trauma: Music therapy with post-traumatic soldiers, The Arts in Psychotherapy 35 (1) (2008) 34–48.

C. Carr, P. d'Ardenne, A. Sloboda, C. Scott, D. Wang, S. Priebe, Group music therapy for patients with persistent post-traumatic stress disorder–an exploratory randomized controlled trial with mixed methods evaluation, Psychology and Psychotherapy: Theory, Research and Practice 85 (2) (2012) 179–202.

J. E. Cohen, Information theory and music, Behavioral Science 7 (2) (1962) 137–163.

T. A. Dalton, R. E. Krout, The grief song-writing process with bereaved adolescents: An integrated grief model and music therapy protocol, Music Therapy Perspectives 24 (2) (2006) 94–107.

B. J. Grosz, S. Kraus, Collaborative plans for complex group action, Artificial Intelligence 86 (2) (1996) 269–357.

D. Harhoff, K. R. Lakhani, Revolutionizing innovation: users, communities, and open innovation, The MIT Press, 2016.

R. E. Hilliard, The effects of music therapy-based bereavement groups on mood and behavior of grieving children: A pilot study, Journal of Music Therapy 38 (4) (2001) 291–306.

J. Hole, M. Hirsch, E. Ball, C. Meads, Music as an aid for postoperative recovery in adults: a systematic review and meta-analysis, The Lancet 386 (10004) (2015) 1659–1671.

C.-Z. A. Huang, D. Duvenaud, K. Z. Gajos, Chordripple: Recommending chords to help novice composers go beyond the ordinary, in: Proceedings of the 21st International Conference on Intelligent User Interfaces, ACM, 241–250, 2016.

L. Hunsberger, Making SharedPlans more concise and easier to reason about, in: Multi Agent Systems, 1998. Proceedings. International Conference on, IEEE, 433–434, 1998.

I.-Y. Jeong, K. Lee, Learning Temporal Features Using a Deep Neural Network and its Application to Music Genre Classification, in: Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR), 434–440, 2016.

C. J. Limb, A. R. Braun, Neural substrates of spontaneous musical performance: an FMRI study of jazz improvisation, PLoS one 3 (2) (2008) e1679.

J. W. Oberholtzer, A Computational Model of Music Composition, Ph.D. thesis, Harvard University, 2015.

J. Pham, E. Kyauk, E. Park, Predicting Song Popularity, URL http://cs229.stanford.edu/proj2015/140_report.pdf .

P. Siangliulue, K. C. Arnold, K. Z. Gajos, S. P. Dow, Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, ACM, 937–945, 2015.

P. Siangliulue, J. Chan, S. P. Dow, K. Z. Gajos, IdeaHound: Improving Large-scale Collaborative Ideation with Crowd-powered Real-time Semantic Modeling, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, ACM, 609–624, 2016.

L. Torvalds, J. Hamano, Git: Fast version control system, URL http://git-scm.com .

T. Van de Cruys, Two multivariate generalizations of pointwise mutual information, in: Proceedings of the Workshop on Distributional Semantics and Compositionality, Association for Computational Linguistics, 16–20, 2011.

G. Widmer, Getting closer to the essence of music: The con espressione manifesto, ACM Transactions on Intelligent Systems and Technology (TIST) 8 (2) (2016) 19.