Jam On: a new interface for web-based collective music performance

Ulysse Rosselet
Information Systems Department
Faculty of Business and Economics
University of Lausanne, Switzerland
ulysse.rosselet@unil.ch

Alain B. Renaud
Experimental Media Research Group (EMERGE)
Bournemouth University
Poole, United Kingdom
arenaud@bournemouth.ac.uk

ABSTRACT

This paper presents the musical interactions aspects of the design and development of Jam On, a web-based interactive music collaboration system. Based on a design science approach, this system is being built according to principles taken from usability engineering and human computer interaction (HCI). The goal of the system is to allow people with no to little musical background to play a song collaboratively. The musicians control the musical content and structure of the song thanks to an interface relying on the free inking metaphor. The design of Jam On is based on a set of quality criteria aimed at ensuring the musicality of the performance and the interactivity of the technical system. The paper compares two alternative interfaces used in the development of the system and explores the various stages of the design process aimed at making the system as musical and interactive as possible.

Keywords

Networked performance, interface design, mapping, web-based music application

1. INTRODUCTION

1.1 Context

Musical performances distributed across remote locations have become increasingly common recently, mostly due to progress in internet connectivity and the increasing availability of bandwidth [25]. Most computer tools used for this purpose are highly specialized pieces of software. These need to be combined and parameterized in a complicated way in order to set up a performance, leading to configurations which are hard to maintain and reproduce. Recent advances in web technologies standards – such as the HTML5 specification [14] – make it possible for users with limited technical skills to engage in rich interactions such as multiplayer online games directly in their internet browser [31].

Building on these recent progresses our research investigates how to engage distant individuals (called musicians in this context) in a collective musical performance using standard day-to-day web technologies and interfaces. Following a design science approach – which involves the design of novel or innovative artifacts and the analysis of the use and/or performance of such artifacts to improve and understand the behavior of aspects of socio-technical artifacts [9], we are exploring this question by building a web-based remote music performance system [4] called Jam On.

The design of such a system has led us to survey which forms of interactions are feasible in this context, and to explore the metaphors relevant to music making under the constraints of an internet browser's graphical user interface. Our system is targeted towards people with little to no music experience. The goal is to enable distant players who typically do not consider themselves musicians to play music together as if they were part of a small ensemble.

In this paper, we first expose the related body of work relevant to our research. We next describe the artifact at the center of our research and its development process, specifically two different alternatives as regards the user interface and its mapping to sound generation; and then explore their implications in terms of musical genre, interactivity and musicality. The development process of our system's interface was guided by approaches from the field of usability engineering [26] and human computer interaction (HCI) [32]. Finally, we discuss the contributions of this research and offer a scope for future work.

NIME'13, May 27-30, 2013, KAIST, Daejeon, Korea.

2. RELATED WORK

2.1 HCI Concepts

Wanderley and Orio [32] have approached the evaluation of input devices for musical expression by drawing parallels to research in HCI. This research field investigates user interface software and tools as well as their usage. To carry out a given task, the user has to monitor the system's status and manually modify its parameters through the use of input and output devices [32]. Typical evaluation tasks in HCI are pursuit tracking, target acquisition, freehand inking, tracing and digitizing, constrained linear motion, and constrained circular motion. These tasks cover the vast majority of possibilities offered by a basic set of computer input devices, i.e. a pointing device such as a mouse or trackpad, and a keyboard. The field of HCI has evolved to focus not only on the task, but to incorporate the experience of the user in the evaluation of the system.

These authors propose a list of contexts related to interactive computer music [32]. We consider the following ones as relevant to our system:

1. Note-level control, or musical instrument manipulation (performer-instrument interaction), i.e., the real-time gestural control of sound synthesis parameters, which may affect basic sound features such as pitch, loudness and timbre.
2. Score-level control, for instance, a conductor's baton used to control features to be applied to a previously defined sequence.

3. Sound processing control, or post-production activities, where digital audio effects or sound spatialization of a live performance are controlled in real time.

2.2 Existing systems

When looking at available products using similar approaches (drawing and online interactions), there are two categories that have inspired us in the design of Jam On. We define them as graphical music creation tools and collaborative digital audio workstations (CDAWS).

Graphical music creation tools. When looking at available products, we found several systems that use similar interaction principles in terms of graphical interface. We limited our review to the manipulation of pitches to create melodies and rhythms, and to the ordering of the song's sections.

- The UPIC System [19] is a graphic composition tablet, developed for composer Iannis Xenakis to compose pieces such as Mycènes Alpha (1978). The tablet allows users to draw shapes to generate electroacoustic music. Highly inspired by UPIC, HighC [13] is a software application allowing users with no musical experience to draw complex musical shapes. These two systems rely on the classical mapping of the pitch-versus-time space: time is represented on the horizontal axis from left to right, and pitch on the vertical axis from bottom to top. Different modes are available, for example to control the various parameters of , a synthesizer.
- Vuzik [24], allows users to paint a musical gesture on a large interactive canvas and acts as a graphic score for controlling external sound generators using the classical mapping of the pitch-versus-time space mentioned above.
- Different Strokes [34] is a computer music software for creating dance and minimal "glitch" styles electronic music. Loops are created by particles traveling along lines freely drawn by the user. Different sounds are available and represented graphically as lines of different colors. Intersections between lines give rise to looping and triggering of sounds as particles propagate along other lines. In this case, the pitch-versus-time mapping depends on the speed of the different drawing gestures and the triggering of segments by traveling particles.
- FMOL (F@ust Music Online) is an asynchronous collective composition software built by Sergi Jorda for trained and untrained musicians [17]. Among the various frontends used to interact with FMOL's sound engine, the Bamboo interface behaves simultaneously as an input by allowing musicians to pluck virtual strings and adjust modulation effects parameters, and as an output for visual "sonic" feedback by showing multiple oscilloscope views of the channels' sounds.
- Public Sound Objects (PSO) is a networked musical system focused on performance aimed at implementing and testing new concepts for online music communication [3]. PSO consists in a shared sonic environment where a central sound synthesis engine is influenced by a network of distributed performers – the clients. Each client controls the motions of a bouncing ball. The resulting musical material is streamed to the clients and diffused in a public installation site.
- Jam2Jam [7] allow users to interact musically in real time via the internet. The system allows users to drag and drop shapes of familiar instruments and loops over a canvas. Jam2Jam doesn't require any previous musical knowledge to be operated.

Collaborative digital audio workstations (CDAWS). An increasing amount of digital audio workstations (DAW) are including a collaborative aspect as standard or can be extended with collaborative features, effectively enabling distributed music production [25]. All these systems rely on a common mapping of the pitch-versus-time space: time is represented on the horizontal axis from left to right, and pitch on the vertical axis from bottom to top. There are a quite a few of them available and listing all of them would be out of the scope of this paper. However, here are a few examples seen as relevant to the development of Jam On:

- OhmStudio [21] is one of the first DAWs to be created with a strong focus on online collaboration. The application allows users to work on a common DAW session by sharing tracks of audio or MIDI via the internet. There is also an online forum allowing participants to post requests for specific services, such as adding a vocal track. Participants can also chat in real time and exchange musical ideas.
- Bitwig Studio [5] is a brand new DAW, currently in beta stage, which includes a new workflow for digital music creation. The DAW will include two collaborative aspects: multi-user LAN jamming, which allows users to share performance data and audio over a local area network and multi-user production over the internet, allowing the sharing of a Bitwig session over a wide area network. In addition, the DAW will also include a modular synthesis system, which will be sharable with other users.
- Cubase [8] is one of the oldest DAWs on the market. The new version, Cubase 7, offers a collaborative function, VST Connect SE, which allows other Cubase users to record and exchange audio tracks straight into the DAW session, inspired by the digitalmusician.net technology [10]. In addition, the functionality offers a chat function, talkback and video streaming. The system uses time stamping of signals so that any user with a decent internet connection can collaborate at sample accuracy.

3. Methodology

Combining technological and aesthetical considerations with aspects related to ergonomics and expressiveness, the field of music and technology research is well suited to a design science investigation approach. In effect, the relationship between information technology artifacts, their usage and their socio-cultural effects is at the center of the design science research paradigm [16]. Following this epistemological posture, our goal is to create knowledge by building artifacts – which can consist of constructs, models, methods, and instantiations [12]. Vaishnavi and Kuechler [29] provide a general methodology for design science research. Following the awareness of the problem – in our case the various challenges of networked musical performance [25] – the suggestion [29] phase of the research explores the potential answers to the problem. These potential answers constitute the starting point of the tentative design [29] which is implemented in the development phase. In our case, it corresponds to the prototype described in this paper. Justificatory knowledge from the fields of music performance and interaction design research have been used at this phase to ensure the rigor of the design process [12]. The development leads to the evaluation and validation phases [29]. For Jam On, we have adopted a three steps approach to evaluation: demonstration, experimentation and live testing.

Firstly, the implementation of the system in the form of a fully functional prototype demonstrates the feasibility of such a networked performance system [30]. Secondly, we had users

with little or no musical background use the web application and they were rapidly able to create patterns and move the song from one section to the other. We are at the moment using these informal tests as a guide for formulating metrics that will be refined during controlled experimentation sessions. These metrics, related to the criteria of musicality and interactivity will be used in the validation of the design choices resulting in the Jam On implementation [30]. In order to separate the aspects related to the interaction design and to the interplay between musicians, we will first assign individual tasks to the participants before adding the collective performance element. The insight gained from studying the interplay strategies between musicians will lead us to the development of musical rules to drive the choreography of collective play. Indeed, a robust set of rules will provide musicians with a reliable starting point to play with the song's structure and content through the interface. The concept is close to the principle of structured improvisation as illustrated by Bailey [2].
With a view to apprehend the effect of the network latency and connectivity issues on musical interplay, we will then test the system in a variety of network conditions.
Thirdly, based on the metrics developed in the first two evaluation phases, we will conduct full scale testing of the system deployed on the internet to understand the engagement of users on the platform. At this stage of our research, we will implement a set of network algorithms to automatically manage latency between remote musicians and assign group membership. Indeed, once the system starts being used by many users at the same time, it will be important to let musicians belong to specific groups and allow the switching to another group based on network conditions according to rules that respect the musical aspects of the performance.
It should be noted that the results of these different evaluation phases will be used to iterate over the artifact's design and modify the Jam On system according to our findings. The iterative nature of the research process is inherent to the design science approach [22]. Each of these phases will lead to communications about the insights gained on networked music performance and interaction design.
The next section exposes the choices made during the tentative solution and development phase of our research.

4. DESIGN OF THE JAM ON SYSTEM

In this section, we describe the issues that our system is to address, formulate quality criteria for the musical performance and explain our design choices.
The musicians can set musical elements and trigger section changes to go from one part of the song to the other (for instance from verse to chorus). The song evolves at two different levels of granularity: the musical content of the section results from aggregation of the **notes** drawn by each musician and the **structure** of the song results from the section changes triggered by the musicians.
The arrangement of the song is mainly pre-determined in terms of rhythm, timbre and harmony and based on a collection of samples loops or tone generators. Each musician can set and manipulate the different parts of the song through to the graphical user interface. Looped samples, tone generators, effects and melodies are triggered or generated in real time according to the musicians' actions and aligned to the song's tempo. The musicians have a shared view of this "macro" song.
The main goal of Jam On is to achieve a good quality of performance and to allow musicians to produce a pleasing and lively performance while enjoying it.
To achieve this goal, we considered two dimensions of the performance: musicianship and musicality one the one hand and interactivity and interaction on the other.

Musicianship and Musicality
In a traditional music playing setting, membership to a performance group is determined by two factors: musicianship defined as the mastery of the physical hurdles imposed by the instrument and musicality defined as how inventive and how novel the musician will be with the instrument [11]. As far as the criteria of musicality is concerned, as defined by Godlovitch [11], the harmonic and rhythmic constraints embedded in Jam On purposely limit inventively and novelty to the different combination of parts, mutes, melodic lines and order of sections. The musicians are more steering the direction of the song rather than playing the different instruments.
In the case of Jam On, musicianship depends on the ability of the musician to master the interface. Musicality depends on the creativity of the musician within the constraints of the provided sonic material.
In order to achieve a pleasing performance in terms of musicality, we identified the following relevant criteria:
Harmonic/melodic coherence: the harmonic content of each musical item follows accepted genre-based rules for notes combination. For example, the set of notes available to the musicians at a certain time must depend on the current harmony of the song. As we follow mainstream music conventions, each harmonic/melodic progression in the system is goal directed and each chord/melody has a strong functional relation [27] with preceding and forthcoming elements to avoid eventual atonal results.
Rhythm/timing precision: the musical events must happen in synchronization with the song's tempo and its subdivisions in order to ensure that the result has a certain "groove". Each part has a certain rhythm or rhythmic entity [27] resulting in its own musically recognizable identity. This requirement of mainstream music makes it impossible to tolerate timing inaccuracies in the reproduction of the musical events.
Structural cohesiveness: the structure of a song (as a sequence of sections) must have an overall uniformity. It should sound like one "song". In this regard, the ordering and repetition of sections must follow patterns where the same sections come back during the performance. For example, structure in the form AABABCAB, where A could be considered as the verse, B the chorus and C a bridge.
Diversity/richness: the performance should be varied. It should not remain static and repetitive whilst keeping harmonic and rhythmic coherence.

Interactivity and Interaction
Interactivity is defined in music as technology which responds to the input of a performer [20]. Our system should support interactivity between one musician and the music software and amongst musicians.
For this purpose we have devised the following criteria in relation to interactivity:
Responsiveness: performers must be able to have a direct and immediate effect on the music they play. The goal is also to achieve an emotional response to the music being played so that each performer feels immersed in their interactions by focusing on specific elements of attention such as melody, dynamics, timbre and rhythm [18].
Feedback: the musicians must know what is happening in the performance. They should have the feeling of belonging to the same band in spite of the restricted interaction possibilities inherent to the networked nature of the performance. The performers should be informed of their status and position within the overall performance either/or through visual or auditory feedback mechanisms. Visual or auditory elements are included in the performance to mark a change of section or

changes in rhythmical structure with a direct relationship *"between actions and feedback"* to *"determine the amount of disruption that results"*[23].
Mutability: the musicians must have the possibility to change the content of the song. Their role should be more than arranging loops. They should be able to change the content and steer the direction of the performance.
Fun/Enjoyment: performers should have fun conducting the performance. As our system proposes online musical interactions, it may be necessary to find a balance between tasks perceived as too difficult (leading to frustration and anxiety) or too easy (leading to boredom) to avoid yielding a negative experience – as suggested by research on massively multi-player online games [28].

Based on the above characteristics regarding musicality and musicianship as well as interactivity and interaction, our system needs to support interaction contexts for interactive computer music in the following ways:
Note-level control/musical instrument manipulation: to achieve an engaging level of interactivity, the musicians need to have the possibility to create musical phrases and patterns on the fly. The musicians set the content of the musical sections in real-time. Existing systems rely mostly on a traditional vertical/horizontal mapping of the pitch/time space (typical of DAWs) or on the real-time adjustment of virtual knobs and sliders (like Jam2Jam) for this purpose.
Score-level control: the musicians decide the ordering of sections in the song in real-time by triggering section changes on the fly. The system embeds rule to ensure that transition between sections occur at a musically acceptable moment based on the content on the concerned sections. It also ensures that musicians have a way to oppose to section changes suggested by others – this is implemented as a time restricted "veto right". Existing systems rely mostly on a generalization of the pitch/time mapping to support the score-level control context: the different musical parts are stacked vertically, and the different sections laid out in their playing order on the horizontal time-axis. It should be noted that, besides the traditional linear view of a piece's structure, the Ableton Live DAW [1] offers an alternative approach to score level control. In effect, the "session view" displays the musical blocks – called clips – in a grid fashion resembling a spreadsheet. The different musical parts constitute the columns of the spreadsheets while the rows represents "scenes" which can be considered sections of the song. The user can trigger clips in the different cells of the grids – building customized, hybrid sections – and trigger full section changes in a nonlinear fashion – jumping to non-contiguous sections – by using the scene trigger button which launches a entire row.
Sound processing control: the content of the musical parts not only consists in notes but relies also on variations of effects parameters to achieve an interesting sound to the song. In our application, this context is intertwined with the note-level control as the axis of the two-dimensional control space can be mapped to pitch, timbre and audio effects at the same time.

Interface and interaction design
The overall interface of Jam On (shown in Figure 1 below) is divided in three zones, from left to right: the sound palette, the musical canvas, and the section overview and section change interface. The musical performance structure is organized in sections. The content of the selected section is viewable and editable in the canvas.
The initial interface designed for this application was not intended for mainstream music. The sounds were depicted as geometric quadrilateral shapes that would be manipulated by the musician. Each sound had a different color. The mapping between the shape and the sonic parameters are shown in Figure 2 below. The left bottom angle of the shape corresponds to the initial pitch of the musical sample; the right bottom angle corresponds to the final pitch of the sample. The duration of the sample is controlled by the length of the bottom side of the shape relative to the canvas's width.

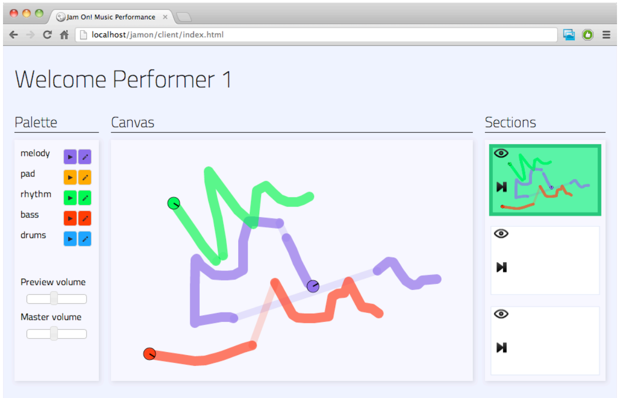

Figure 1. Overall view of the Jam On software

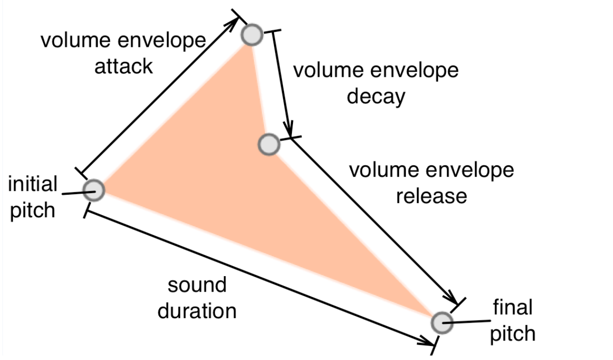

Figure 2. First prototype: mapping between shape and sound

This interaction mode is well suited to the control of sounds, analogous to manipulating pitch and volume envelope on a synthesizer. When it comes to the shaping of individual sounds, it offers a good range of possibilities and the mapping between the visual and musical elements is straightforward. It is well suited to the creation of sound textures with various attack and decay parameters.
However, when it came to a collective performance of mainstream music, this interface posed several challenges. When playing a musical phrase or a sequence of parameter modulations, this interaction mode was found to be tedious to work with. As one has to draw a single shape for each note in the sequence, the number of target acquisitions and constrained motions needed increases significantly. It makes it difficult and time consuming to input musical phrases – hindering the responsiveness of the music software.
In order to improve the way musical phrases can be inputted in the software, a second interface, which is the current front end of the Jam On system, was built upon the freehand inking metaphor – a classical task of HCI design [32]. This approach can be effectively used to input trajectory just by clicking, moving the cursor following the desired trajectory and releasing the cursor when finished, gestures known to be very natural and which exhibit interesting, human variability [34].

We have deliberately chosen to exclude traditional musical interactions (piano roll or guitar fret board for example) so that the system could be more easily accessible for people with little or no musical background. Drawing basic shapes on a canvas is an intuitive process, accessible to all. The musical content resulting from the drawing can be almost instantly heard by the musicians.
Although freehand inking is a relatively simple task to achieve with a computer mouse, mapping the drawing to musical parameters poses several challenges. The traditional approach to mapping freehand drawing to music relies on assumptions stemming from classical music notation: the horizontal position represents time and the vertical position represents pitch. This was not satisfactory for our system because we wanted a complete section to be visible in the browser window.
We will now explore the decisions of mapping freehand lines to sound, as depicted in Figure 3 below. All the lines of a single part (lines of the same color) form a single pattern[1].

- The length of this pattern corresponds to the sum of its segments lengths – equaling the length of the corresponding path. In terms of musical durations, patterns are constrained to power of two of quarter notes (in the case of 4/4 material), i.e. 1 beat, 2 beats, 1 bar, 2 bars, 4 bars, etc.
- The gap between two lines represents a silence in the pattern. The pattern is played as a loop, when the last point of the pattern is reached, it plays from the beginning again.
- The events of the pattern are quantized according to a rhythmic value relevant to the song; for instance, a dance music song with a 4/4 time signature will have its patterns quantized at the sixteenth note (or semiquaver) level.
- A visual indicator of the current position and progression of the pattern is displayed as a small clock at the beginning of the pattern. A full turn of the clock represents one pattern loop.

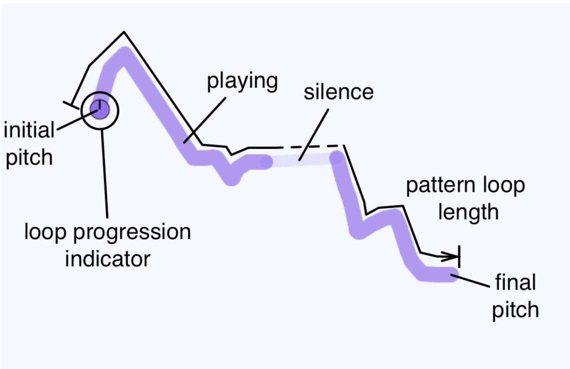

Figure 3. Adopted interface: freehand drawing of musical patterns.

At the sound engine level, we implemented the musical parts in two different ways. Firstly, loops from sound files, which are triggered according to the lines drawn by the musicians and modulated in relation to the pattern composed by these lines. Depending on the parameters of the song, these modulations can include filters, mutes, delays and volume or panning changes. We refer to this type of part as "snaps". Secondly "synths" are musical items rendered in real-time. It consists in a software synthesizer combining elements from frequency modulation and subtractive synthesis. Depending on the type of part – snaps or synth, the mapping of positions along the line to musical elements such as pitch or tones changes. For synth parts, the vertical position controls pitch, while the horizontal position impacts timbral characteristics such as effects balance or frequency modulation amount. For snaps parts, the drawing space is separated in four zones. Each zone corresponds to a variation of the same part – and is stored in a different sound file. This provides a multi-parametric mapping, which is considered to be more engaging for the users when it comes to using the simple controls of a computer interface to perform rich musical tasks, as suggested by the results of Hunt and Kirk [15]. For example, the pad part can vary from a lush string sound to staccato pulsating synth chords.
The musician is free to draw lines in any direction and of any length on the canvas.
What we believe is innovative with this interface is the ability to have different patterns (one for each part) with different lengths coexist within the same interface. Some may loop every bar while others loop every 4 or 8 bars. This is comparable to "session view" in the Ableton Live software (see above). In addition, with Jam On, the content of the pattern is visible at the same time.
To ensure the accessibility and cross platform compatibility of our software, we chose to implement our networked performance system as a web application. The client application is entirely built using HTML5 technologies. It is coded in Javascript, using the KineticJS library to simplify interactions with the HTML canvas, and relies upon websockets for communication and on the Webaudio API [33] for the audio engine. This latter specification offers high quality audio processing and rendering with low latency in the browser. It provides several primitives that are handy for the development of musical applications such as oscillators, sample accurate sound file playback, panners, filters, delays or waveshapers. These elements are referred to as audio nodes and can be connected to each other to build the desired signal path. Programmers can also implement their own audio processing nodes in Javascript.
As regards to the interactions between musicians, it should be noted that only control and coordination messages are exchanged through the network; as in the case of Jam2Jam [6] and FMOL [17] we chose not to exchange audio streams between the performers' machines in order to reduce the potential impact of latency and bandwidth issues. All musical events triggered by the performers are synchronized to the central time transport. For instance, when a section change is triggered, its timestamp is based on the central server time. The client Jam On application then computes the appropriate local time for the transition based on its time offset relative to the server. When a line is added or deleted, the corresponding pattern is triggered at the next acceptable time depending on the pattern length and on the transport position. With these timing safeguards, latency issues can result in a pattern picking up from the second beat instead of the first one for example, but cannot lead to unpleasant timing inaccuracies of the musical output. As far as user experience is concerned, participants of the tests have commented positively on the fast reaction of the system and its solidity in terms of timing and synchronization.
All the audio processing and rendering is done locally in real time on the musicians' computers. At the network level, the system currently relies on a client-server architecture. The modular design of the application allows for different network architectures – for instance peer-to-peer – to be implemented and tested.

[1] a pattern is the musical phrase of a part for a given section of the song. A pattern is composed of lines, which corresponds to the strokes drawn on the canvas. A line is composed of segments, which connect the points composing the line.

5. CONCLUSIONS AND FUTURE WORK

At the music and interactions levels, the Jam On interface has proved to be a good basis for exploring collaborative music drawings and sound mappings via a web interface. Jam On combines some of the functionalities of graphical music creation tools and CDAWs – and lies in between these two categories.

The actual implementation of Jam On validates the first stage of our evaluation by demonstrating the feasibility of such a web-based networked performance system, and initial tests with the interface have proved to offer a high level of interactivity on two aspects: playing and rearranging collectively with other musicians an existing set of tracks intuitively over the internet. The free inking metaphor used proves to be an adequate form of interaction for such a system.

We are currently elaborating the criteria for the experimental evaluation of the Jam On system as explained in the methodology section.

On the content side of the system, we are building a crowdsourcing platform that will allow users to upload and share their own songs to be performed on Jam On. This innovative approach will allow the collection and performance of content coming from a wide variety of cultures.

The explicit adoption of a design science research approach in the field of new interfaces for musical expression is another original aspect of this research.

6. REFERENCES

[1] Ableton Live 9: *https://www.ableton.com/*. Accessed: 2013-02-08.

[2] Bailey, D. 1993. *Improvisation: its nature and practice in music*. Da Capo Press.

[3] Barbosa, A. 2005. Public Sound Objects: a shared environment for networked music practice on the Web. *Organised Sound*. 10, 03 (2005), 233–242.

[4] Barbosa, Á. 2003. Displaced soundscapes: A survey of network systems for music and sonic art creation. *Leonardo Music Journal*. (2003), 53–59.

[5] Bitwig Studio: *http://www.bitwig.com/bitwig-studio*. Accessed: 2013-02-05.

[6] Brown, A.R. 2010. Network jamming: distributed performance using generative music. *Proceedings of the New Instruments for Musical Expression Conference 2010* (2010), 283–286.

[7] Brown, A.R. and Dillon, S. 2007. Networked Improvisational Musical Environments: learning through online collaborative music making. *Music education with digital technology*. (2007), 96–106.

[8] Cubase: what's new?: *http://www.steinberg.net/en/products/cubase/whats_new.html*. Accessed: 2013-02-08.

[9] Design Science Research in Information Systems: 2004. .

[10] Digital Musician: *http://digitalmusician.net/*. Accessed: 2013-02-08.

[11] Godlovitch, S. 1998. *Musical performance: A philosophical study*. Routledge.

[12] Hevner, A.R. et al. 2004. Design science in information systems research. *Management Information Systems Quarterly*. 28, 1 (2004), 75–106.

[13] HighC - Draw your Music: *http://highc.org*. Accessed: 2013-04-28.

[14] HTML 5.1: A vocabulary and associated APIs for HTML and XHTML: *http://www.w3.org/TR/html51/*. Accessed: 2013-02-09.

[15] Hunt, A. and Kirk, R. 2000. Mapping strategies for musical performance. *Trends in Gestural Control of Music*. 21, (2000).

[16] Iivari, J. 2007. A paradigmatic analysis of Information Systems as a design science. *Scandinavian Journal of Information Systems*. 19, 2 (2007), 39.

[17] Jordà, S. 2002. FMOL: Toward User-Friendly, Sophisticated New Musical Instruments. *Computer Music Journal*. 26, 3 (Oct. 2002), 23–39.

[18] Madsen, C.K. 1997. Emotional response to music. *Psychomusicology: A Journal of Research in Music Cognition*. 16, 1-2 (1997), 59.

[19] Marino, G. et al. 1993. The UPIC System: Origins and Innovations. *Perspectives of New Music*. 31, 1 (Jan. 1993), 258–269.

[20] McNutt, E. 2003. Performing electroacoustic music: a wider view of interactivity. *Organised Sound*. 8, 03 (2003), 297–304.

[21] Ohm Studio: *http://www.ohmstudio.com/*. Accessed: 2013-02-08.

[22] Peffers, K. et al. 2007. A design science research methodology for information systems research. *Journal of Management Information Systems*. 24, 3 (2007), 45–77.

[23] Pfordresher, P.Q. 2006. Coordination of perception and action in music performance. *Advances in Cognitive Psychology*. 2, 2-3 (2006), 183–198.

[24] Pon, A. et al. 2012. Vuzik: A Painting Graphic Score Interface for Composing and Control of Sound Generation. *Proceedings of the International Computer Music Conference* (Ljubljana, Sep. 2012).

[25] Renaud, A. et al. 2007. Networked music performance: State of the art. *Proceedings of the AES 30th International Conference* (2007).

[26] Rosson, M.B. and Carroll, J.M. 2003. Scenario-based design. *The human-computer interaction handbook*. J.A. Jacko and A. Sears, eds. L. Erlbaum Associates Inc. 1032–1050.

[27] Scruton, R. 2009. *Understanding music: philosophy and interpretation*. Continuum.

[28] Shim, K.J. et al. 2011. An Exploratory Study of Player Performance, Motivation, and Enjoyment in Massively Multiplayer Online Role-Playing Games. *2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing* (2011), 135–140.

[29] Vaishnavi, V.K. and W. Kuechler 2007. *Design science research methods and patterns: innovating information and communication technology*. Auerbach Pub.

[30] Vaishnavi, V.K. and W. Kuechler 2007. *Design science research methods and patterns: innovating information and communication technology*. Auerbach Pub.

[31] Vaughan-Nichols, S.J. 2010. Will HTML 5 Restandardize the Web? *Computer*. 43, 4 (Apr. 2010), 13–15.

[32] Wanderley, M.M. and Orio, N. 2002. Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*. 26, 3 (Oct. 2002), 62–76.

[33] Web Audio API: *http://www.w3.org/TR/webaudio/*. Accessed: 2013-02-05.

[34] Zadel, M. and Scavone, G. 2006. Different strokes: a prototype software system for laptop performance and improvisation. *Proceedings of the 2006 conference on New interfaces for musical expression* (2006), 168–171.