Improvisation Planning and Jam Session Design using concepts of Sequence Variation and Flow Experience

S. Dubnov
CRCA/Music, UCSD
sdubnov@ucsd.edu

G. Assayag
IRCAM
gerard.assayag@ircam.fr

ABSTRACT

We describe a model for improvisation design based on Factor Oracle automation, which is extended to perform learning and analysis of incoming sequences in terms of sequence variation parameters, namely replication, recombination and innovation. These parameters describe the improvisation plan and allow designing new improvisations or analysis and modification of plans of existing improvisations. We further introduce an idea of flow experience that represents the various improvisation situations in a mental space that allows defining interactions between improvisers in terms of mental states and behavioural scripts.

1. INTRODUCTION

In the field of music improvisation with computers there has been recently a great advance in music modeling that allows capturing stylistic musical surface rules in a manner that allows musically meaningful interaction between humans and computers. One of the main challenges in producing a larger form or improvisation of significant span is in creating “handles” or means of control of musical generation, so that the result becomes more then accidental play of imitation and response. In this paper we try to identify the principles and possible methods for creating a meaningful play between computers and human improvisers. The main tasks at hand are the following:

1. Defining meaningful controls for music material generated by computer.
2. Allowing machine analysis and recognition of these parameters.
3. Characterization of the overall musical experience that is created as a result of specific improvisation choices.
4. Defining rules of interaction between players (human and machine) that enhances or inhibits (supports or contradicts) the choices of the different participants in the improvisation.

As a model for machine improvisation method we choose Factor Oracle (FO), an automaton that effectively captures all sub-phrases (factors) in a sequence. This automation is extended so as to allow production of variations on a template sequence with control over the amount of randomness or innovation and analysis of new material so as to recognize which FO and which segment within FO was used to produce a variation, to what extent the variation differs from the reference template and the rate of innovation versus replication and recombination of the different materials represented by FO.

Equipped with these improvisational and listening tools, we proceed to construct an interface for communication between the improvisers in terms of higher order notions of emotional or related cognitive descriptive characteristics. One of the problems in designating such a mapping between system or data parameters and cognitive categories is the lack of clear definition and agreement on emotional terminology and representation. Various schemes have been proposed, such as basic emotional categories, emotional dimensions or spaces, grouping according to cognitive eliciting conditions etc.

In this work we use one such model, which relates emotions only indirectly to mental states using a notion of experience flow. The concept of Flow Experience has been introduced in psychology to describe an optimal experience of humans when dealing with tasks that involve certain balance of task skills and task complexity. This concept has been recently applied to design of media presentation, such as choice of levels in computer games. The idea of flow describes the overall engagement of a player in dynamic experience, such as learning or playing computer games [4]. We shall explain in more detail the flow model later on in the paper. It should be noted that our model of flow differs from the original flow idea in terms of the parameters and axes of the flow space. The purpose of using flow in our work is to allow identifying and labeling improvisation states for designing musical interactions. Before proceeding to discuss these aspects of our model we introduce the methods for sequence modeling and the improvisation model that are used in this work.

2. THE MODEL

2.1 Sequence Modeling, Improvisation and Analysis

Machine improvisation and closely related style learning problems usually consider building representations of time-based media data, such as music or dance, either by explicit coding of rules or applying machine learning methods. Stylistic machines for dance emulation try to capture movement by training statistical models from a sequence of motion-capture sequences [3]. Stylistic learning of musical style use statistical models of melodies or polyphonies to recreate variants of musical examples [5]. These and additional researches indicate that emulation of particular behaviors is possible and that credible behavior could be produced by a computer for a specific domain.

Deterministic model for sequence modeling and improvisation was introduced in [2]. The model employs an automation called Factor Oracle (FO) [1] that is computed incrementally and efficiently represents all factors in a sequence with least number of states and linear number of transition called factor links. Beyond its compactness, FO also provides, via its construction, a set of pointers, called suffix links, from every point in the sequence to its last repeated factor. The transition and suffix links allow "traveling" from any point in the sequence to future and past locations where similar factors reside. Following factor links replicates factors from the sequence. Following suffix links creates a recombined sequence in which a suffix is known to belong to the original sequence, a process close to variable memory Markov models. Figure 1 shows an example of FO of the sequence ABABABABAABB. Traversing this FO can generate new sequences. For instance, by following two factor links, one suffix link, and one factor link, one generates the sequence ABBAB. Same sequence can be generated also using other sequences of links (such as three factor links, one suffix link and then two factor links). Musical FO's have been implemented in real-time environment called OMAx (OpenMusic+Max) by one of the authors and Marc Chemillier, and successfully experimented in real-life, live situations with Jazz performers.

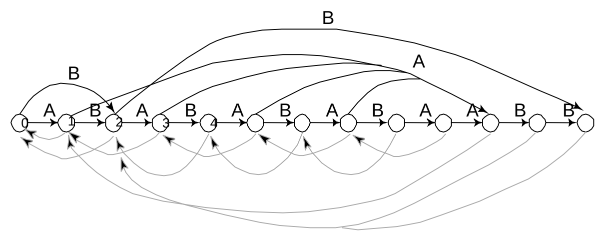

Figure 1: Factor Oracle of a sequence. Black arcs represent the forward transitions (factor links). Grey arcs are the suffix links

2.2 The Improvisation Model

Our model consists of one or more players, set of template sequences, which may preexist, or may be learned on the fly by an FO listening to a player, and a "composition design" that represents the improvisation parameters and / or sequence of mental states and interactions according to these states, possibly changing in time. An individual player consists of an automation (FO) that efficiently captures the improvisational and analysis possibilities with respect to the set of template sequences. As will be explained below, one of the goals that we try to accomplish in this work is to find a mapping between improvisation parameters and states of the improviser, which we shall call "mental states". This mapping shall be done using a flow diagram, which is a modification of the flow experience model, as will be explained in the next paragraph.

In the case of a single improviser, the player operates according to a predetermined sequence that specifies explicitly the improvisation parameters or their mapping through mental states. In the case of two improvisers, a communication exists between the players by exchanging musical sequences and inferring the mental states of each player. Interaction rules specify the logic of the player reactions to each other's mental states.

2.3 Analysis Module

The analysis of an incoming sequence is done relatively to the available templates (modeled by FO's), by estimating some sort of similarity with these templates. The FO's may have captured previous sequences played by the performers during the same session, or they may relate to formerly archived sessions, or even to pieces in the repertoire.

The general situation is: player B receives a new sequence S from player A. B tries to relate this sequence to an FO F which models a template sequence T. So B runs S along F trying to devise to which extent S is build by repetition of T, by recombination of some T material, or by introducing brand new material not present in T. Of course the three situations may happen in different proportion and this is precisely what is going to be measured as "improvisation parameters". We shall name these parameters "replication rate", "recombination rate", and "innovation rate".

The FO structure allows this computation in a very straightforward way, by running the incoming sequence along the FO states, starting in the initial (leftmost) state, then following the arrows that bear the current popped symbol in S. Factor links are followed as much as possible and they account for the replication rate. When no factor link is available for the current popped symbol, suffix links are tried, and if available, they account for the recombination rate. If both fail, then there's a symbol in S that is brand new with regards to T. It accounts for the innovation rate, and in this case we reset the current state to the initial state. These three parameters are normalized by the sequence length, and by construction their sum amounts to 1.

When a series of successive incoming phrases are analyzed this way, the improvisation parameters ordered in time may be considered as an estimated improvisation plan. For example, if T is a Jazz standard theme, and S is a new captured improvisation that is based on T, the estimated plan could be something like: use the exact theme during *x* phrases, then introduce some recombination variants during *y* phrases, then increase recombinations more and more up to a climax after *z* phrases, then create a surprise by introducing new material, etc. Another way to specify the same plan in probabilistic terms is: use the exact theme with replication probability *Prep* close to 1, recombination probability *Prec = 1-Prep* and no innovation (*Pinn = 0*) during *x* phrases, then increase *Prec* during *y* phrases, keep increasing it until climax after *z* phrases and then introduce new material by assigning high probability to *Pinn*, and so on. It should be noted that using the probabilistic interpretation, same plan results in different improvisations (different instances of same plan).

A particular case that must be considered is "learning on the fly", described as follows: Suppose we do not have any template at hand and we start from an empty FO. A performer plays an improvisation, which is incrementally learned into this FO. Our system would like to estimate his improvisation plan: Each incoming phrase is analyzed with respect to FO *before* the phrase is learned (otherwise, the replication rate would be *Prep = 1*). Only after the improvisation parameters have been computed, the phrase is learned into FO.

This method results in a local estimate of our parameters, i.e. we estimate with respect to what has already been played whether the new phrase consists merely of repetition, recombination, or innovation. Moreover, the process is incremental, which means that if another virtual performer listens and plays in parallel, it will have incremental access to our performer's improvisation plan and will be able to take fast decisions (with a granularity size of a single phrase). Of course the same approach can be used offline, either for musicological purposes or for music generation by modifying the original plan.

In this incremental improvisation plan estimation there is an inherent analysis grain that we have called "phrase". In the experiment described below we have used a simple segmentation principle in order to cut an incoming improvisation into such phrases, based on the detection of agogics (durations longer than a given threshold). Other possible segmentation methods could be self-similarity [7] applied to MIDI, or various methods for change detection or switching distribution in sequences [8]. It might be possible to devise criteria for change detection using FO, such as using some distance function to compares between FO's of adjacent phrases, or by observing the rate of growth of forward arrows vs. suffix links incrementally in a single FO. The subject of segmentation using FO is left for future research.

3. EXPERIMENTAL RESULTS

In order to test the analysis module we used an FO to create different improvisations with varying values of improvisation parameter *Prec*. Then we performed analysis of the resulting improvisation using the same FO. The results show that the analysis module achieves quite an accurate estimation of *Prec*. The accuracy of the estimation decreases for higher recombination values, i.e. when many suffix jumps are present in the new sequence. One possible explanation for this effect is the fact that FO accepts a set of expressions that is larger then the set of all factors. In such a case, some of the improvisations would be accepted by the forward transitions in FO, not “noticing” that they were not true factors of the template.

To test the musical applications of the method we carried out two experiments, the first generative and the other analytical. In the first experiment, we started from a known FO modeling a musical piece and two players. We generated several improvisations by one of the player using this FO. By estimating the improvisation parameters in one improvisation, we derived improvisations of the second player characterized by some sort of relation (e.g. symmetry) to the first one. The details of this experiment are specified next.

3.1 Analysis of an actual improvisation

In the second experiment, we start from an empty FO, a Midi recorded piano improvisation by Chick Corea and we compute the improvisation plan. The plot of the patch in Open Music used to analyze the data is shown in Figure 2. The results, which can be considered as an "improvisation plan" by Corea, are shown in Figure 3. The main finding could be described as:

- Replication plot shows a series of peaks with subsets of more or less equidistant peaks.
- Recombination plot shows an arch- or two bell curves overall shape, skewed towards the end (recombines more and more then less and less, with a last ascending slope at the end)
- Innovation “fills-in” the holes left by the replication-recombination parameters, distributed in blocks, with more intense innovations towards the first third of the piece and before recapitulation.

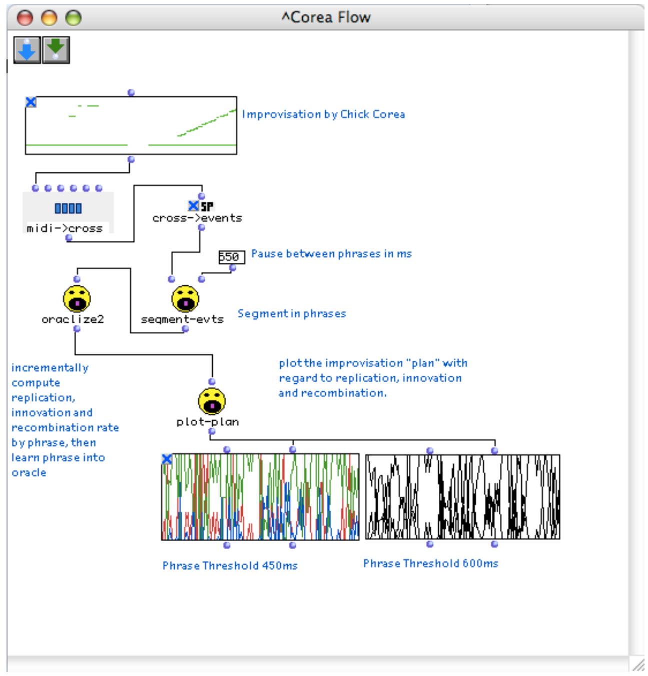

Figure 2: Analysis patch using FO implemented in OpenMusic.

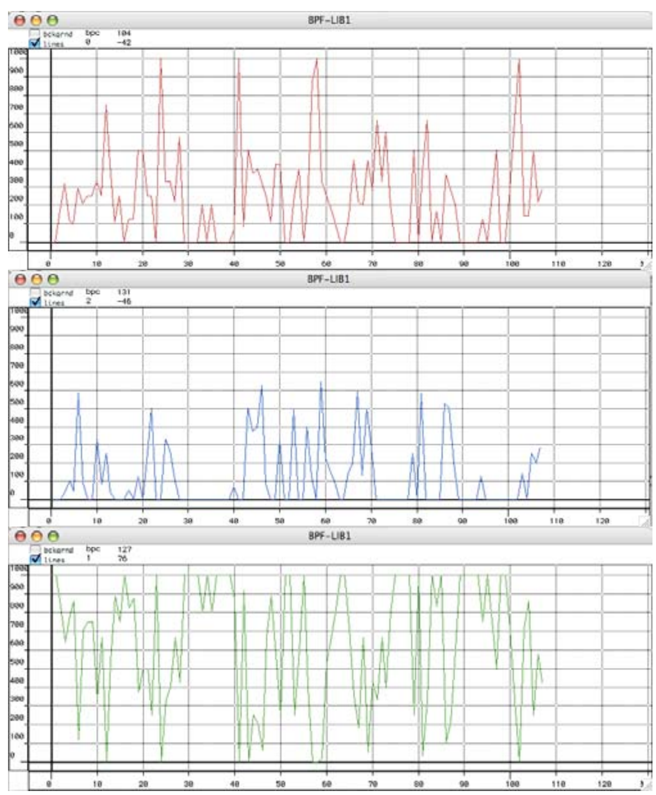

Figure 3: Plots of the Replication, Recombination and Innovation graphs shown in top-down manner, respectively.

This analysis is definitely far from random and it seems to show a form. It is rather stable when varying the phrase segmentation parameters. We can imagine that a virtual performer can follow incrementally that form and take decisions. The musical plan can be stored into the oracle, as part of the knowledge gathered by the learning process.

4. THE FLOW MODEL

Machine improvisation allows producing variations on a template sequence using improvisation parameters for rendering the states of FO automation. The amount of variations produced by each FO is controlled by low-level parameters of replication, innovation and recombination. We use the model of flow experience to group the different parameter combinations into states that might have a musically / psychologically relevant interpretation.

The model of flow is usually used to describe the emotional states in the process of dealing with challenging situation, defines an optimal experience of balancing the player skills against the challenges of the task he is dealing with. The model defines also the emotional, or motivational or mental states of the player for different combinations of skill and challenge. We modify the original flow model to describe improvisation by substituting the original challenge / skill axes by familiarity / emotional force axes, as shown in Figure 4.

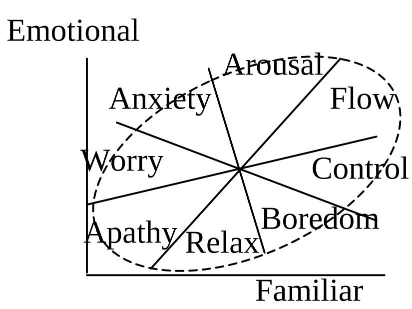

Figure 4: Flow channel and mental states

The terms of emotional force and familiarity are "borrowed" from a recent psychological research on perception of music by listeners [6]. The different combinations of emotional force and familiarity seem to correspond to the different mental states captured by a flow model. For instance, improvisation instances that are highly emotional and familiar seem to induce flow, while materials that are neither emotional nor familiar seem to correspond to apathy state. Materials that are highly emotional but unfamiliar might correspond to anxiety, versus music instances that are familiar but unemotional resulting probably in boredom situation.

The main step required in order to use the flow model in conjunction with the variation parameters derived earlier is to find a mapping between the variation parameters and the flow axes. The following combinations of the replication, recombination and innovation may be suggested as axes of the flow diagram:

- Replication seems to be neutral with respect to emotion and familiarity since exact replication of the sequence does not entail surprise in terms of the type of music material nor in terms of its development or reference to other material in time. In the diagram it would refer to being close to the center of the ellipse.
- Innovation seems to be related to big change in anticipation or surprise due to introduction of new and unfamiliar music material. As such, it might have a component along the familiarity axis, which corresponds to the worry-control or towards the flow-apathy mental dipoles.
- Recombination seems to be more related to more subtle changes in the anticipation or surprise that happen within a certain musical context, i.e. situation where anticipations are established and evaluated in terms of a reference FO. Accordingly, the recombination parameter seems to correspond more to the emotional force axis and operate in the arousal-relax or the anxiety-boredom mental states.

It should be noted that our interpretations of the different axes in relation to the improvisation parameters at this point are intuitive and need to be validated by appropriate psychological testing. We plan to carry out psycho-acoustic experiments in order to validate this model. At this point the mental states may be considered more as metaphors and/or as a possible conceptual framework for software / interface design for improvisation systems.

4.1 Behavior Scripts

Using discrete mental states allows defining logic for musical behaviors and interaction in the context of improvisation. For instance, different improvisation plans can be described in terms of mental states, rather then the variation parameters. Moreover, interaction or responses of different participants in improvisation are more easily described in terms of mental states, creating some sort of behavioral rules or scripts for the different jam session participants. There are several possibilities for defining the jam session behavioral scripts. One possibility is to let the system operate in a totally interactive mode where each player's response is driven by actions of the other players, without following any a-priori improvisation plan or specifying the mental state trajectory for a leading player that other might follow. This creates dynamics that might lead to different final results, depending on the interaction logic. Another option is to specify a sequence of mental states for one of the improvisers and logic of mental responses by the others. One could design an imitative logic (same as the others), contradictory logic (choosing the opposite state), moving forward or leading logic (trying to advance the flow), dragging or slowing logic and etc. Some possibilities for programming such logic include Active Logic [9] or Pi-Calculus with extension to logical constraints [10]. These extensions will be considered in future research.

It could be noted also that other learning schemes like IP[1] or PST [5] could be devised in order to control improvisation flow in please of FO. In such a case the equivalent of replication, recombination and innovation parameters have to be defined using statistical terms, such as distances between the model of improvisation source and the new improvisation sequences. In case when IP is used with bounded memory length, or in case that PST is used, the distinction between replication and recombination is lost, with replication being a recombination with negligible distance from the original and innovation being a situation where the sequence either can not be modeled by LZ or is very improbable for PST.

[1] IP is incremental parsing method based on Lempel-Ziv algorithm for lossless compression. PST is probabilistic suffix trees method related to so called "lossy" methods of sequence modeling or compression.

5. REFERENCES

[1] Allauzen C., Crochemore M., Raffinot M., Factor oracle: a new structure for pattern matching , 1999, In Proceedings of SOFSEM'99, Theory and Practice of Informatics, J. Pavelka, G. Tel and M. Bartosek ed., Milovy, Lecture Notes in Computer Science 1725, pp 291-306, Springer-Verlag, Berlin.

[2] Assayag, G., Dubnov, S., 2004, Using Factor Oracles for Machine Improvisation, Soft Computing 8, pp. 1432-7643, September.

[3] Brand, M., and Hertzmann, A, 2000, Style Machines, In Proceedings of SIGGRAPH 2000, New Orleans, Louisiana, USA

[4] Csikszentmihalyi, M., 1990, Flow: The psychology of Optimal Experience, Harper & Row Publishers, Inc

[5] Dubnov, S., Assayag, G., Lartillot, O., Bejerano, G., 2003, Using Machine-Learning Methods for Musical Style Modeling, IEEE Computer, October 2003, Vol. 10, n° 38, p.73-80.

[6] Dubnov S., S.McAdams and R. Reynolds, Structural and Affective Aspects of Music from Statistical Audio Signal Analysis, Journal of the American Society of Information Science and Technology, (in press), 2005.

[7] Foote, J. and M. Cooper., Media Segementation using Self-Similarity Decomposition., Proc. SPIE, 5021:167--75, 2003

[8] Freund Y., Ron, D., Learning to model sequences generated by switching distributions, Proceedings of the eighth annual conference on Computational learning theory, p.41-50, July 05-08, 1995, Santa Cruz, California, United States

[9] Perlis D., and K. Purang and D. Purushothaman and C. Andersen and D. Traum. Modeling time and meta-reasoning in dialogue via active logic. 1999. Working notes of AAAI Fall Symposium on Psychological Models of Communication.

[10] Rueda C., G. Alvarez, L.O.Quesada, G. Tamura, F. Valencia, J. F. Diaz, G. Assayag, Integrating Constraints and Concurrent Objects in Musical Applications: A Calculus and its Visual Language, Constraints 6, 21-51, 2001.