Market-based Control in Interactive Music Environments

Arjun Chandra[1], Kristian Nymoen[1], Arve Voldsund[1,2], Alexander Refsum Jensenius[2], Kyrre Glette[1], and Jim Torresen[1]

[1] fourMs, Department of Informatics, University of Oslo, Norway
[2] fourMs, Department of Musicology, University of Oslo, Norway
chandra|krisny|kyrrehg|jimtoer@ifi.uio.no
arve.voldsund|a.r.jensenius@imv.uio.no

Abstract. The paper presents the interactive music system SoloJam, which allows a group of participants with little or no musical training to effectively play together in a "band-like" setting. It allows the participants to take turns playing solos made up of rhythmic pattern sequences. We specify the issue at hand for enabling such participation as being the requirement of *decentralised coherent circulation* of playing solos. Satisfying this requirement necessitates some form of intelligence within the devices used for participation, with each participant being associated with their respective enabling device. Markets consist of *buyers* and *sellers*, which interact with each other in order to trade commodities. Based on this idea, we let devices enable buying and selling, more precisely *bidding* and *auctioneering*, and assist participants trade in musical terms. Consequentially, the intelligence in the devices is modelled as their ability to help participants trade solo playing responsibilities with each other. This requires them to possess the capability of assessing the *utility* of the associated participant's deservedness of being the soloist, the capability of holding *auctions* on behalf of the participant, and of enabling the participant *bid* within these auctions. We show that holding auctions and helping bid within them enables decentralisation of co-ordinating solo circulation, and a properly designed utility function enables coherence in the musical output. The market-based approach helps achieve decentralised coherent circulation with artificial agents simulating human participants. The effectiveness of the approach is further supported when human users participate. As a result, the approach is shown to be effective at enabling participants with little or no musical training to play together in SoloJam.

Keywords: active music, collaborative performance, conflict resolution, market-based control, decentralised control, algorithmic auctions

1 Introduction

In many musical cultures and genres there is often a large gap between those who *perform* and those who *perceive* music. In such ecosystems, the performers (musicians) *create* the music, while the perceivers (audience) *receive* the music [14]. Even though perceivers may have some control of the music creation in a concert situation, by means of cheering, shouting, etc., this only indirectly changes the musical output. The divide between performer and perceiver is even larger in the context of recorded music, which is typically mediated through some kind of playback device (CD, MP3 file, etc.). Here the perceiver has very limited possibilities in controlling the musical content besides starting/stopping the playback and adjusting the volume of the musical sound.

The last decades have seen a growing interest in trying to bridge the gap between the performance and the perception of music [8]. Examples of this can be seen as interactive art/museum installations, music games (e.g. Guitar Hero) [9], keyboards with built-in accompaniment functionality [2], "band-in-a-box" types of software, mash-up initiatives of popular artists [13], sonic interaction designs in everyday devices [11], mobile music instruments [4], active listening devices [6, 10], etc. An aim of all such *active music* systems is to give the end user control of the sonic/musical output to a greater or lesser extent, and to allow people with little or no training in traditional musicianship or composition to experience the sensation of "playing" music themselves [7].

There are numerous challenges involved in creating such active music experiences: everything from low-level microsonic control (timbre, texture), mid-level organisation (tones, phrases, melodies) to large-scale compositional strategies (form). In addition comes all the challenges related to how one or more participants can control all of these sonic/musical possibilities through mappings from various types of human input devices. In this paper we will mainly focus on creating a system that is flexible enough for the participants' interaction, yet bound by an underlying compositional idea.

Our approach in SoloJam is to allow for a group of participants with little or no musical training to come together and behave as a "band" of musicians, wherein, they play their respective solos in turn. Thus, the responsibility of playing solos circulates around the band and continues to do so until an indefinite period. To solve the problem of co-ordinating the circulation of responsibility of playing these solos autonomously and effectively, we propose an approach inspired by the economic sciences, in particular markets. Specifically, we borrow the concepts of *auctions* and *utility* to address the problem. Our investigation shows that trading the responsibility of being the soloist via auctions does indeed help decentralised, thus autonomous, circulation of solos within the group. In addition, a careful consideration of the utility function helps participants produce coherent musical output.

We start by introducing the interactive musical scenario that we refer to as SoloJam in Section 2, specifying the issue with enabling participation within it. We then describe our proposed market-based approach to tackling the issue, and the implementation details for the same, in Section 3. Section 4 then looks at the application of the approach within SoloJam, investigating the approach for its effectiveness in enabling participation by artificial agents (who simulate participants with little or no musical training) and human users. This section also discusses typical modes of communication in bands in relation to SoloJam. This is followed by a discussion on the flexibility offered by SoloJam. We conclude in Section 5.

2 The Musical Scenario

In our current context we are interested in creating a system that allows for a group of participants with little or no musical training to get the feeling of being involved with creating music, yet defined in such a way that a certain level of musicality is ensured in the final sounding result. The participants are to play music using a device that assists them for the same. Such a device, together with the participant using it, is what we call a *node* in this paper. The participant may either be a human user using the device, or an artificial agent behaving in a specified manner simulating a user, and as such associated with the device. Multiple participants would thus aim at controlling various musical features within the composition. As such, we will need the devices to help co-ordinate the participants' intentions. Situations might arise where multiple participants intend on controlling the same musical feature, giving rise to a conflict with regards to who might eventually control. This would be more prevalent when participants have little or no musical training, as they are not likely to be conversant with the typical modes of communication that trained musicians use when facing this problem. Thus, the devices will have to resolve these conflicts. For such interactive compositions, conflict resolution should be a necessary constituent part of the system, but indeed, not necessarily the only thing.

In this paper, we focus our attention on this conflict resolution aspect of compositions. As such, we imagine a band of musicians who want to play their respective solos pertaining to the same musical feature. Only one musician ever plays their respective solo at a time. We call this musician the *soloist*. However, over time, the playing of solos circulates across the band, as and when other musicians become soloists. The control of circulation of solos happens in a decentralised manner.

The musical space within the system considered in this paper is made up of rhythmic patterns. A sequence of rhythmic patterns when played by one node, is viewed as a *solo* in the context of this paper, until another node commences playing rhythmic patterns. Each rhythmic pattern has a specified number of beats, which we consider as one bar. Thus the musical output is supposed to be a series of rhythmic patterns, one in each bar. Each bar in a sequence can either be a repetition of the rhythmic pattern in the previous bar or not, specifically when played by one node as a solo. And, the next solo, which would be played by another node, should start with a rhythmic pattern that is not exactly the same as, and ideally only slightly different to, the one played by the soloist in the previous bar. The composition is specified by the aforementioned elements, which also describe the boundaries or constraints to which the musical output should adhere to.

As such, SoloJam can be seen as a compositional idea, or *musical scenario*, where a group of nodes acting in a decentralised fashion come together and take turns in playing a piece based on rhythmic solos. Though nodes act in a decentralised fashion, they must also be able to produce a coherent musical result.

2.1 The Issue With Enabling Participation

Given the scenario mentioned above, if a group of participants are to play music, the devices that they use for this participation cannot be traditional instruments. Instead, the devices need to possess some form of artificial intelligence which might allow the group to produce a coherent musical output, and help the participants do so via decentralised interactions with other participants, i.e. without requiring an expert to direct their interactions. Devices helping with coherence are required due to the assumption that the participants do not possess sufficient musical knowledge to produce a satisfactory result on their own. As such, what gets played should be influenced by the devices to some extent, whilst making sure that the participants are still able to explore the musical space themselves. Devices helping with decentralised interaction are required in order to adhere to the vision of a "band" where members organise themselves into taking turns playing solos, without a central authority directing them. Moreover, participants not possessing sufficient musical knowledge also renders them to be unfamiliar with the modes of communication used by trained musicians to a large extent. Thus, deciding who plays the solo next should be dealt with by the devices interacting intelligently with each other on behalf of the participants. Such intelligence in the devices forms the crux of the issue with making participants play together effectively within our musical scenario.

We define *decentralised coherent circulation* as giving us a yardstick against which to evaluate the effectiveness of the solution to the issue of allowing participants to effectively play together in SoloJam. *Decentralisation* means that there is no central control over the circulation of playing of solos by participants. *Coherence* in our case means for nodes to be playing slight variations of each others rhythmic patterns over time as and when they become soloists, such that the next soloist plays a slight variation of the rhythmic pattern played by the current soloist. Thus, our goal is to design an intelligent system that allows for both decentralised control and coherent musical output. It should enable participants who do not possess much musical knowledge to play together without requiring an expert to direct their interaction with other participants.

3 Market-based Approach For Enabling Participation

In recent years, there has been a surge of ideas being borrowed from the economic sciences for designing systems with interacting autonomous components. In the context of our work, a node can be seen as an autonomous component interacting with other nodes. A family of such ideas, known as market-based control, is aimed at applying economic principles to tackling resource allocation problems in distributed computing systems [3]. One of the key characteristics that markets possess, which market-based systems benefit from, is that of rendering the interacting components as being part of a decentralised system.

A typical market-based system consists of software agents representing components of the system, where each component has its own task to perform. These agents act and make decisions autonomously on behalf of the components, yet interact with each other based on a defined market mechanism, e.g. auctions or bargaining. Taking on roles of buyers and sellers of resources as and when needed, or indeed bidders for and auctioneers of such resources, they are able to engage in trades. Such resources may be needed by them in order for the components they represent be able to perform their respective tasks. Typically, virtual money is introduced in the system to facilitate exchanges as the agents interact in accordance with the market mechanism. As they interact with each other to trade, they attempt at maximising their utility function, which is usually derived from task requirements, the higher the utility, the better they enable performing individual tasks. With buyers paying more for resources they value more, and sellers charging what they are able to get away with, resources tend to go to components that value them most, all in a decentralised way.

A wide range of application domains have applied this concept since its inception [3]. We envisage software agents, as described above, as modelling the intelligence in the devices which form part of a node in our context. Thus, we take this concept into the domain of music in order to tackle the issue described in Section 2.1. A detailed specification of the market-based approach now follows.

3.1 Specification of the Approach

One can see the problem of decentralised control of circulation of solos as a resource allocation problem, where the resource can be viewed as a metaphor for *having the responsibility of playing a solo*. This responsibility is what needs to be continuously allocated to the node who may be *most deserving* of being the soloist within SoloJam at any point in time.

The concept of auctions has a long standing history in human society, where the idea is to have a mechanism in place that allows for the allocation of resources/goods/services via the exchange of these resources/goods/services with other resources/goods/services, or indeed some currency. Anything that may be exchanged has some value for the parties between which the exchange happens. This is where the concept of utility comes in. Utility [5, 15], as a concept, has a long history in the economic sciences as being an idea that allows for expressing the value of a choice or decision that one needs to make. For example, how much may one be willing to spend on buying a type of guitar amongst other choices, is the value of the guitar for the individual. This value can, with certain assumptions about the preferences of the individual with respect to making choices, be quantified in the form of a mathematical function. Such numerical expression of value makes exchanging resources/goods/services practical.

Assuming that it may be possible to compute the deservedness of being the soloist, at every time step, whilst the soloist is playing its solo, we make it also hold (broadcast) an *auction*, in which all other nodes can *bid* in order to become the next soloist. We thus design the node such that every node can evaluate the deservedness of itself being the soloist. This is computed as the viability, or in economic terms, *utility* of its current rhythmic pattern being played in the next bar. The utility values derived from their respective rhythmic patterns are what the nodes use as their respective bids. As such, at any given time, the node with the highest utility must be the soloist, provided this value is computed truthfully (or honestly). At every time step, a bidder node can also change its respective rhythmic pattern, in order to come up with a new rhythmic pattern from which a higher utility may be derived, as compared to the utility derived from its current rhythmic pattern. The transfer of responsibility happens when a bidder node wins the auction held by the soloist. This necessitates a gain for the soloist, i.e. the auction can only be won if the soloist gains from handing over the responsibility to the highest bidder. This implies that the utility derived from the rhythmic pattern that the soloist is currently playing, must, at the time of the transfer, be lower than the highest bid it receives. We now detail the auction mechanism used for node interaction, helping achieve decentralised circulation of responsibility, and elaborate on the computation of utility which quantifies deservedness and helps garner coherence in the musical output.

Auction Mechanism. The soloist takes the role of an auctioneer and holds a second-price sealed-bid auction, in particular, the *Vickrey auction* [16] in every bar. This is done in order that the soloist receive bids from the bidders, which then are used to decide whether or not there is a winner to whom the responsibility of playing the solo would pass in the next bar. The reason for this design choice is that Vickrey auctions deem truthful bidding to be the dominant bidding strategy. In our case, this means that a bidder can do no better than bidding with the true utility value derived from its rhythmic pattern. The second-price nature of the auction suggests for the winner of the auction to make a payment equal to the value of the second highest bid to the soloist. The second price aspect of this auction mechanism makes truthful bidding a dominant bidding strategy. However, in the current setup we do not exchange money[3] (in the form of such payments by bidders to the soloist). This means that, although the transfer of responsibility necessitates a gain for the soloist, as mentioned above, the soloist only ever compares the received bids and the current utility derived from its own rhythmic pattern, in order to ascertain whether or not it should hand over the responsibility to the highest bidder. Ties in bids, when the bids are higher than the soloist's rhythmic pattern utility, are broken randomly. The sealed-bid nature of the auction requires that the bids are not public and only known to the bidder and soloist. We leave the consideration of exchange of

[3] The auction and bidding setup in SoloJam allow for money (or virtual money), in the form of bid values to be exchanged. But, we only consider monitoring the utilities for now.

money and other possibilities offered by this auction mechanism to the future, when dealing with more complex variants of SoloJam.

Utility. To participate in the auction effectively, each node must have a way of evaluating and communicating a value that it considers playing its current rhythmic pattern in the next bar to be worth. A rhythmic pattern in SoloJam is represented as a bit string parsed from left to right, whereby, a 1 indicates 'triggering a beat' and a 0 represents 'not triggering a beat'. For each node, we define a utility function which the node uses to evaluate the value its current rhythmic pattern can yield, both in relation to itself and to the soloist, knowing its role as either a bidder or the auctioneering soloist. The following equation specifies part of this utility function:

$$u_i = \frac{c}{(1 + aD_l)(1 + bT_l)} \qquad (1)$$

Here, D_l is the hamming distance of a node's current rhythmic pattern with respect to the soloist's current rhythmic pattern, T_l is the length of time a node has been playing the solo, i.e. the number of bars a node has played rhythmic patterns as a soloist, the coefficient a is the importance (in terms of a weighting) given to D_l, the coefficient b is the importance (in terms of a weighting) given to T_l, and c is a normalisation constant. In addition to this, two more conditions completely specify the utility function. These clauses being:

1. The utility is *zero* for a bidder node if D_l goes below $\epsilon\lambda$, where ϵ is a small percentage of the length of the rhythmic pattern (λ).
2. The utility is *zero* for a bidder node if the node has handed over control to a new soloist node in the previous time step.

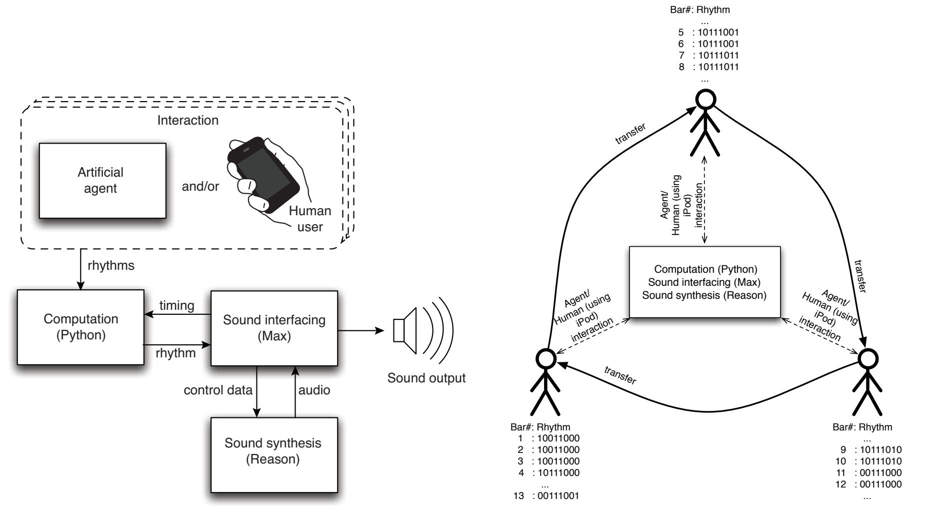
According to the utility function above, the longer (in terms of bars) a node is the soloist, the lesser it values its current rhythmic pattern, indicating boredom or fatigue, of which the node is made aware via the utility function. The node also possesses knowledge about the hamming distance between its own and the soloist's respective rhythmic patterns. This knowledge can be used by the node to come up with rhythmic patterns that yield higher value, given the soloist's rhythmic pattern. The closer a node can match its rhythmic pattern against the soloist's pattern, the more is the value it can derive from its pattern. This remains true as long as the match does not get closer than or equal to $\epsilon\lambda$, allowing for the node to stir clear of intending to play a rhythmic pattern that may be very similar to or exactly the same as that of the soloist (as per the first clause above). Additionally, we can see that this specification of utility, taking the soloist's rhythmic pattern into consideration, also provides the node with a gradient (i.e. the closer the rhythmic pattern to that of the soloist, the higher the value it yields), which it may make available to the participant in order for them to come up with rhythmic patterns which are slight variations (at least $\epsilon\lambda$ different) of the soloist's rhythmic pattern. As such, in addition to computing deservedness, we see the utility function as a means of instilling coherence in the musical output from SoloJam. Note that D_l forms the main link between nodes (the node in question and the current soloist node), and the coefficient a associated with D_l emphasises or otherwise, the strength of this link. We will put this coefficient to use for the investigation carried out in this paper in Section 4. The clauses above further indicate a way of carefully considering designing the utility function in order for a globally coherent piece of music to result from decentralised interactions within SoloJam. The first clause suggests for there not to be a perpetual repetition of the same rhythmic pattern by all the nodes of SoloJam, which would be monotonous. The second clause allows for a node to not take over the responsibility soon after it released it, which may happen otherwise, since the node's rhythmic pattern would already be a slight variation of the new soloist that took over the responsibility from this node. Not considering this clause may thus reduce the variations that may occur in the music performance in the global sense.

3.2 Implementation

Fig. 1 shows the building blocks of the implementation of SoloJam. Fig. 1(a) outlines the schematic of the implementation of SoloJam. The current SoloJam scenario has been implemented on a Macintosh computer, in conjunction with iOS devices for human interaction within the scenario. The setup can be broken down into 4 modules: the Computation module, the Interaction module, the Sound interfacing module, and the Sound synthesis module.

The Computation module is implemented in Python and simulates our market-based approach for effective participation described in Section 3.1, with a thread representing each node. These threads interface with the Interaction module as well as the Sound interfacing module. The Interaction module can function in two ways. If an artificial agent is to be part of the node, the thread in the Computation module representing this node is made to implement the functionality of the agent in terms of the manner in which this agent comes up with rhythmic patterns. If a human user is to be part of the nodes, iOS devices (specifically iPod Touch) are used for sensing the shaking of the device (using the built-in inertial sensors). The signals from shaking are sent as Open Sound Control (OSC) [17] messages to a thread in the Computation module associated with the device, which are then converted into rhythmic patterns within this thread. The bit strings representing rhythmic patterns are further sent as OSC messages to the Sound interfacing module, together with the utilities/bids (computed within the Computation module) that the soloist/bidder nodes derive from their respective rhythmic patterns in every bar.

The Sound interfacing module is implemented as a Max/MSP patch. It serves as a control module for the SoloJam scenario, accepting strings of rhythmic patterns, synchronising and converting them to control signals for the Sound synthesis module. The audio streams from the Sound synthesis module are channeled back to the Sound interfacing module for mixing and effects processing. The Sound interfacing module also performs a visualisation of various aspects of the



(a) Implementation. (b) Illustration.

Fig. 1. Building blocks of the implementation of SoloJam showing (a) a schematic of the implementation of SoloJam, and (b) an illustration of the SoloJam scenario within the context of this implementation.

system, such as node utilities. The Sound synthesis module is currently instantiated as a virtual sound module rack in Reason. A drum kit synthesiser module is used for each node. Reason is controlled by the Sound interfacing module through ReWire. MIDI signals are sent to the synthesisers, and the audio streams are sent back to the Sound interfacing module.

Fig. 1(b) illustrates the SoloJam scenario within the context of the aforementioned implementation. It shows 3 agents or human users participating in the scenario. The rhythmic patterns associated with each participant at various bars are shown. These rhythmic patterns are fed in to our market-based approach for effective participation simulated by the Computation module. As per the rhythmic patterns shown, one possibility for the transfers of responsibility of playing solos is indicated in the figure.

4 SoloJam with Participants

We now look at how the market-based approach proposed in this paper, consisting of auctions and a properly designed utility function, enables effective participation within the composition. We primarily look at the case where artificial agents are considered as simulating the behaviour of participants with little or no musical training, and act within SoloJam as participants. The case where SoloJam involves human participants is also discussed.

4.1 SoloJam with Artificial Agents: Enabling Participation

Although SoloJam involves human interaction, in order for behavioural equivalence across the participants, we consider experimenting with artificial agents in this section. Moreover, an artificial agent can be designed to behave as a participant with little or no musical training with little effort. As such, we get artificial participants behaving in a specified manner operating the respective nodes similarly. This allows for evaluating a base line system, which is a system that must work when all the nodes are operated by participants with little or no musical training. Otherwise, one could argue that a human operator may influence the system towards having the requisite functionality, even if the system did not work. Thus, artificial agents allow for controlling the nature of the interaction of the operator, removing human induced functionality into the circulation of solos, which may be hard to account for.

We primarily investigated the effects of the utility function specification within SoloJam, considering the manner in which knowledge about the soloist node affects the circulation of solos within the group of participating nodes. Since we are only interested in the effect of the utility function on the circulation, fixing other factors which may influence the circulation, makes a plausible case for using artificial agents with a fixed behaviour. In this study, these artificial agents use the notion of mutation to generate the bit strings that represent rhythmic patterns. This mutation is such that the agents can flip each bit in their bit string with a probability $1/\lambda$, where λ is the length of the rhythmic pattern. In so doing, the agent generates a new rhythmic pattern, which is a mutation of its old rhythmic pattern. This mutation based rhythmic pattern generation process is essentially used by bidder nodes in every bar they have to bid in, as they search for slight variations of the soloist's rhythmic pattern. We limit our study with agents to the case where, once the soloist starts playing their solo, they do not change their rhythmic pattern for the duration of the solo (which should be some bars long), i.e. a solo is made up of repetitions of the same rhythmic pattern. This limitation allows us to clearly observe if the bidder nodes are indeed able to search for slight variations of the soloist's rhythmic pattern, which, upon winning the auction, they eventually play.

Note that the coefficient a, within Equation 1, signifies the importance (in terms of a weighting) that a node gives to the distance D_l between its current rhythmic pattern and the soloist's current pattern. Setting the value of this coefficient to 0.0 within a node, allows for switching off knowledge about the soloist node. In essence, the node then only knows its own rhythmic pattern and the duration it has played a rhythmic pattern when acting as a soloist. Setting a to a positive value makes the node consider knowledge about the soloist. We take $a = 0.0$ and $a = 1.0$ in order to explicitly investigate the effects of not disclosing and disclosing respectively, the knowledge about the soloist node to other nodes. Note that the soloist node remains unaffected from a change in the value of a, because D_l is zero for it, thus making a irrelevant.

We can now detail the effects of such knowledge within the workings of SoloJam, specifically looking at the nature of the decentralised circulation of solos and also the coherence that can be achieved in the generated piece of music. We first look at the piece resulting from the system, and then provide a discussion based on the evolution of the utilities of the nodes, both with respect to such knowledge. For our study, we use the following parameter settings: *Rhythmic pattern length* (λ) $= 8$, $\epsilon = 0.1$, *Node count* $= 3$, $c = 2$, $b = 0.05$.

Observations About the Resultant Piece. Figs. 2 and 3 show snapshots of rhythmic patterns that are generated when the agents play SoloJam, under two specific cases, one where bidder nodes do not consider using knowledge about the soloist's rhythmic pattern when evaluating the utility derived from their own rhythmic patterns, and the other where they do so. These two cases are realised by $a = 0.0$ (Fig. 2) and $a = 1.0$ (Fig. 3) respectively within the part of the utility function (Equation 1) used by each node for this evaluation.

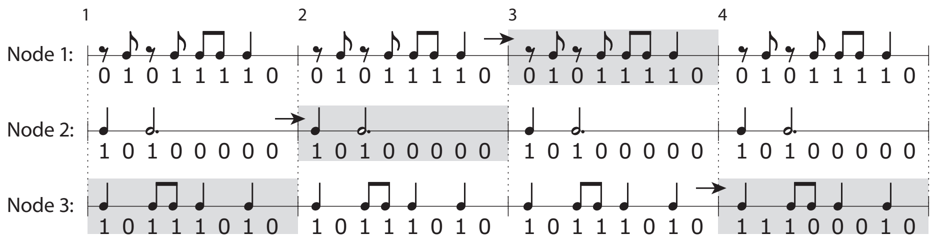


Fig. 2. Snapshot of the rhythmic patterns when $a = 0.0$. There is maximal circulation of responsibility of playing solos (at every bar). The musical output is incoherent as there is no mutation towards closer rhythmic patterns by bidders. In effect, there is no active participation via mutation. Enabling participation is not effective.

These figures show the rhythmic patterns as bit strings and in music notation, for each node in the system. Since we have 3 nodes, 3 lines with bit strings and music notation correspond to each node, as indicated. These lines can be read from left to right for each node. At the end of the 3 lines, the reader can continue at the left of the next 3 lines (see Fig. 3), and so on. Each bar is clearly marked as enclosing the respective rhythmic patterns (of length 8 bits) for each node. The shaded regions denote the current soloist. An arrow between bars denotes a rhythmic pattern being sufficient for a transfer to happen. Mutations within a pattern from a previous bar for a node are denoted by dotted circles. The numbers above bars are bar numbers, wherein a range means that the rhythmic pattern is repeated for all the bars in that range, without any mutations or transfers.

For the case with $a = 0.0$, the 3 nodes do not mutate their respective rhythmic patterns over successive bars. Moreover, the transfer of control of responsibility for the solo happens in every bar, as indicated by the shaded regions in the figure. For the case with $a = 1.0$, we can see a more interesting final result: it can be seen that at bar 21, the rhythmic pattern with which Node 1 bids in the

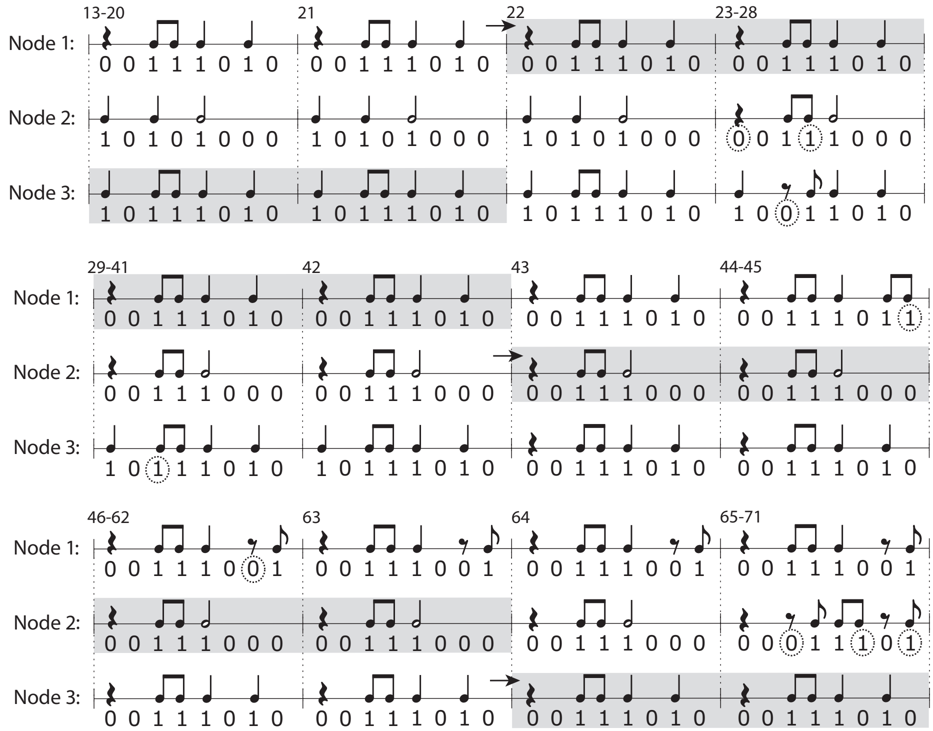


Fig. 3. Snapshot of the rhythmic patterns when $a = 1.0$. Decentralised coherent control is exhibited. The transfer of responsibility of playing solos happens after the soloist having played their rhythmic pattern for some bars. Coherence results from the nodes actively searching for closer variants, via mutation, of the soloist's rhythmic pattern, and the closest rhythmic pattern being played by the respective bidder, provided the bidder wins the auction. Enabling participation is effective.

auction held by Node 3 (the then soloist), differs less (different by 1 bit) from Node 3's rhythmic pattern, as compared to the rhythmic pattern associated with Node 2 (different by 2 bits). Node 1 wins this auction in this bar, and from bar 22 onwards until bar 42, plays its rhythmic pattern. At bar 23, Node 2 and 3 mutate their rhythmic patterns, a further mutation happening at bar 29 for Node 3. Note that in bar 23, Node 3 comes up with a rhythmic pattern that is 2 bits different from the soloist, as compared to its rhythmic pattern in bar 22. This is because the rhythmic pattern in bar 22 has its value reduced to zero in the following bar in accordance with the utility function. Thus, any mutation of that rhythmic pattern in the bar following that will have a value greater than zero. As such, this mutation will replace the previous rhythmic pattern. Other than such a situation, the mutations that are generated over time take the nodes closer to the rhythmic pattern of the soloist, as can be seen in the figure. In bar 42, there is a tie between Node 2 and Node 3, which is broken randomly and

Node 2 takes over the responsibility of playing its rhythmic pattern as a solo. In bar 44, and then in 46, Node 1 mutates towards a closer variant of Node 2. This is followed by a tie again in bar 63, which is then broken randomly in favour of Node 3. In bar 65, Node 2 mutates away from Node 3, again due to the nature of our utility function, as described above. It is clear from Fig. 3 that the nodes actively search for closer variations of the soloist's rhythmic pattern via mutation, and the node (that is sometimes decided upon by a tie break) with the closest match, becomes the soloist in the next bar, provided this node wins the auction.

Discussion Based on the Utilities of Nodes. Fig. 4 plots the utilities that each node derives from its respective rhythmic pattern in each bar, as individually evaluated by these nodes using the utility function described in Section 3.1, and the sum of these utilities. These figures correspond to the snapshots of the pieces from the system (Figs. 2 and 3).

As observed with the corresponding piece (Fig. 2), for the case when $a = 0.0$, the transfer of control happens at every time step, thus a soloist node only ever plays its rhythmic pattern for one bar. The auction held by the soloist immediately leads to the bidder who was not the previous soloist, to take over the control from the soloist, thus becoming the new soloist, but for only one bar. This happens due to the nodes not considering using the knowledge about the soloist's rhythmic pattern, and thus having a utility and bid of $u = c = 2.0$, if they were not the soloist in the immediate previous time step. The process of such transfers of control carries on. Note that all possible mutations of rhythmic patterns for a bidder who was not the soloist in the previous bar, have the same value of 2.0. Thus, the agent has no pressure towards coming up with bids of higher value. We see however, that there is not enough time for the bidders to search (via mutation) for new rhythmic patterns. This is because when a mutation results in a new rhythmic pattern, the previous rhythmic pattern has its value equal to the value of this new rhythmic pattern at all times, be it in the round after the round in which the node was the soloist (the value for both rhythmic patterns is 0.0 in this case), or the rounds after this (value is 2.0). As such, the rhythmic patterns with which the nodes started with in the first bar, either as a soloist or bidder, remain as the rhythmic patterns associated with these nodes forever, as can also be observed in the corresponding piece for the $a = 0.0$ case (Fig. 2). In effect, coherence remains an issue, since the initial rhythmic patterns of the nodes will not necessarily be slight variations of each other. Moreover, the fact that nodes play their rhythmic patterns for only one bar, goes against the whole idea behind playing solos, unless of course playing for only one bar were to be a requirement from the composition. Most importantly, however, the agents are not able to actively participate to explore the composition. The current utility function with $a = 0.0$ is thus not suitable for being used when participants are to play rhythmic solos within a band-like setting. Using this, there would be maximal circulation of control (at every bar, thus no solo being played), the musical output will be incoherent, and there would be no active participation.

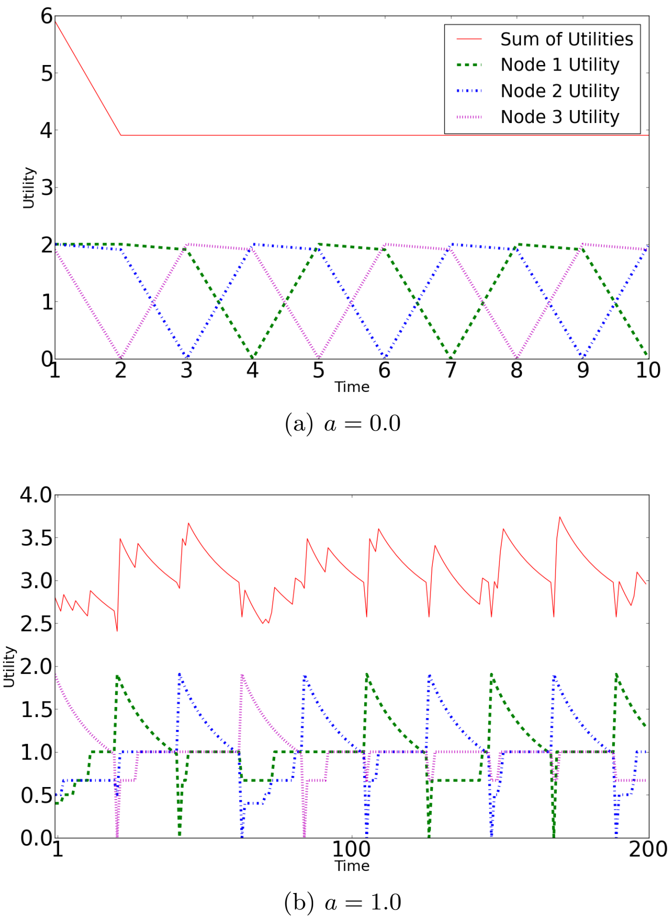


Fig. 4. Utilities of nodes (a) without ($a = 0.0$) and (b) with ($a = 1.0$) knowledge about the soloist's rhythmic pattern.

For the case when $a = 1.0$ however, the playing of solos and transfer of control over time happens in a more favourable manner with respect to the envisaged goal of decentralised interactions producing a resultant globally coherent piece of music, or decentralised coherent circulation. Fig. 4(b) shows spikes in the node utilities, which indicate the start of nodes playing their rhythmic pattern as solos, and these utilities depleting over time. Whilst the soloist node's rhythmic pattern utility depletes, the bidder nodes have their artificial agents search towards slight variations of the soloist's rhythmic pattern, as indicated by the increase in their utilities over time. As a result, the soloist gets to play its rhythmic pattern as a solo for some time and then hands over control to the bidder managing to search and bid to play the closest variation of the soloist's rhythmic pattern, as observed with the corresponding piece in Fig. 3. The flat regions in the utility graphs (Fig. 4) indicate agents associated with bidder nodes having found rhythmic patterns at a distance D_l of $\epsilon\lambda$ from the soloist's rhythmic pattern. Note that there are always multiple rhythmic patterns that the agent could come up with, all of which differing by distance $\epsilon\lambda$, or indeed differing by a given distance from the soloist's rhythmic pattern, which can be seen as the flexibility in the composition that may be explored by a participant based on their preferences, e.g. preferring one rhythmic pattern over another, even though these rhythmic patterns yield the same utility as assigned to them by the device. The artificial agents mimicking participants have thus been enabled to play rhythmic solos in a decentralised and coherent fashion via the consideration of a utility function that takes the knowledge of the soloist's rhythmic pattern into account. The agents must now, as compared to the case where $a = 0.0$, actively participate to search for a rhythmic pattern, and upon being the soloist, play them. The solos that get played adhere to the composer defined boundaries as defined in Section 2, and the system maintains a decentralised coherent circulation. As mentioned before, having a decentralised coherent circulation shows that the system enables the agents to play through the composition effectively.

It would be interesting to consider how the increase in the number of nodes affects the resultant behaviour of the system, with nodes possessing a utility function such as the one defined in this paper, for the case with $a = 1.0$. We leave this as future work.

4.2 SoloJam with Human Users

SoloJam with human participation has also been implemented. As mentioned before, human participation involves a human user using a device that allows for the exploration of the composition. The iPod Touch devices that we use for human participation, one for each human user, have a thread each in the Computation module representing them. Upon shaking the device, the signals from this shaking are received by the associated thread and converted into a rhythmic pattern, which becomes the candidate rhythmic pattern for the next bar for the node in question. The human user, unlike the agent, may change the rhythmic pattern in any bar when part of a soloist node.

A video of SoloJam with human participation can be found online[4]. The video shows three people using iPod Touch devices to play through the piece, playing rhythmic solos as soloists, and bidding for playing slight variations of the soloist's current rhythmic pattern as bidders, whenever a conflict arises. Fig. 5 shows a labelled screenshot of this video. The Max/MSP patch (our Sound interfacing module described in Section 3.2) in the background visualises the

[4] `http://fourms.uio.no/downloads/video/SoloJam.mov`

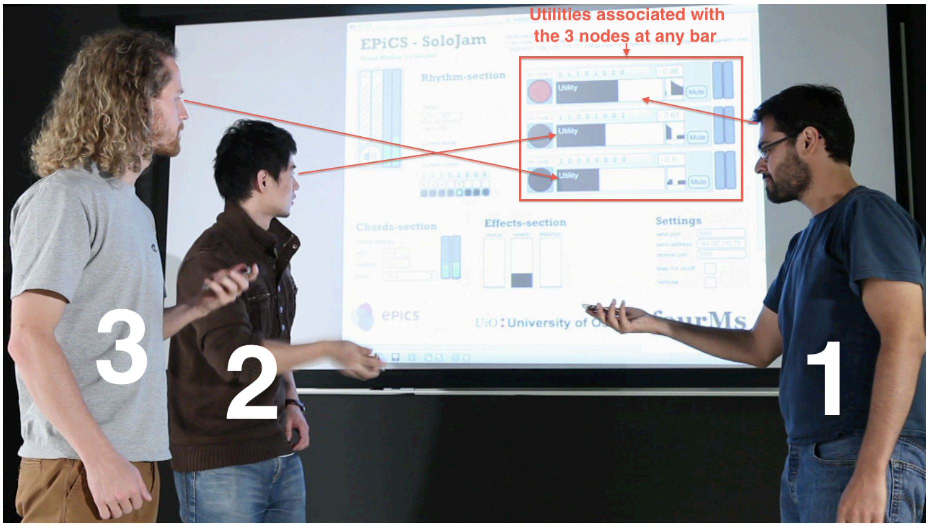


Fig. 5. Labelled screenshot of the video of SoloJam with human participation.

utilities (three horizontal bars at the top right part of the patch) for each node. The top horizontal bar is the utility associated with the person on the right (Node 1). The middle horizontal bar is associated with the person on the left at the back (Node 2). The lower horizontal bar is associated with the person on the left in the front (Node 3). The reader is advised to focus on the rhythmic patterns resulting from the users shaking their devices and the horizontal bars representing utility for each node. In the video, it is possible to see that the transfer of responsibility happens when the soloist node's utility goes below the utility of the highest bidder node. Furthermore, the bidder nodes have their utilities increased, as and when they come up with closer (in terms of hamming distance) variations (but not exact copies) of the soloist's rhythmic pattern.

The circulation of solos in this particular video follows the sequence: *Node 1* → *Node 2* → *Node 3* → *Node 2*. Node 1 starts off with the control of the rhythmic feature in the music as a soloist and sets a rhythm. The node then joins in with the others in influencing some other musical features not needing conflict resolution[5]. In every bar, the soloist holds an auction. Thus, the device which is part of Node 1 holds an auction, as long as Node 1 is a soloist. At some point during the course of the piece, Nodes 2 and Node 3 individually decide on controlling the rhythmic feature, whilst Node 1 still has control, resulting in a conflict. The conflict gets resolved as Node 1 receives Node 2's bid with a utility value higher than that offered by Node 3, making Node 2 the soloist. After becoming the soloist and setting the rhythm, Node 2 joins in with the others to influence other musical features. While Node 2 still is the soloist, Node 3

[5] These other features are part of an extension to the work being presented in this paper, so we limit their discussion.

feels the urge to become the soloist and starts shaking the device. This leads to Node 3 producing rhythmic patterns which are closer matches to Node 2's rhythmic pattern, increasing Node 3's bids for becoming the soloist. Node 3 takes over eventually, while the others continue influencing other musical features. Moving further with Node 3 as the soloist, others then decide to control the rhythm as well, with Node 2 again taking over as the soloist. This is followed by all participants playing other musical features. Thus, our approach encourages human users to come up with rhythmic patterns that are slight variations of the soloist. The closest bidder is then aptly rewarded by this bidder becoming the next soloist, once this bidder wins the auction held by the current soloist.

4.3 The Relationship Between SoloJam and Typical Modes of Communication in Bands

Avoiding conflicts in a band with trained musicians can take various forms. Consider the case of a jazz band. In such a band, the band members communicate both before and during the performance [12]. Communication that occurs during the performance is primarily non-verbal, which may refer to bodily gestures, such as lifting the instrument, or even a barely noticeable nod or a glance. Further, non-verbal communication between musicians occurs through the musical sound, e.g. by responding to phrases by other musicians with a "matching" phrase, or by adjusting the intensity, for instance to signal the transition to a new section.

One particular communication task in a jazz band is to signal the beginning of a solo. The order of soloists and/or the start time and duration of each solo may be pre-planned, but this is not necessarily the case always — another possibility is that the soloist and the time of the solo is decided during the performance. In this case, it is quite common that a band leader decides when the next solo be played and who might play it, and signals this through an indicative gesture [1]. This can be seen as a form of centralised control, and is indeed a predominant mode of communication. It is clear that this mode of communication violates the fundamental requirement of decentralised control within the interactive musical scenario considered in the paper. If indeed one participant were to control the circulation of solos, that participant would need greater musical training than the others, which further violates our assumption about the participants of our system having little or no musical training.

On the other hand, there are bands with trained musicians which have no pre-defined leader. In this case, the selection of the next soloist may be based on a musical or gestural initiative of the current or would be soloist, or based on the initiative of one or more (e.g. forming a consensus) of the other members of the group. It is also possible for certain members in the band to have a greater impact on the choice of the next soloist, as compared to others. In addition to such gestural cues, the notion of "empathetic attunement" [12] can also be a possibility. This notion suggests that people can get musically attuned to each other in time, and can thus seamlessly and coherently play solos, as if they individually know what the group wants, thus spending very little time in explicit communication. One can view these modes of communication between band members, as a decentralised way of resolving conflicts which could arise if such communication were not present. Such decentralised control requires musical training. However, a group that is not trained musically will need assistance for decentralised interactions, which is where our work comes in.

Remarkably, the modes of communication mentioned above have much in common with interaction mechanisms that have thoroughly been investigated in the social and economic sciences, and indeed markets, albeit not within a musical setting. For example, if gestural initiatives represent expressions of interest in performing solos, one could see them as equivalent to bidding for the position of soloist. If the initiatives are a form of consensus, they can be seen as votes resulting in a socially acceptable choice of soloist. Thus, the commonalities between the social setup of a band, and social interaction schemes studied in the social and economic sciences, have much to offer towards formulating interaction schemes for a band consisting of members with little or no musical training, in order to realise decentralisation. Such schemes forming part of the intelligence in devices helps enable the requisite type of participation, as we show via market-based control. It would indeed be interesting to explore other interaction schemes, for example, voting to choose a soloist, or indeed other auction types, as part of future work.

4.4 Flexibility of SoloJam

The presented system is extendable beyond the currently presented implementation, both in terms of musical output and the type of responsibility for which conflict resolution via auctions may be useful. For example, the solos in the present system could take the more traditional form of melodic phrases rather than a sequence of rhythmic patterns. This would require small changes in the representation of musical output, and correspondingly in the utility function, e.g. replacing the bit-string by a string of integer midi notes, or real-valued frequencies. Further, instead of bidding for the position of soloist, the bid could be for the control of some larger structure in the music. For example, bidding to decide a chord progression with a duration of multiple bars, or to decide the downbeats of a drum-pattern. We have already implemented the latter as one extension to the SoloJam system presented in this paper. An online video example[6] shows three agents bidding for control of a single drum module. As in the system presented throughout this paper, the musical output is represented by bit-strings, but rather than controlling each individual beat directly, the bit-strings are mapped to downbeats in the rhythm. A heuristic is implemented to generate full drum patterns from the downbeat pattern. As a consequence, a dynamic musical output is obtained, while SoloJam ensures coherence in the downbeat patterns as the control circulates across nodes.

[6] `http://fourms.uio.no/downloads/video/SolojamDownbeat.mov`

5 Conclusions

We have outlined and discussed the issue with enabling participants with little or no musical training to play together in the interactive music system SoloJam. An approach inspired by the economic sciences, in particular markets, specifically considering the concepts of auctions and utility, is proposed in order to address this issue. Nodes that possess the capability of evaluating the deservedness of being able to take on the responsibility of playing the solo starting in the next bar (via a utility function), and auctioning and bidding capabilities, are shown to exhibit decentralised co-ordination when circulating solos in SoloJam. Furthermore, a careful design of the utility function enables participants (simulated by artificial agents) to come up with an output that is musically coherent. This is highlighted by the manner in which the agents, as bidders, search towards higher utility deriving variants of the soloist node's rhythmic pattern. These variants, in fact, are slight variations of the soloist node's rhythmic pattern. We further exhibit human user participation within SoloJam supporting our approach. In effect, decentralised coherent circulation that results from our market-based approach, demonstrates the effectiveness of the approach towards enabling participation within SoloJam. Having proven the concept, our next step will be to conduct usability tests with human participants. In addition to testing the system with participants with little or no musical training, we are also interested in seeing how music students and professional musicians interact with the system.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n^{o} 257906 and the Norwegian Research Council through the project Sensing Music-related Actions (project n^{o} 183180).

References

1. Bastien, D.T., Hostager, T.J.: Jazz as a process of organizational innovation. Communication Research 15(5), 582–602 (1988)
2. Bunce, G.: Electronic Keyboards: their use and application in Secondary School Music teaching. Master's thesis, Royal Holloway, University of London (2005)
3. Clearwater, S.H. (ed.): Market-based Control: A Paradigm for Distributed Resource Allocation. World Scientific, Singapore (1996)
4. Essl, G., Rohs, M.: Interactivity for mobile music-making. Organised Sound 14(2), 197–207 (2009)
5. Fishburn, P.C.: Utility theory. Management Science 14(5), 335–378 (1968)
6. Goto, M.: Active music listening interfaces based on signal processing. In: IEEE International Conference on Acoustics, Speech and Signal Processing. vol. 4, pp. 1441–1444 (2007)
7. Jennings, K.: Toy symphony: An international music technology project for children. Music Education International 2, 3–21 (2003)

8. Leman, M.: Embodied Music Cognition and Mediation Technology. MIT Press, Cambridge, Massachusetts (2008)
9. Miller, K.: Schizophonic performance: Guitar hero, rock band, and virtual virtuosity. Journal of the Society for American Music 3(4), 395–429 (2009)
10. Moens, B., van Noorden, L., Leman, M.: D-jogger: Syncing music with walking. In: Sound and Music Computing Conference. pp. 451–456. Barcelona, Spain (2010)
11. Rocchesso, D.: Explorations in Sonic Interaction Design. Logos Verlag, Berlin (2011)
12. Seddon, F.A.: Modes of communication during jazz improvisation. British Journal of Music Education 22(1), 47–61 (2005)
13. Shiga, J.: Copy-and-persist: The logic of mash-up culture. Critical Studies in Media Communication 24(2), 93–114 (2007)
14. Small, C.: Musicking: The Meanings of Performing and Listening. Wesleyan University Press, Hanover, New Hampshire (1998)
15. Stigler, G.J.: The development of utility theory, Parts I and II. Journal of Political Economy 58, 307–327 and 373–396 (1950)
16. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance 16(1), 8–37 (1961)
17. Wright, M.: Open Sound Control: an enabling technology for musical networking. Organised Sound 10(3), 193–200 (2005)