

Live Patching e Interação Remota: Uma Abordagem Prática Intercontinental com Kiwi

(Live Patching and Network Interaction: An Intercontinental Practical Approach with Kiwi)

Marcello Messina¹, João Svidzinski², Deivid de Menezes Bezerra³ and David Ferreira da Costa⁴

¹Universidade Federal da Paraíba e Núcleo Amazônico de Pesquisa Musical (NAP)

²Université Paris 8, ³Universidade Federal do Acre, ⁴Universidade Federal do Acre

¹marcello@ccta.ufpb.br, ²svidzinski@gmail.com, ³deivideletronecnic@gmail.com,

⁴davidfeemcristo@gmail.com

Abstract

This paper introduces, documents and reflects on an intercontinental live patching experience based on simultaneous remote interaction using the software Kiwi, and that can be subsumed under several features of Ubiquitous Music. The experience involved academic groups from the two Brazilian Federal Universities of Acre and Paraíba, and a working group based at the University Paris 8 in France. The intercontinental simultaneous interaction may trigger reflections on the implications of the presence/absence of the human being, on the implicit patterns of territorialisation reproduced in the context of intercontinental live patching, and on the operative action of mnemonic processes within this practice. In the context of this research, these reflections emerged as part of the development of effective strategies of creative utilisation of the software.

Resumo

Este artigo apresenta, documenta e reflete uma experiência intercontinental de *live patching* baseada na interação remota simultânea usando o software Kiwi. Este estudo pode ser analisado a partir de várias características da música ubíqua. A pesquisa envolveu grupos de pesquisa de duas universidades federais brasileiras, Acre e Paraíba, e um grupo de trabalho baseado na Universidade Paris 8 na França. A interação intercontinental simultânea desencadeou reflexões sobre as implicações da presença/ausência do ser humano, sobre os padrões implícitos de territorialização reproduzidos no contexto do *live patching* intercontinental e sobre a ação operativa dos processos mnemônicos dentro da prática musical digital. No contexto desta pesquisa, essas reflexões emergiram como parte do desenvolvimento de estratégias efetivas de utilização criativa do software.

1. Introdução

Este artigo apresenta, documenta e reflete uma experiência intercontinental de *live patching* e interação remota simultânea usando o software Kiwi. A experiência envolveu dois grupos acadêmicos baseados em três universidades diferentes entre Brasil e França: o grupo de pesquisa *Live/Acc/Patch*, das duas Universidades Federais Brasileiras do Acre e Paraíba, e o grupo de pesquisa baseado na Universidade Paris 8 na França, representado pela disciplina de graduação *Introduction à la programmation avec Kiwi, Max et Pure Data 1*.

Este estudo pode ser analisado a partir de características da música ubíqua. Em primeiro lugar, assim

como afirmam Luzilei Aliel e José Fornari, essa prática é um tipo particular de "música eletroacústica na qual equipamentos eletrônicos são ubiquamente interconectados para juntos criarem música" (Aliel & Fornari, 2014). Em segundo lugar, como Keller *et al.* afirmam, a "multiplicidade de dispositivos interconectados" substitui a operatividade de "instrumentos musicais identificáveis", assim como a complexa "parafernália social (partituras, concertos, rituais pré-estabelecidos, um público definido e separado dos músicos, etc.), que caracterizam as práticas musicais habituais" (Keller et al., 2010). Finalmente, o conjunto particular de circunstâncias materiais envolvidas nesta prática implica o encontro e a confrontação entre a presença/ausência humana, assim como a caracterização da territorialização nas ações criativas. Isto evoca a formulação de Keller e Lazzarini sobre uma "prática criativa ecologicamente fundamentada" (Keller & Lazzarini, 2017). O termo "ecologicamente" aqui não implica necessariamente o ativismo direto, mas é mais referido ao intento de reproduzir e dar conta de discursos e aspetos de espacialidade e temporalidade nos processos criativos [1].

O software Kiwi nasceu em um contexto social, científico e musical em pleno desenvolvimento. A colaboração e a interação no meio digital são entendidas como conceitos chave no contexto social atual. A computação em nuvem [*cloud computing*] permite explorar o potencial do processamento digital e armazenamento de informação através de redes de comunicação - principalmente a internet.

Foi neste contexto que aplicações como o Google Docs transformaram radicalmente a prática colaborativa, sincrônica e interativa dos editores de texto e de planilhas. Este princípio elimina a necessidade de agrupar geograficamente os autores, permitindo que vários indivíduos e instituições trabalhem juntos, simultânea e interativamente, no mesmo espaço de trabalho, mas em contextos práticos específicos.

O projeto ANR MUSICOLLL [2] (2016-2018), coordenado pelo laboratório francês Musidanse/CICM [3] em parceria com a empresa privada Ohm Force [4], desenvolveu uma pesquisa centrada na música colaborativa e portátil em tempo real. Os seguintes objetivos foram propostos: (1) o desenvolvimento de uma nova ferramenta para essa prática: o software Kiwi [5]; (2) um estudo detalhado de sua manipulação (principalmente para estudantes universitários e de curso de música); (3) uma investigação sobre a inovação causada por esse software no contexto do ensino "tradicional" do software "clássico" de *patching*, como Pure Data e Max; e finalmente (4) a disseminação dessas práticas na comunidade científica e musical. Após a finalização do projeto em 2018, todos esses objetivos foram alcançados com sucesso. O software open source e gratuito está disponível para download. E sua aplicação pedagógica foi atestada em diversos ambientes acadêmicos, como o Conservatorio *di Musica Arrigo Boito di Parma*, e no curso online *La musique au-delà du numérique* organizado por INA-GRM e Université Paris Lumières (Sèdes et al., 2017). No entanto, o potencial criativo do Kiwi foi pouco impulsionado.

A prática do *live patching* foi o ponto de partida para uma nova aplicação deste software. Com o princípio composicional de começar um patch "do zero", em uma sessão colaborativa e simultânea internacional, espera-se que emergja uma série de propriedades originais potenciais desta nova ferramenta.

Este projeto foi iniciado com a associação de dois grupos de pesquisa, que colaboram com as mesmas

ferramentas de criação musical digital: o projeto de pesquisa Live/Acc/Patch, baseado em duas universidades federais brasileiras (Acre e Paraíba), e o grupo de trabalho associado à disciplina de graduação *Introduction à la programmation avec Kiwi, Max et Pure Data 1*.

Esta colaboração une dois países diferentes, França e Brasil, com suas próprias concepções de criação musical, assim como seus próprios *know-how* intrínsecos. O uso do software Kiwi como uma ferramenta compartilhada constituiu o espaço interseccional dessa colaboração. Isto resultou em uma prática que visa articular a criação musical específica para cada ambiente criativo.

2. Live/Acc/Patch

Live/Acc/Patch é um projeto de pesquisa iniciado em 2018 por meio do fomento do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Universidade Federal do Acre (UFAC), como parte do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Os membros do projeto são pesquisadores e alunos entre os 25 e os 35 anos, com experiência nas práticas criativas e na educação musical.

O projeto visa explorar conexões possíveis e viáveis entre a aplicação colaborativa do *live patching* e o aprendizado de programas modulares gráficos para manipulação de áudio, como o Pure Data (Puckette, 1996). Uma primeira parte do projeto envolveu a prática de patches colaborativos em tempo real com usuários já iniciados com Pure Data. Uma segunda parte, em andamento, consiste em acompanhar usuários não iniciados na aplicação de patches, com o objetivo de transferir noções básicas do *live patching*.

2.1. Live coding e live patching

O termo "*live patching*" deriva do conceito de *live coding*, definido por Thor Magnusson como uma "forma de performance musical que envolve a composição musical em tempo real por meio de código textual" (Magnusson, 2011). O *live coding* surgiu como uma prática criativa, assim como uma linha de pesquisa em música nos últimos 15-20 anos. Collins *et al.* situam o *live coding* como uma nova área de performance baseada no risco, na imprevisibilidade e, portanto, com grande potencial de geração de eventos inesperados (2010). Magnusson ilustra a transição da partitura para o algoritmo, introduzido pelo *live coding*, como uma nova forma de criação de significado musical (2011). Apresentando a ferramenta de *live coding* Gibber, Roberts e Kuchera-Morin exploram as relações entre o *live coding* e o navegador de internet (*browser*), entendido como uma "plataforma musical em tempo real" (2012).

Enquanto o *live coding* é um termo geral que evoca implicitamente interfaces textuais, TUIs [*textual user interfaces*], o *live patching* refere-se à mesma prática aplicada a interfaces gráficas GUIs [*graphical user interfaces*] (Harazim, 2017). Estas facilitam o aprendizado, pois as GUIs são geralmente mais intuitivas do que as TUIs. Assim, as GUIs são mais indicadas para a colaboração e para os propósitos educacionais do projeto Live/Acc/Patch. Os projetos *blind date* e *rec.wie.m* da associação Pd~Graz

(Ritsch, 2014) [6] são experiências coletivas de *live patching* anteriores a este projeto que privilegiam também as GUIs. Mesmo com a vantagem das interfaces gráficas, o projeto Live/Acc/Patch demanda a formulação de estratégias que visam facilitar a compreensão das sintaxes modulares e assim acelerar o aprendizado por parte de usuários sem experiência neste tipo de linguagem.

3. De Pure Data a Kiwi

Logo após seu início na Universidade Federal do Acre, o projeto Live/Acc/Patch foi parcialmente transferido para a Universidade Federal da Paraíba, em João Pessoa. Em seguida, foi instituída uma colaboração, fundamentada na aplicação do software Kiwi, com a Universidade Paris 8.

Até então, Pure Data tinha sido o único programa de patches praticado no projeto. Vários desafios foram observados, envolvendo principalmente a impossibilidade (ou a dificuldade) de trabalhar simultaneamente em um mesmo patch. Inicialmente, os participantes trabalhavam colaborativamente em sessões de *live patching*. Esta prática era fundamentada no revezamento consecutivo, em sessões alternadas de trabalho em um mesmo arquivo Pure Data. Mais tarde, quando a Paraíba foi incluída no projeto, os participantes trabalhavam remotamente, cada um salvando patches em redes compartilhadas e intervindo em estruturas já iniciadas por outros autores do mesmo projeto.

A colaboração com Paris 8 introduziu a possibilidade de integrar o software Kiwi. Esta ferramenta solucionou as dificuldades ligadas às distâncias que condicionaram a interação colaborativa na pesquisa. Ao mesmo tempo, permitiu editar os mesmos *patches* simultaneamente, usando computadores diferentes, ou compartilhando o mesmo espaço físico, como será mencionado na seção 5. Os membros do projeto logo decidiram que Kiwi não iria suplantiar Pure Data, e que, pelo contrário, as duas ferramentas poderiam coexistir no projeto.

3.1. Características do Kiwi

Na página oficial do projeto Kiwi, esta ferramenta é definida como "um ambiente gráfico de programação dedicado à música e criação de som, no mesmo modelo de software como Max ou Pure Data, mas oferecendo uma abordagem colaborativa em tempo real: Kiwi permite que vários usuários distantes trabalhem simultaneamente no mesmo patch hospedado online" [7].

Kiwi é, por sua vez, o principal resultado do projeto de pesquisa ANR MUSICOLL, financiado pela Agência Nacional de Pesquisa Francesa (ANR) de 2016 a 2018. Este mesmo projeto estudou a implementação de modelos de interação remota em tempo real.

O modelo de colaboração de Kiwi é fundamentado no compartilhamento de um mesmo espaço de trabalho, ou seja, em um patch online. No entanto, os parâmetros de controle e os sinais de áudio são apenas locais. Dessa forma, com um mesmo patch, cada participante pode controlar individualmente os parâmetros, produzindo resultados sonoros distintos. "Por outro lado, [...] se os usuários desejarem compartilhar dados do seu patch, por exemplo sincronizar objetos GUI, é possível fazê-lo com o objeto

[hub], específico desta linguagem. Este transmite mensagens para todos os usuários conectados ao mesmo patch" [7].

3.2. Notas sobre a GUI do Kiwi

As funcionalidades e a interface de Kiwi são similares às de Pure Data. Os objetos tendem a ter os mesmos nomes, e a interagir de forma muito parecida. Além disso, os atalhos de teclado usados para criar objetos, números e outros itens são os mesmos que em Max ("N" para um novo objeto, "M" para uma mensagem, "l" ou "F" para um número, etc.). As semelhanças entre o Kiwi e o Pure Data ajudaram os membros do Live/Acc/Patch a transitar sem dificuldade de uma aplicação para a outra e vice-versa. O patch abaixo exibe a afinidade entre os dois aplicativos (fig 1).

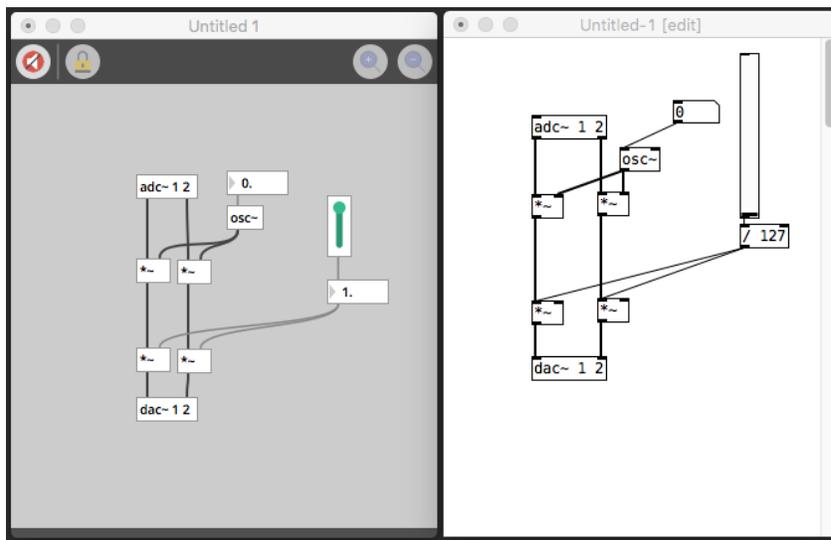


Fig 1. Um exemplo de patch de *ring modulation*. À esquerda com Kiwi, à direita com Pure Data.

Uma característica importante de Kiwi é a presença de um diretório comum onde patches colaborativos podem ser criados e colocados a disposição dos outros usuários. Todos os usuários conectados ao servidor têm acesso a uma lista de patches: eles podem baixar, excluir, carregar renomear, e duplicar patches desse diretório. Destaca-se que em cada patch vários usuários podem estar conectados ao mesmo tempo. Desta maneira, cada usuário é livre para realizar modificações em tempo real em um mesmo patch. As instâncias de colaboração intercontinental discutidas neste artigo aconteceram, cada uma delas, dentro de limites de patches específicos.

4. A classe “Introduction à la programmation avec Kiwi, Max et Pure Data 1” na Universidade Paris 8

O segundo grupo de trabalho, participante desta colaboração internacional, é uma turma da disciplina *Introduction à la programmation avec Kiwi, Max et Pure Data 1* do departamento de música, especialização CAO - *composition assistée par ordinateur* - da Universidade de Paris 8, nível *Licence 2* (equivalente ao

segundo ano de graduação). Os participantes, na maioria com idade entre 20 e 25 anos, são alunos compositores de diversos estilos musicais. Nesta disciplina são ensinadas noções básicas da sintaxe modular gráfica para síntese e tratamento de áudio, ou seja, os princípios básicos do funcionamento de Max, Pure Data e Kiwi. Neste curso são abordadas também noções de processamento digital de sinais e composição musical eletroacústica.

Entre 2017 e 2018, a disciplina *Introduction à la programmation avec Kiwi, Max et Pure Data 1* do curso de música da Universidade de Paris 8 foi o laboratório para o desenvolvimento do software Kiwi. Assim, esta ferramenta foi testada em um contexto predominantemente pedagógico. Certamente, este era o principal alvo de pesquisa do projeto MUSICOLLL, mas ainda é necessário tirar partido das suas virtudes criativas, principalmente a colaboração e o *live patching* colaborativo..

Em 2019, a metodologia pedagógica desta disciplina foi modificada em relação aos anos anteriores. Entre 2017 e 2018, somente o software Kiwi foi utilizado. Mesmo se este princípio foi bem aceito e validado pela comunidade acadêmica, a mudança foi principalmente motivada para que o ensino contemplasse múltiplas ferramentas que compartilham o mesmo funcionamento. Desta maneira, os pontos positivos de cada software foram explorados nos momentos adequados. Por exemplo, a compreensão dos princípios básicos de processamento de sinal é melhor assimilada com o uso de tabelas que permitem a visualização dos valores. Essa funcionalidade é melhor desenvolvida em Pure Data em comparação com Kiwi.

Após terem cursado esta disciplina, os alunos ficam livres para escolher a ferramenta adequada à sua prática, bem como ao seu próprio contexto de trabalho. Assim, o ensino do software Kiwi não substitui os ambientes “clássicos” como Max e Pure Data.

4.1. Aprendizado com Kiwi na Universidade Paris 8

A turma de 2019, que participou da colaboração internacional tratada neste artigo, era formada por aproximadamente dez alunos. A maioria não tinha sido iniciada na composição musical com meios informáticos.

Nas onze primeiras aulas, foram abordadas noções de processamento digital e alguns pontos introdutórios para a composição eletroacústica, como a modulação de frequência, a modulação de amplitude, uso de *delay*, *flanging* (ou pseudo *flanging*) e *sampling*. O software Kiwi foi introduzido somente na sétima aula, quando os alunos já estavam familiarizados com os princípios básicos do *patching*. O objetivo específico da colaboração internacional com Kiwi para essa turma foi proporcionar uma atividade criativa complementar às aulas para auxiliar o aprendizado do conteúdo já abordado em aula (com os programas Pure Data, Max e Kiwi).

Os dois patches criados colaborativamente unicamente por essa turma (antes da colaboração com o grupo Live/Acc/Patch) coincidiram com o momento em que a modulação de frequência foi introduzida. O patch KiwiMaxPd20192 (fig 2) contém quatro instâncias de um modelo básico de síntese FM de John Chowning (1973). A classe foi dividida em quatro grupos, cada grupo se ocupava de um alto-falante. O

objeto [hub], que permite a colaboração de valores de controle, era utilizado para que todos os participantes pudessem enviar *bangs* ativando todas as instâncias do patch.

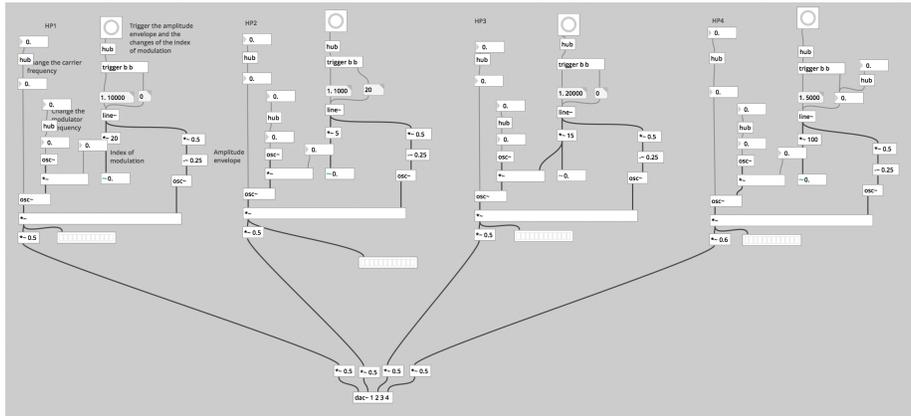


Fig 2. Captura de tela exemplificativa do patch KiwiMaxPd20192, baseado na modulação de frequência, realizado pela turma Introduction à la programmation avec Kiwi, Max et Pure Data 1. Para acessar o patch completo é necessário baixar a ferramenta Kiwi ao endereço <https://kiwi.univ-paris8.fr/>. Após a instalação e a registo/login, o usuário terá acesso remoto à lista de patches compartilhados.

5. Live patching simultâneo intercontinental

No final de 2018, depois de uma série de práticas locais destinadas a se familiarizar com Kiwi, os membros do projeto Live/Acc/Patch iniciaram sessões de *live patching* entre os dois estados brasileiros, Acre e Paraíba. Em abril de 2019, foram realizadas três tentativas de aplicação de *live patching* intercontinentais entre a França e o Brasil (fig 3).

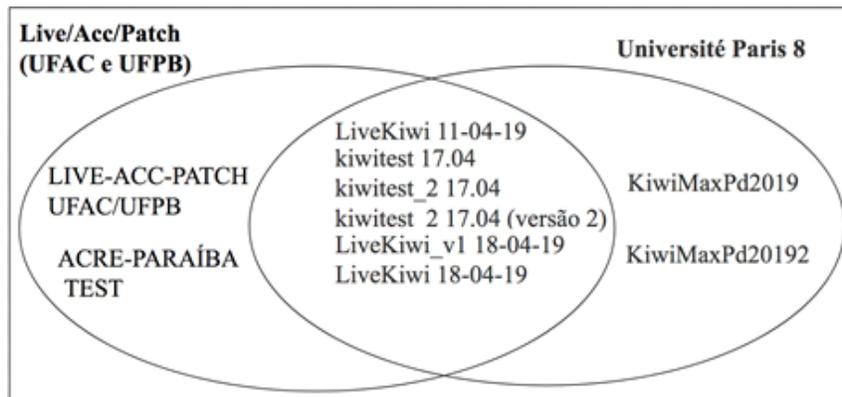


Fig 3. Um diagrama completo dos patches Kiwi desenvolvidos como parte desta pesquisa.

5.1. As primeiras sessões interestaduais no Brasil

Uma primeira rodada de sessões interestaduais ocorreu entre o Acre e a Paraíba nos últimos meses de 2018. As sessões estão disponíveis no servidor Kiwi como LIVE-ACC-PATCH UFAC/UFPB seguidas

por números consecutivos [8]. No total foram criados quatro patches entre 17 de outubro e 28 de novembro.

5.2. Sessões intercontinentais

As sessões intercontinentais relevantes incluem uma sessão de teste em 11 de abril (disponível como LiveKiwi 11-04-2019), três ensaios em grupos menores em 17 de abril (disponível como kiwitest 17.04.19; kiwitest_2 17.04.19; kiwitest 17.04.19 (versão 2)), e um evento final colaborativo em 18 de abril (fig 4), que foi dividido em sessões consecutivas (disponíveis como LiveKiwi 18-04-2019 e LiveKiwi_v2 18-04-2019). É importante ressaltar que, devido a algumas falhas repetidas no servidor durante a sessão de teste em 11 de abril, os participantes baseados no Brasil decidiram começar um patch separado naquele dia (disponível como ACRE-PARAÍBA TEST) que de fato funcionou sem problemas [8]. Graças à ajuda dos desenvolvedores do Kiwi, o problema no servidor não se repetiu nas sessões seguintes.

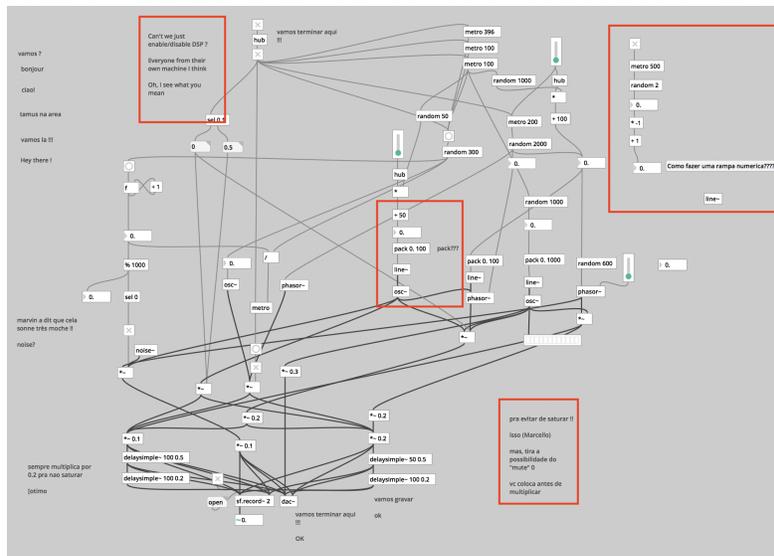


Fig 4. Captura de tela exemplificativa do patch LiveKiwi 18-04-2019. Destacam-se em caixas vermelhas vários exemplos de comunicação verbal entre os participantes no mesmo espaço de trabalho. Para acessar o patch completo é necessário baixar a ferramenta Kiwi ao endereço <https://kiwi.univ-paris8.fr/>. Após a instalação e registro (login), o usuário terá acesso remoto à lista de patches compartilhados.

6. Reflexões sobre Kiwi, *live patching* e interação remota simultânea

As diferentes sessões mencionadas acima foram úteis para formular reflexões críticas e analíticas sobre o conjunto de operacionalidades, funcionalidades e conceitos que estão em questão quando se opta pelo uso de Kiwi.

6.1. Os limites práticos e conceituais da interação remota simultânea

Desde a nossa perspectiva, a inovação mais atraente oferecida pelo Kiwi é, sem dúvida, a interação

coletiva no *live patching*, sendo potencialmente simultânea e remota (Paris *et al.*, 2017) [9]. Embora tenhamos desfrutado abundantemente destas duas características ao longo das várias sessões, os problemas acima mencionados, que ocorreram no primeiro teste intercontinental em 11 de abril, desencadearam uma reflexão mais ampla sobre a conveniência da simultaneidade no contexto da interação remota. Ou seja, é sempre necessário interagir diretamente em tempo real? Ademais, é sempre seguro e razoável confiar na simultaneidade?

Kiwi permite, de fato, que a simultaneidade seja deslocada temporalmente. Por exemplo, os patches que foram criados há vários meses ainda estão disponíveis para todos usuários, antigos e novos, que desejam acessá-los e modificá-los. Assim, Kiwi armazena patches coletivos e nunca os "conclui": eles permanecem abertos na rede para que a comunidade continue modificando-os. Nesse sentido, a princípio, eventos coletivos anteriores envolvendo (ou não) lugares diferentes, também podem ser integrados a atividades atuais. Esse é certamente um desenvolvimento futuro que merece ser levado em consideração nesta pesquisa.

Outra questão conceitual importante é o distanciamento, que pode ser descrito como a simulação de uma presença *in loco*. Este fenômeno se manifesta através de reprodução visual de operações realizadas nas telas dos computadores. Ao considerar sistemas de detecção biométrica, Joseph Pugliese parafraseia Derrida na expressão "metafísica da presença", e identifica uma série de potenciais falhas envolvidas no exercício tecnológico de reconhecer e transmitir os sinais da presença humana (Pugliese, 2014). Na sequência deste trabalho, é útil perguntar quais são os limites da simulação da presença no contexto do *live patching*. Mais especificamente, quais perdas fundamentais relativas à aplicação de patches em tempo real estão implícitas quando os usuários interagem remotamente? E, de uma perspectiva oposta, essa possibilidade de interação torna o uso do Kiwi significativo e desejável apenas se for realizado de forma remota? [10]

Durante as várias sessões remotas mencionadas acima, os membros do projeto ficaram dependentes de comentários textuais nos patches a fim de compensar a ausência de comunicação verbal com os participantes distantes. É importante esclarecer que a interação performática entre os participantes não era imediatamente dependente destes comentários, porém eles preenchiam uma função importantíssima. A performance não precisava imediatamente das caixinhas com texto, mas esses comentários proporcionavam um encanto e um atrativo maior à dialética da interação criativa. A mistura de inglês, português, francês e até italiano se acompanhava com uma mistura de tópicos comunicativos diferentes: perguntas sobre objetos específicos, cumprimentos, brincadeiras, etc. O texto escrito, em suma, preencheu uma lacuna metafísica na interação com outros seres humanos.

Esta necessidade de contacto verbal imediato torna, sem dúvida, a interação local significativa e pertinente, juntamente com as mais remotas. De fato, novamente por ocasião da falha do servidor em 11 de abril, a sessão alternativa aberta por participantes do Brasil (disponível como ACRE-PARAÍBA TEST) consistiu principalmente em uma interação entre dois usuários que estavam trabalhando simultaneamente em duas máquinas diferentes, ocupando o mesmo espaço em Rio Branco, Acre, isto é, o Laboratório de

Arte e Tecnologia Musical (LATMus).

Neste caso, a interação permitiu que os dois participantes tivessem a possibilidade de estabelecer um maior nível de sintonia durante o trabalho colaborativo. Isto ocorreu devido à possibilidade de comunicação verbal e não-verbal imediata. Além de significativa, quando comparada à interação remota, este tipo de interação, próxima e simultânea, preenche uma lacuna importante no software de patching existente. Como mencionado acima, diferentemente de Kiwi, o patching colaborativo em Pure Data, no contexto do projeto *Live/Acc/Patch*, só poderia ser consecutivo, ou seja, uma interação em revezamento entre os usuários.

6.2. As implicações geopolíticas do *live patching* remoto

Como sugerido acima, a interação simultânea pode ser problemática. A simultaneidade exige um elevado desempenho durante um período limitado. Assim, essa situação nem sempre é possível em qualquer lugar e em qualquer momento. A variação geográfica também pode proporcionar mudanças dramáticas na conexão com o servidor, no desempenho da máquina, etc. Diferenças de fuso horário também são relevantes, especialmente se considerada a diferença de 7 horas entre Rio Branco e Paris. João Pessoa, naquela época do ano, estava 2 horas à frente de Rio Branco e 5 horas atrás de Paris. Estas diferenças tiveram uma influência relevante na praticidade das sessões intercontinentais. O fuso horário de Paris (ECT) foi usado como referência para todos os outros. Embora um dos princípios que orientou o trabalho colaborativo fosse a horizontalidade, o projeto correu o risco de se alinhar em coordenadas cartesianas muito específicas de poder e de normatividade (Messina & Souza, 2018).

O conflito geopolítico ou territorial é também uma metáfora poderosa que emerge quando se considera a atividade concorrente de um número relativamente alto de participantes em uma sessão de *live patching* em um único espaço. Neste contexto particular, o *live patching* se torna semelhante à luta por um número limitado de recursos, ou à luta para garantir algum controle territorial. De fato, os participantes têm a liberdade de apagar os objetos ou interromper as conexões criadas por outros, e isso tende a acontecer com muita frequência caso o número de usuários ativos simultaneamente seja alto.

De qualquer modo, os múltiplos e abrangentes significados da palavra "patch" passam a englobar, fora dos seus usos computacionais, a ideia de uma porção limitada de território separada por fronteiras. Suvendrini Perera, por exemplo, considera como ponto de partida a descrição do Pacífico como "*our patch*" pronunciada pelo então Primeiro-Ministro australiano John Howard. Isto ocorreu com a finalidade de considerar e analisar a violenta promulgação de um amplo projeto de dominação sobre a região. Metáforas à parte, a conexão entre a interface visual e a possibilidade de interação simultânea de múltiplos usuários, que caracteriza o Kiwi, permite criar um paralelo entre o *live patching* como atividade musical e a política do desejo territorial que enquadra as ações humanas no espaço. Nesse sentido, o *live patching* colaborativo é muitas vezes a materialização de algum tipo de conflito. Ademais, essa propriedade é nítida nos resultados sonoros. Neste contexto, são relevantes as considerações de Jacques Attali em relação ao

som/ruído como "simulacro" de formas rituais de conflito (1985).

Afinal, o conflito é algo que está intrinsecamente envolvido na prática coletiva da improvisação musical, e ao mesmo tempo, na esfera política, a capacidade de improvisar torna-se uma importante prioridade no contexto decisional e de resolução de conflitos. Se, por um lado, Hoom Hong Ng descreve a improvisação musical coletiva como "esforço sociocomunicativo" (2018), por outro lado, Kathleen M. Eisenhardt argumenta que o modelo social da improvisação, inspirado justamente nas práticas musicais, possa servir como modelo produtivo de tomada de decisões estratégicas (1997).

6.2. *Live patching*, memória e o trabalho em fragmentos

Obviamente, a referência ao conflito é apenas simbólica e não se reflete no clima geral de amizade e camaradagem das sessões intercontinentais. Estas reflexões, no entanto, são úteis para desvendar as implicações sociais e culturais deste tipo de atividade. Nesse sentido, outro "conflito" importante que caracterizou a primeira sessão de testes em 11 de abril foi a divergência entre um número de usuários que começaram a partir de estruturas pré-programadas, e um número de usuários que tentaram construir estruturas gradualmente, objeto por objeto. Logo após a sessão de teste, foi decidido que a colagem de grandes estruturas pré-programadas deveria ser evitada. A atividade em tempo real deveria ser implementada gradualmente. Assim, os usuários interagiram gradualmente .

Nesta prática, foram previstas também a anulação e a correção de ações de outros usuários. Desta maneira, o trabalho envolveu ativar a memória para lembrar de sintaxes, de funcionalidades e até mesmo de nomes de objetos. Nesse contexto, a múltipla dialética entre lembrar e esquecer, entre reter e apagar (Ricoeur, 2004) produz uma série de fragmentos mnemônicos que desempenham um papel crucial na dinâmica do *live patching*. Os participantes confiam em processos de lembranças esboçadas, em vez de consolidarem um corpus constante de conhecimentos específicos de cada linguagem.

Deste modo, a estética do *bem-construído* e do instrumento digital completo é substituída por uma poética do incompleto, do experimental e do aberto. Neste contexto, os fragmentos que sobrevivem ao esquecimento são simultaneamente simulacros de atividades passadas e precursores de novas e imprevistas possibilidades. O *Live patching* é, neste sentido, uma poética do incessante inesperado.

Em termos mais gerais, a complexidade dos patches intercontinentais resulta da combinação de pequenos elementos que interagem dinamicamente se tornando uma estrutura própria. Essa negociação suscitou questões importantes relacionadas à escuta, temporalidade e perenização. Ora, as escolhas composicionais foram guiadas majoritariamente pelo retorno audível local de cada participante, confirmando o conceito de "ação/percepção em loop feedback" proposto por Horacio Vaggione (2001). A temporalidade na criação contemporânea, tema muito discutido pela comunidade científica musical atualmente, ganha uma nova dimensão com o *live patching*. Ademais, a perenidade e efemeridade da computação musical em contexto ubíquo também merece uma reavaliação com a prática colaborativa síncrona. Essas confrontações, de ordem local e global, confirmam os valores didáticos, objetivos comuns do grupo francês e brasileiro. Essa colaboração atesta a vocação científica do fazer musical na essência

do ensino universitário.

7. Conclusão

A pesquisa em curso, relatada neste artigo, abre perspectivas para o futuro da colaboração dos grupos Live/Acc/Patch, UFPB e Universidade Paris 8. Esta pesquisa se enquadra na linha da pesquisa-criação [*practice-led research*], pois os produtos criativos musicais, o desenvolvimento de software, e a reflexão do fazer musical se unem em uma prática musical científica.

Os resultados musicais, as sessões colaborativas e os patches resultantes são dignos do status de criação musical. Ademais, as gravações realizadas por cada participante [11] das sessões testemunham o momento da performance e também constituem um potencial material para uma nova criação musical. Entretanto, o ponto chave para todo esse processo é a confrontação de ecossistemas locais e globais. Os patches LIVE-ACC-PATCH UFAC/UFPB e KiwiMaxPd2019 foram realizados respectivamente pelo grupo Live/Acc/Patch e pela Universidade Paris 8 de forma local. Cada ecossistema criou assim uma maneira própria de *patching*. No primeiro encontro internacional, essas maneiras próprias foram confrontadas resultando na emergência de uma prática colaborativa comum que culminou na última sessão.

O caráter colaborativo, não hierárquico, por vezes é considerado um defeito para desenvolvedores de software. É desejado que os autores de um documento possam “escolher” ou mesmo “autorizar” a colaboração de um outro autor. De fato, este é o princípio colaborativo de ferramentas como Google Docs. Entretanto, com Kiwi, todos os participantes têm os mesmos direitos. Além disto, as operações não deixam traços genéticos. Ou seja, é impossível saber quem criou ou deletou um objeto ou escreveu um comentário. Assim, potenciais barreiras hierárquicas (professor/aluno, grupo X/ grupo Y por exemplo) são evitadas. Kiwi e o *know-how* do *live patching* conduzidos pelo grupo Live/Acc/Patch selam assim um compromisso de criação musical colaborativa com desenvolvimento informático. Ora, as características da própria ferramenta influenciam na maneira de compor. Ao mesmo tempo, o *know-how* composicional pode aportar melhorias para o desenvolvimento deste software.

Nesta perspectiva, esta pesquisa atesta um comportamento recorrente na história da computação musical: o desenvolvimento de novas ferramentas está vinculado a questões de ordem composicional e social ligadas especificamente à criação musical. Assim, os “problemas técnicos”, por exemplo, têm o mesmo status que os critérios criativos composicionais, pois eles também determinam as escolhas criativas realizadas pelos compositores.

Para o futuro desta pesquisa, duas perspectivas merecem destaque. Para compensar as limitações técnicas do software Kiwi (que não oferece alternativas para o desenvolvimento de módulos dinâmicos, como uma API ou mesmo de *subpatches*), é possível integrar a linguagem Faust [12] para o desenvolvimento de novos objetos e de novas bibliotecas. Para isto, é necessária uma nova etapa pedagógica, pois Faust não é equivalente a nenhuma linguagem visual como Pure Data. Trata-se de uma linguagem textual TUI, cuja sintaxe não é intuitiva para iniciantes. Deve-se então investir muito tempo para

sua assimilação. Uma nova perspectiva pedagógico global e local fruto desta colaboração será, portanto, necessária. Do ponto de vista da performance musical e suas implicações morfológicas (Costa, 2016), o patch ao vivo pode ser combinado com práticas de performance experimental baseadas em improvisação, obras abertas e complexidade sonora. O grupo de música experimental Artesanato Furioso, baseado na Paraíba, por exemplo, será um campo de testes para o futuro deste projeto, e uma primeira colaboração remota com eles está prevista para a segunda metade de 2020.

8. Notas

1. cf. a entrevista de Tate Carson a Damián Keller nesse mesmo dossiê.
2. Musicoll homepage, <http://musicoll.mshparisnord.org/>, último acesso 01/05/2019.
3. CICM (Centre de recherche et création musicale) homepage, <http://cicm.mshparisnord.org>, último acesso 01/05/2019..
4. Ohm Force homepage, <https://www.ohmforce.com/HomePage.do>, último acesso 01/05/2019.
5. Kiwi homepage, <http://kiwi.mshparisnord.fr>, último acesso 01/05/2019.
6. Pd~Graz homepage, <https://pd-graz.mur.at/>, último acesso 01/05/2019.
7. Musicoll/Kiwi GitHub page, <http://musicoll.github.io/Kiwi/>, último acesso 01/05/2019.
8. Para acessar os *patches* é necessário primeiramente baixar a ferramenta Kiwi em <https://kiwi.univ-paris8.fr/>. Após a instalação e a regisração/login, o usuário terá acesso remoto à lista de patches compartilhados.
9. No entanto, um proxy da comunidade Pure Data, chamado *peer data*, foi desenvolvido por IOhannes m zmölnig, e permite que vários usuários compartilhem simultaneamente o mesmo patch. Esta versão foi utilizada com sucesso no projeto mencionado acima *blind date* do Pd~Graz. Outras abordagens existem para colaboração remota em Pure Data, como o projeto Destino Pirlampo, de Luzilei Aliel da Silva e José Eduardo Fornari Novo Junior (2014). Neste exemplo, os usuários remotos enviam sinais para um patch central unificado, mas sem controle direto no patch principal.
10. Sobre criatividade musical e metafísica da presença, cf. também Messina e Aliel (2019) e Messina et al. (2019),
11. As gravações estão disponíveis em: https://mega.nz/#F!b8ZniKAK!QgoFc8bjKu9UtkcibsX_-Q.
12. Faust é uma linguagem funcional dedica à síntese e processamento de áudio digital em tempo real. Ademais, ela oferece um conjunto de ferramentas para geração de código C++ e um compilador em tempo real que permite executar o código Faust dentro de aplicativos terceiros. A versão mais recente de Kiwi integra esta funcionalidade. Faust homepage, <https://faust.grame.fr>, último acesso 01/05/2019.

9. Referências

- Aliel da Silva, L., & Fornari Novo Junior, J. E. (2014). Projeto Destino Pirlampo: um Estudo sobre a Composição de Meta-Soundscapes em Música Ubíqua. *Revista Música Hodie*, 14(1), 105-121.
- Attali, J. (1985). *Noise: The political economy of music* Minneapolis: The University of Minnesota Press.
- Chowning, J. M. (1973). The synthesis of complex audio spectra by means of frequency modulation.

- Journal of the audio engineering society*, 21(7), 526-534.
- Collins, N., McLean, A., Rohrhuber, J., & Ward, A. (2003). Live coding in laptop performance. *Organised sound*, 8(3), 321-330.
- Costa, V. F. d. (2016). *Morfologia da Obra Aberta: esboço de uma teoria geral da forma musical*. Curitiba: Editora Prismas.
- Eisenhardt K.M. (1997) Strategic Decision Making as Improvisation. In: Papadakis V., Barwise P. (eds) *Strategic Decisions*. Boston: Springer.
- Galleron, P., Maestri, E., Millot, J., Bonardi, A., & Paris, E. (2018). Enseigner le patching de manière collective avec le logiciel collaboratif Kiwi. In *Actes des Journées d'informatique Musicale (JIM)*, pp. 106-114.
- Harazim, M. (2017) Music, Computers and Culture in Live Coding. BMus dissertation. University of Aberdeen.
- Keller, D., Barreiro, D. L., Queiroz, M., & Pimenta, M. S. (2010). Anchoring In Ubiquitous Musical Activities. In *Proceedings of the International Computer Music Conference (ICMC)*. Ann Arbor: University of Michigan Library, 319-326.
- Keller, D., & Lazzarini, V. (2017). Ecologically grounded creative practices in ubiquitous music. *Organised Sound*, 22(1), 61-72.
- Magnusson, T. (2011). Algorithms as scores: Coding live music. *Leonardo Music Journal*, (21), 19-23.
- Messina, M., & Aliel, L. Ubiquitous Music, Gelassenheit and the Metaphysics of Presence: Hijacking the Live Score Piece Ntrallazu 4. In *14th International Symposium on Computer Music Multidisciplinary Research* (p. 685-695).
- Messina, M., & Souza, J. d. A. (2019). Rios, pontes, balsas e fronteiras: uma provocação desde a brasilidade liminar e precária do Vale do Rio Acre. *Muiraquitã - Revista de letras e humanidades*, 6(1), 80-93.
- Messina, M., Svidzinski, J., de Menezes Bezerra, D., & da Costa, D. F. (2019). Live Patching and Remote Interaction: A Practice-Based, Intercontinental Approach to Kiwi. In *14th International Symposium on Computer Music Multidisciplinary Research* (p. 696-703).
- Ng, H. H. (2019). Collective Free Music Improvisation as a Sociocommunicative Endeavor: A Literature Review. *Update: Applications of Research in Music Education*, 37(2), 15–23.
<https://doi.org/10.1177/8755123318784109>
- Paris, E., Millot, J., Guillot, P., Bonardi, A., & Sèdes, A. (2017, May). Kiwi: vers un environnement de création musicale temps réel collaboratif premiers livrables du projet Musicoll. In *Journées d'Informatique Musicale*.
- Paris, E. (2018). *Une approche du patching audio collaboratif*. Doctoral dissertation. Université Paris 8.
- Perera, S. (2007). Our patch: Domains of whiteness, geographies of lack and Australia's racial horizons in the war on terror. *Our Patch: Enacting Australian Sovereignty Post-2001*, 119-46.
- Puckette, M. (1996). Pure Data: another integrated computer music environment. *Proceedings of the second intercollege computer music concerts*, 37-41.
- Pugliese, J. (2014). The alleged liveness of "Live": Legal visibility, biometric liveness testing and the metaphysics of presence. In Wagner, A., Sherwin, R. K. (eds.) *Law, culture and visual studies* (pp. 649-669). Springer, Dordrecht.
- Ricoeur, P. (2004). *Memory, history, forgetting*. Chicago: University of Chicago Press.
- Ritsch, W. (2014). ICE-towards distributed networked computer music ensemble. In Georgaki, A.,

Kouroupetroglou, G. (eds.) *Proceedings ICMC|SMC|2014*.

Roberts, C., & Kuchera-Morin, J. (2012). Gibber: Live coding audio in the browser. In *ICMC*.

Sèdes A., Bonardi A., Paris E., Millot J. et Guillot P., (2017) "Teaching, investigating, creating: MUSICOLL", in *Innovative Tools and Methods for Teaching Music and Signal Processing*. <hal-01581698>

Vaggione, H. (2001). Some ontological remarks about music composition processes. *Computer Music Journal*, 25(1), 54-61.