

A Performance Interativa em VAGA

Fernando Iazzetta

Departamento de Música - Universidade de São Paulo
Programa de Comunicação e Semiótica - Pontifícia Universidade Católica de São Paulo
São Paulo - SP - Brasil

iazzetta@usp.br

***Abstract.** This paper describes the composition of VAGA, an interactive piece created in a system based on the MAX/MSP environment for Macintosh computers. We describe the musical material used in the composition as well as the software modules developed to control this material in real time. The system is composed by a computer and midi controllers that are taken as a musical instrument. We describe the interactive processes used in the composition, analyze the system's potentialities and point out to some problems to be addressed in further developments.*

***Resumo.** Nesse artigo descrevemos a composição VAGA criada a partir de um sistema interativo baseado no ambiente MAX/MSP para Macintosh. São descritos os materiais utilizados na peça, bem com os módulos programados para a realização e controle dos parâmetros sonoros. O sistema, composto por um computador portátil e interfaces, é tomado como um instrumento musical e são descritos os seus modos de interação. Finalmente comentamos sobre o potencial do sistema e avaliamos os problemas que ainda necessitam ser sanados.*

1. Introdução

Interação musical é um termo que vem ganhando espaço crescente no contexto da música eletroacústica, especialmente a partir do início dos anos 90. O termo tem sido usado para designar obras que incorporam processos de performance envolvendo instrumentos e aparelhos eletrônicos usados em tempo-real, em contraposição à chamada música sobre suporte. Enquanto na música interativa os processos de controle e geração sonora ocorrem no momento da execução musical, na música sobre suporte, esses processos são realizados anteriormente no ambiente do estúdio, para posterior reprodução do material gerado.

Embora essas duas categorias coexistam na produção atual, ambas apontam para procedimentos diferentes no que diz respeito à composição e performance. A princípio, é de se esperar que questões composicionais sejam mais evidentes que as questões interpretativas na música sobre suporte, enquanto que na música interativa ocorreria o oposto. Obviamente isso é mais uma tendência do que regra. Hoje em dia as ferramentas eletrônicas de produção musical encontram-se num estágio de desenvolvimento em que essas separações entre práticas de estúdio e práticas de performance vão se diluindo.

Nosso interesse pela questão da interação surgiu por volta de 1992, nos primeiros contatos com o ambiente de programação musical MAX, no Laboratório de

Linguagens Sonoras da PUC-SP. Na época foi formado um pequeno grupo de estudo onde foram realizadas as primeiras experiências com o *software*.

MAX (Pukette & Zicarelli, 1990) é um ambiente gráfico de programação musical no qual "objetos" que desempenham funções específicas são arranjados e conectados entre si para realizar funções lógicas em geral, bem como o gerenciamento, em tempo real, de eventos musicais. O nome do programa é uma homenagem a Max V. Mathews e foi criado no IRCAM no final dos anos 80 por Miller Puckette para a estação de trabalho NeXT. Posteriormente David Zicarelli preparou uma versão para Macintosh, atualmente comercializada por sua empresa Cycling'74.

MAX é baseado no conceito de "programação orientada por objetos" (Pope, 1991), embora não satisfaça plenamente esse conceito. Um objeto pode ser entendido como um módulo autônomo do programa que pode realizar sozinho uma determinada função, sem ser afetado pelo modo funcionamento de outros objetos. Objetos comunicam-se formalmente entre si, permitindo a passagem de dados. Quando uma tarefa é executada duas vezes em um mesmo programa, pode-se usar duas "instâncias" do mesmo objeto, sem a necessidade de realmente duplicá-lo. O ambiente MAX contém uma biblioteca de objetos com funções pré-definidas capazes de executar diferentes tipos de tarefas como processamento matemático, receber e enviar múltiplas mensagens MIDI, gerenciar eventos temporais, ler e armazenar diversos tipos de dados, criar e gerenciar tabelas e gráficos, controlar e mostrar imagens e vídeo digitais (*QuickTime*), mostrar números e textos.

MAX oferece um ambiente de programação muito mais claro e objetivo que as linguagens de programação tradicionais, baseadas em linhas de comando textual.

Até pouco tempo, uma das limitações desse ambiente era a de que o mesmo havia sido projetado basicamente visando a manipulação de dados MIDI. Embora bastante flexível nesse aspecto, a dificuldade de se controlar eventos de áudio limitava o escopo de atuação do ambiente MAX e direcionava sua utilização como uma espécie de controlador de múltiplos instrumentos capazes de "compreender" informação MIDI. A maior parte dos sistemas interativos que se utilizavam desse ambiente de programação consistiam de um *setup* de módulos de som, sintetizadores e *samplers* conectados a uma interface MIDI e controlados por um computador Macintosh rodando aplicativos gerados em MAX.

Esse panorama se modifica radicalmente por volta de 1997 quando David Zicarelli lançou o pacote MSP (Wells, 1998), espécie de camada complementar a ser adicionada ao ambiente MAX original e cuja função seria o tratamento de áudio digital e operações de DSP em tempo-real, dentro do próprio programa. O MSP configura-se como uma coleção de objetos semelhantes aos objetos tradicionais do ambiente MAX, mas que ao invés de trabalharem com dados numéricos e eventos MIDI, destinam-se à geração e processamento de sinais de áudio digital.

O surgimento do MSP está obviamente ligado ao poder de processamento alcançado pelos computadores atuais e tem modificado a forma como o computador passou a atuar nos processos de composição e performance musical. Primeiro porque possibilitou a coexistência, num mesmo ambiente, de ferramentas tradicionalmente ligadas ao trabalho dentro do estúdio (processos de síntese, manipulações espectrais,

processos formais de composição) e ao ambiente de performance (controle de parâmetros sonoros e composicionais em tempo-real), tornando o computador uma espécie de instrumento genérico de produção e performance musical. Segundo porque trouxe para o computador o domínio do material sonoro a ser gerado, o que anteriormente era relegado a uma série de equipamentos periféricos (sintetizadores, *samplers*, processadores de efeito, etc).

2. A Composição de VAGA

Muito já foi dito sobre a questão do estúdio como ambiente composicional e as diferenças que esse ambiente traz para a música. A idéia de canalizar diversos estágios do fazer musical num mesmo ambiente como o computador subverte, de certo modo, algumas divisões clássicas da atividade musical. Num ambiente como MAX/MSP não pode haver distinção clara entre composição e performance ou entre instrumento e partitura. Todos esses elementos estão conectados ao *software* e ao *hardware* que formam o sistema. A composição de VAGA leva em conta esses fatores e tenta explorar as possibilidades, desafios e contradições geradas por um processo em que a composição não se separa da performance e o instrumento funciona também como uma partitura.

Esse trabalho nasceu de experiências realizadas em parceria com o compositor Silvio Ferraz na realização de diversas performances eletroacústicas em que eram usados elementos de interação em tempo-real. Basicamente cada um de nós criava um "instrumento" musical em seu computador usando o ambiente MAX/MSP. Então nos juntávamos para explorar as potencialidades desses instrumentos em sessões de improvisação. A partir desse trabalho eram selecionados os materiais mais interessantes para compor uma peça. Para isso era elaborada uma partitura que funcionava muito mais como uma espécie de roteiro para a organização dos materiais sonoros e gestos musicais que iriam compor a peça.

É interessante notar que esse modo de compor levava a um processo contínuo de aperfeiçoamento da composição e da performance. A cada nova apresentação, embora baseada na mesma "partitura" alcançávamos resultados diferentes, devido, especialmente, a dois fatores. Em primeiro lugar, havia sempre uma melhora em nossa habilidade de controlar e explorar os instrumentos que havíamos criado. Em segundo lugar, esses instrumentos eram constantemente revisados de modo a corrigir eventuais defeitos ou para incorporar novos processos de geração e manipulação sonora.

A concepção de VAGA segue esses mesmos princípios embora tenha sido concebida para apresentação solo. Toda a peça é gerada em tempo-real a partir de um computador Macintosh rodando diversos aplicativos realizados no ambiente MAX/MSP.

3. O Material Composicional

O material sonoro provém basicamente de três fontes:

- **Arquivos de sons pré-gravados:** são fragmentos de fontes diversas armazenados na forma de uma biblioteca de sons. Parte desses arquivos (como sons de vidro e sinos processados) são utilizados como "pedais" durante a peça. São arquivos sonoros relativamente longos (geralmente executados em *looping*) de sons ricos, porém sem

muita variação dinâmica. Esses arquivos desempenham duas funções principais dentro da composição. A primeira é a de servir de base para a conexão entre outros eventos sonoros; a segunda é a de aliviar a quantidade de parâmetros que o intérprete precisa controlar, já que, uma vez disparados, esses sons não necessitam de intervenções periódicas. Basicamente eles podem ser ligados e desligados, colocados em *looping*, e sofrer alteração de volume. Um pequeno *patcher* em MAX controla a execução de até 6 desses arquivos simultaneamente. Outros arquivos pré-gravados foram extraídos CDs de música popular. Fragmentos musicais são armazenados em *buffers* a partir dos quais é construído um *patcher* que funciona como um *sampler* oferecendo opções como ponto de *looping* e transposição que podem ser modificados em tempo-real. Nesta peça, diversas configurações sonoras são armazenadas e associadas às teclas alfa-numéricas do teclado do computador que serve como interface durante a performance.

- **Sons sintetizados:** embora quase todos os sons gerados em VAGA tenham sido processados de alguma maneira, são poucos os sons gerados por meio de síntese tradicional. Basicamente são usados processos de síntese de FM que geram notas longas e graves no início e final da peça.
- **Sons captados e processados em tempo-real:** constituem o material mais importante de VAGA. Basicamente são usados sons vocais captados por um microfone externo e sons percussivos proveniente de um pequeno tamborim equipado com um captador de contato, os quais são conectados à entrada de áudio do computador.

Os sons dessas duas fontes jamais são ouvidos em seu estado original durante a peça. Eles são sempre processados por 3 módulos diferentes. O primeiro deles consiste na verdade de 3 sub-módulos que realizam as seguintes operações: a) modulação em anel: opera a multiplicação do som original por até dois sons senoidais diferentes posteriormente distribuídos para dois canais estéreo; b) a saída desse processamento está ligada a um segundo sub-módulo que realiza delays variáveis de 5 a 2000 milissegundos independentemente para cada canal estéreo; e c) o terceiro sub-módulo realiza transposição do sinal original (até +/- 2 oitavas) permitindo ainda o controle de outros parâmetros como tamanho do *buffer*, *delay* e *feedback*.

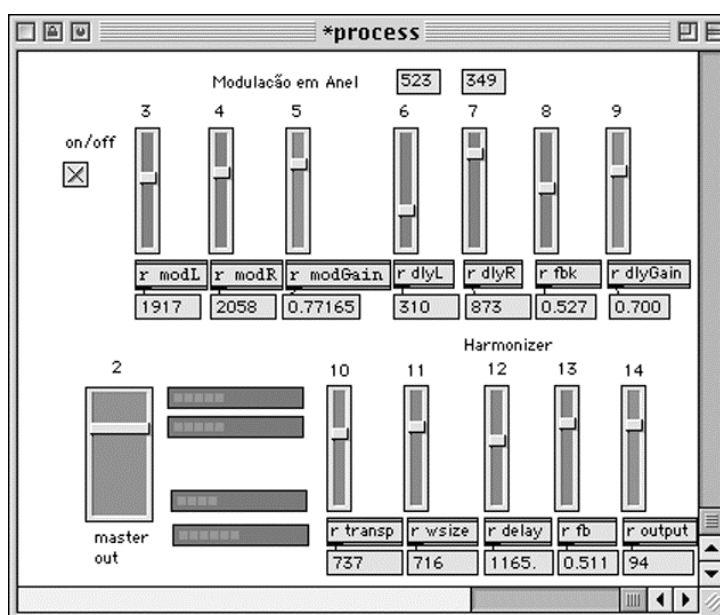


Figura 1: Interface do primeiro módulo de processamento que realiza processamentos de modulação em anel, delays variáveis e transposições.

O segundo módulo realiza processos de síntese granular: esse módulo é uma variação do aplicativo realizado por Richard Dudas em 1998 e oferece uma interface de controle para vários parâmetros como transposição, imagem estereofônica, tamanho e densidade dos grãos. Os sons a serem granulados são captados em tempo-real durante a performance e armazenados em um buffer de tamanho variável (neste caso estamos usando um *buffer* de 3 segundos). A flexibilidade desse módulo consiste em permitir que o conteúdo do *buffer* seja substituído a qualquer momento sem interrupção do processamento. Para isso, basta clicar em um botão para que o aplicativo entre imediatamente em modo de gravação.

O terceiro módulo consiste de 3 *buffers* de 2 segundos nos quais pode-se gravar sons diferentes a qualquer momento. Quando ativado, esse módulo seleciona pequenos fragmentos do material gravado nos *buffers* e realiza pequenos ciclos de *looping*. Isso cria uma espécie de movimento ondulatório. Os fragmentos são selecionados aleatoriamente a cada ciclo o que garante um resultado continuamente variável. A saída desse módulo está ligada a uma unidade de *delay* que pode ser controlada pelo intérprete durante a performance.

4. A Estrutura de VAGA

A estrutura da peça é bastante simples e sua dinâmica pode ser comparada à forma de um arco em que os materiais vão sendo apresentados sucessivamente e sendo superpostos num crescendo que culmina praticamente no segundo terço da peça. Nesse momento arquivos contendo fragmentos de música popular são usados como interferência junto aos materiais apresentados até então. Esses fragmentos se chocam nitidamente com os outros materiais sonoros da peça pela sua referência direta à música popular (basicamente jazz e rock) e por serem apresentados praticamente sem nenhum processamento. Essa se constitui a parte mais densa da peça e serve como transição para o trecho final que dura pouco mais de um minuto. Nesse trecho, de modo inverso ao que

foi apresentado no início da peça, os elementos vão sendo diluídos até que uma nota-pedal greve marca o fim da peça.

5. Interfaces e Performance

Em termos de implementação computacional, o grande desenvolvimento de VAGA não se refere nem aos materiais sonoros, nem à estrutura da peça, mas sim à criação do ambiente de performance. Nossa idéia inicial foi a de transformar o computador num sistema suficientemente rico e complexo para ser tomado como um instrumento musical. De fato, os vários módulos implementados oferecem uma larga gama de possibilidades de geração e controle sonoro. Para a performance de VAGA usamos um computador portátil Macintosh G3 rodando o ambiente MAX/MSP. Nesse ambiente foram criados todos os aplicativos que oferecem as operações e controles necessários para a realização da peça.

Entretanto, ao ser tomado como instrumento musical, esse sistema se torna problemático em relação à interface de controle. Obviamente o teclado alfa-numérico e o *track-pad* do computador portátil nem de longe se mostram adequados para o controle em tempo-real das dezenas de parâmetros disponíveis no sistema. A interface de cada um dos módulos é composta de botões, chaves e *sliders* que só com muita dificuldade podem ser acessados com o *mouse*. A saída encontrada foi a incorporação de interfaces externas para o controle do sistema. A opção foi a utilização do controlador Peavey PC-1600. Esse controlador MIDI oferece 16 *sliders*, 16 botões tipo liga/desliga, um botão giratório e a opção de uso de até 2 pedais. Todos esses dispositivos podem ser facilmente programados para enviar qualquer tipo de mensagem MIDI, oferecendo ainda a opção de gravação de diversas configurações (*presets*) que podem ser acessadas instantaneamente. Com isso o controle dos parâmetros oferecidos pelos módulos do programa ficou extremamente enriquecido e o sistema como um todo ainda se manteve portátil.

Em VAGA utilizamos duas configurações diferentes, dobrando a capacidade oferecida pela interface, ou seja, permitindo o controle de até 32 parâmetros por meio dos *sliders* e o uso dos botões com 32 funções diferentes. Somando-se a isso o *input* do microfone conectado diretamente ao computador temos um instrumento realmente bastante rico e flexível.

6. Conclusões

Como qualquer sistema, esse usado na peça VAGA contém limitações. A mais evidente parece ser a dificuldade no controle do grande número de parâmetros disponíveis. Embora a interface Peavey PC-1600 seja bastante flexível, foi necessário optar um número reduzido de parâmetros que seriam programados para controle em tempo-real, mantendo outros fixos ou dependentes de processos pré-programados. Ainda estamos trabalhando nisso e a interface do sistema vem sofrendo constante atualização. Outra limitação está ligada à nossa intenção de manter o sistema extremamente portátil. O uso do computador *laptop* traz mobilidade, mas limita a saída do sistema à dois canais de áudio. Num futuro bem próximo espera-se que estarão disponíveis interfaces de áudio de até 8 canais para computadores portáteis, o que resolveria esse problema.

Finalmente, temos feito uso constante de outros elementos durante a performance, especialmente com a participação de bailarinos em cena e o uso de imagens projetadas. Para VAGA pode ser utilizado opcionalmente um acompanhamento de imagens geradas por computador e projetadas durante a performance. O programa utilizado nesse caso é o PixelToy, um sintetizador de imagens que oferece algumas opções de programação e pode interagir com o sinal de áudio em tempo-real. Embora a utilização da imagem demande a existência de um segundo computador, o efeito sincronizado com o áudio amplia significativamente o efeito da peça como um todo.

Referências Bibliográficas

Pope, S. T. (1991). *The Well-Tempered Object: Music Applications of Object-Oriented Software Technology*. Cambridge, Mass.: The MIT Press.

Pukette, M. & Zicarelli, D. (1990). *MAX - An Interactive Graphic Programming Environment*. Menlo Park, California: Opcode Systems

Wells, Thomas (1998). *Cycling'74: A breakthrough in real-time synthesis and MIDI-controlled DSP*. *Electronic Musician*, October 1998, pp. 176-182.

Agradecimentos

** Esta pesquisa foi parcialmente financiada pela FAPESP.