Silvia Laurentiz

# Sistemas autônomos, processos de interação e ações criativas

palavras-chaves:
lógica; arte; linguagem; sistemas; autonomia.

A partir de Hugh Dubblerly, Usman Haque e Paul Pangaro em **What is interaction: Are there different types?**, como a lógica de diferentes sistemas – lineares, que se autorregulam, e que aprendem – pode modificar o pensamento do artista? Abordaremos este assunto colocando em foco o envolvimento da linguagem de programação na obra de arte e os modos por meio dos quais a relação entre autônomo e autômato pode estimular um novo campo de possibilidades.

keywords:
logic; art; language; systems; autonomy.

According to Hugh Dubblerly, Usman Haque and Paul Pangaro in **What is interaction: are there different types?**, how the logic of different systems – linear, self-regulatory, and learning – can change the thinking of the artist? We will deal with this issue, focusing on the involvement of the programming language in the artwork and asking how the relationship between autonomous and automaton can stimulate a new field of possibilities.

Artigo recebido em 31 de março de 2011 e aprovado em 16 de maio de 2011



Caio Reisewitz, *paretinga*, 2010.

## Introdução

Há algum tempo, temas recorrentes em eventos de "arte e tecnologia" (subentendido arte digital/cibernética/computacional) têm tratado de questões das chamadas 'outras áreas', como Teoria dos autômatos, Cibernética, Sistemas Complexos, Autonomia, Emergência, áreas que, por sua vez, cada vez mais reaproximam artes e ciências. É comum também ouvirmos falar da relação entre a linguagem de máquina e o artista programador. Dessa forma, cria-se uma intrínseca relação entre os processos criativos e a lógica de programação. Mas dizer apenas que os algoritmos trazem possibilidades expressivas importantes como recurso de linguagem não resolve a questão. Encerrar a obra do artista nos procedimentos tecnológicos apoiados na influência dos modos de produção da arte digital é mais um problema do que solução. Entretanto, se quisermos nos aproximar dessas obras, teremos que entender melhor alguns desses procedimentos, um dos objetivos deste artigo.

## 1. Circularidade, realimentação e controle.

A interação humano-computador sempre esteve baseada em estruturas com retornos contínuos ou *loopings,* que são princípios de circularidade. Assim, a informação flui de um sistema (que pode ser um computador ou até mesmo um carro) para outro (que pode ser uma pessoa, por exemplo), retornando ao primeiro novamente. Nessa passagem, o sistema-pessoa é movido por um objetivo e age na tentativa de alcançá-lo, fornece informações para o outro sistema. Este reage a partir das informações recebidas. Depois, o sistema-pessoa mede o efeito de sua ação, interpretando a resposta do outro sistema, e, consequentemente, compara o resultado desejado ao resultado obtido. A comparação – que apontará a diferença ou correspondência entre o rendimento esperado e o alcançado – dirigirá sua próxima ação, recomeçando o ciclo. Assim funciona um simples sistema de autocorreção, ou, mais tecnicamente falando, um sistema cibernético de primeira ordem. Já apresentamos, em outro trabalho, o que seriam sistemas de primeira e de segunda ordem para a cibernética[1].

Não nos prolongaremos muito sobre isto, mas cabe destacar que já contamos com um percurso histórico, dentro do qual destacamos o primeiro período da cibernética (chamado primeira cibernética), que se ocupava, em geral, dos processos pelos quais os sistemas funcionavam para manter a sua organização. O sistema opera de acordo com um propósito ou meta, cujo alcance é garantido por procedimentos de regula-

1. Ver: LAURENTIZ, Silvia. Uma aproximação da cibernética pela poesia digital. In: **ARS.** São Paulo, vol.4, n.8, p. 114-127, 2006, ISSN 1678-5320. doi: 10.1590/S1678-53202006000200011.

ção e controle[2]. Ou seja, mecanismos que controlam de alguma forma os distúrbios que venham a ocorrer, para manter o sistema estável. Naquele período, o sistema cibernético era compreendido como equivalente a uma máquina trivial, que, tendo uma organização e um propósito, operava através da correção de desvios, de modo que se mantiinha estável. Esse tipo de processo de manutenção era chamado de "realimentação negativa". É importante destacar o papel da realimentação no processo, onde as informações coletadas eram confrontadas com o padrão de desempenho programado. Dessa forma, a diferença entre o desempenho realizado e o esperado era transformada na informação que o mecanismo de compensação utilizava para trazer o desempenho futuro a valores mais próximos do padrão esperado, valores também chamados de padrões de referências. Frisamos que ações futuras dependem das ações presentes, comparadas com os padrões de metas. Para Wiener:

> Uma ação complexa é aquela em que os dados introduzidos (a que chamamos de entrada) para obter um efeito sobre o mundo exterior – efeito a que chamamos de saída – podem implicar um grande número de combinações. Combinações dos dados introduzidos no momento com os registros obtidos de dados anteriores armazenados, a que chamamos memória, e que estão registrados na máquina[3].

Isto já diferencia de alguns procedimentos lógicos em que as ações futuras são processadas por um algoritmo independente dos valores atingidos pelas ações do presente e de alguma meta estipulada anteriormente[4]. Como um exemplo muito simples, apenas para efeito didático, pensemos na seguinte situação:

1. Sorteie-se um número de 1 a 10. Ao resultado, qualquer que seja, some-se + 10;
2. Agora, crie-se outra situação: sorteie-se um número de 1 a 10. Compare-se este número ao o valor de referência, que pode ser 5, por exemplo. Se o valor sorteado for igual ao de referência, permaneça-se com ele. Caso contrário, a diferença entre o número de referência e o sorteado deve ser acrescida ou decrescida (conforme o caso) para que seja atingido o valor desejado.

Dentre os dois casos, o segundo parece ser redundante e sem sentido, pois, seja qual for o número sorteado, sempre obteremos o resultado "5", que é o valor de referência; enquanto que, no primeiro caso, sempre obteremos um número novo, o que parece gratificante e compensador. Mas, se pensarmos com mais vagar, perceberemos que cada um dos casos tem seu

2. WIENER, Norbert. **Cibernética e sociedade: o uso humano de seres humanos.** São Paulo: Cultrix, 1978.

3. Ibidem, p. 23.

4. Por exemplo, podemos citar os processos estocásticos, que são modelos que evoluem no tempo de maneira probabilística. Uma cadeia de Markov é um tipo de processo estocástico, que trabalha com estados discretos, ou seja, sobre um conjunto enumerável ou finito. A probabilidade condicional de qualquer evento futuro ocorrer, dados qualquer evento passado e o estado presente, será independente do evento passado e dependente somente do estado presente. Em outras palavras: um processo estocástico é dito ser um processo *markoviano* se o estado futuro depende apenas do estado presente e não dos estados passados. Esse tipo de processo estocástico é também denominado de *'memoryless process'* (processo sem memória), uma vez que o passado é 'esquecido' (desprezado). Muitos trabalhos de poesia permutacional utilizaram processos estocásticos deste tipo.

papel de importância, que pode, no entanto, passar despercebido em um exemplo tão simples. Vamos, por isso, modificar um pouco nosso exemplo:

1. Sorteie-se um número de 1 a 10. O resultado será a posição de um objeto que aparecerá na tela de um programa, no computador;
2. Sorteie-se um número de 1 a 10. Compare-se o resultado com a posição de outros objetos naquela mesma tela do programa, no computador. Se o número corresponder a uma posição de objetos que já existem na tela, some-se alguns valores, até que o objeto atual não se sobreponha a nenhum outro. Caso contrário, o resultado já corresponderá à posição do objeto atual na tela.

Agora, com o problema assim modificado, o valor de referência passa a ser "a posição dos outros objetos na tela". E, caso aconteça de algum objeto ocupar a mesma posição de outro, o programa "corrige" a situação, para que todos os objetos se ajustem e compartilhem a mesma tela, sem que um se sobreponha aos outros.

Estes processos de correção servem apenas para garantir a estabilidade do sistema. Um exemplo típico fora do computador é o modo de funcionamento de um elevador. Não basta que o sistema apenas seja capaz de abrir a porta externa do elevador; é preciso também que o elevador esteja de fato diante da porta no momento exato em que ela se abra. Ou seja: é fundamental que o desengate para a abertura da porta dependa do elevador estar exatamente diante dela, de maneira precisa, nem um pouco acima, nem um pouco abaixo; afinal, no pior dos casos, estando o desengate da porta acionado sem que o elevador esteja de fato ali, o passageiro pode despencar no poço vazio.

Tal controle da máquina com base no seu desempenho efetivo, e não em seu desempenho esperado, é o que chamamos de realimentação (*feedback*). À correção de ajuste que posiciona o elevador no exato lugar que deve estar, só assim permitindo que se abra a porta, chamamos de realimentação negativa. "*A função desses mecanismos é a de controlar a tendência mecânica para a desorganização*"[5].

5. WIENER. Op. cit., p. 24.

Podemos pensar o sistema do elevador da seguinte forma:

```
Quando o elevador parar {
    Tome a posição do elevador;
        Se posição efetiva do elevador for = posição esperada {
            Abra a porta;
            Fim da rotina}
        Senão {
            Corrigir posição atual para posição de referência;
        parar}
}
```

Portanto, realimentação, ou *feedback,* refere-se a qualquer processo por meio do qual uma ação é controlada pelo conhecimento do efeito de suas respostas, comparando-se o efeito esperado com o efetivo. "Controlar", nesse sentido, é fazer com que uma variável do sistema assuma um valor desejado ou, ainda, um valor de referência, tudo isso através de uma ação do sistema. Introduz-se, assim, a ideia de circularidade, onde um fluxo de informação retorna à sua origem, num processo de retorno causal no qual uma saída de dados do sistema produz efeitos que retornam à sua entrada, possivelmente envolvendo outros sistemas também em *looping*.

Uma questão importante é perceber como os efeitos das ações (ou saídas de informação) retornam ao sistema (isto é, retornam às entradas de informação) para influenciar seus estados e ações subsequentes. Já vimos que na realimentação negativa, por exemplo, privilegiam-se os processos de correção dos desvios dos sistemas, obtendo-se uma redução da entropia (ou tendência à desorganização). Mas, cabe ressaltar, há outras ações possíveis.

## 2. Reação *versus* interação

Vários autores já apontaram as diferenças entre sistemas interativos e reativos[6] . Para Dubblerly, Haque e Pangaro[7], em um sistema reativo – aquele que liga o *input* (entrada de dados) ao *output* (saída de dados) –, a transferência de função é fixa; já em um sistema interativo, a transferência de função é dinâmica. Isto é: na interação, a maneira que o *input* afeta o *output* pode se transformar. Há, inclusive, em algumas categorias de interação, o caso em que aquilo que é identificado como entrada de dados e saída de dados também se alterna, ora sendo uma coisa, ora outra.

Como exemplo, os autores citam o controlador centrífugo (*fly-ball*), invenção de James Watt (1736 –1819), que regula o fluxo de vapor de um pistão para fazer girar uma roda. A roda move uma roldana, que move o controlador centrífugo. Este controlador centrífugo possui um eixo central com alavancas com duas pontas, cada uma delas contendo esferas de metal. Quando este eixo central gira em um ritmo mais acelerado, por rotação, a energia cinética das esferas aumenta, o que faz com que os braços da alavanca (dois braços em formato de tesouras) afastem-se do centro, movendo-se para fora e para cima – devido à força centrífuga das massas das esferas em movimento. Desse modo, vemos um movimento de cima para baixo, a partir do eixo central, um movimento de rotação sobre esse mesmo eixo central, e uma força centrífuga, que movimenta para fora ou para dentro as esferas, dependendo da velocidade do movimento de rotação. Os braços estão ligados a outro

6. A esse respeito, ver: LIPPMAN, Andrew. O arquiteto do futuro. In: **Meio & Mensagem,** São Paulo, n. 792, 26 jan. 1998, e MACHADO, Arlindo. **A arte do vídeo**. São Paulo: Brasiliense, 1990. Ver também: McCormack, Jon. Evolving Sonic Ecosystems. In: A Adamatzky (ed), **The International Journal of Systems & Cybernetics** - Kybernetes, Vol 32, Issue 1/2, 2003, Emerald, Northampton, UK, ISSN: 0368-492X, 2003, p. 184-202 (disponível em: ‹http://www.csse.monash.edu.au/~jonmc/research/publications.html›).

7. DUBBERLY, Hugh, PANGARO, Paul and HAQUE, Usman. What is Interaction? Are There Different Types? Written for Interactions magazine by Hugh Dubberly, Usman Haque, and Paul Pangaro. In ‹http://www.dubberly.com/articles/what-is-interaction.html›

eixo, e suas mudanças de posição farão ampliar ou reduzir a válvula de entrada de vapor do pistão – um mecanismo acoplado ao anterior –, o que faz com que a roda gire com menor velocidade, quando há menos vapor, e com maior velocidade, quando há mais. Assim, o dispositivo de Watt utilizava o movimento de duas esferas, gerado pela rotação do mecanismo, para controlar uma válvula de alimentação de vapor. Quando a rotação aumentava, por exemplo, as esferas afastavam-se do seu eixo e um mecanismo de barras transmitia esse movimento para a válvula. Quando a roda diminuía, o controlador ampliava a abertura da válvula de entrada e saída de vapor, aumentando-o e, consequentemente, aumentando a velocidade da roda. Era este o pistão que fornecia o *input* para a roda, mas havia ainda o controlador, que traduzia o *output* da roda em *input* para o pistão. Tratava-se, enfim, de um sistema autorregulador (ou de autocorreção), pois ele se ajustava para manter a velocidade da roda em movimentos contínuos. Era um sistema de *feedback* clássico.

Entretanto, possuía um diferencial: o fato de que a saída de dados de um mecanismo oferecia a entrada de dados para outro mecanismo, agindo de forma a manter o sistema estável e fazendo com que o segundo mecanismo realimentasse o primeiro.

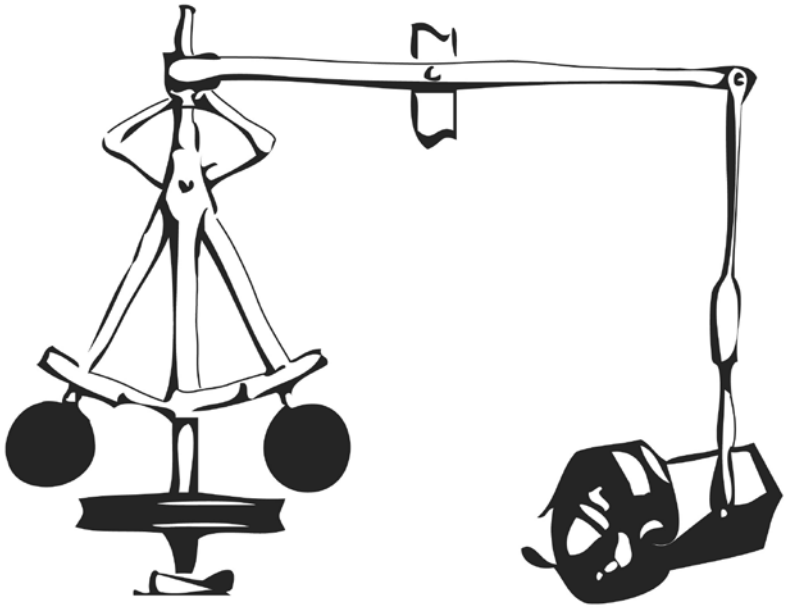


fig. 01
Esquema de funcionamento do controlador centrífugo (fly-ball), de James Watt.

Podemos pensar este sistema a partir do seguinte *script*:

```
Tome-se a velocidade da roda {
      1. Se velocidade atual for = velocidade de referência {
            Mantenha na posição a válvula de entrada de vapor}
                  Senão {
                        2. Se velocidade atual for > velocidade de
                           referência {
                                 Reduza a válvula do vapor }
                                       Senão {
                                             Aumente a válvula do
                                             vapor }
      }
Volte ao início
}
```

É claro que o mecanismo de Watt era totalmente mecânico, mas pensá-lo de acordo com um raciocínio lógico estruturado como esse permite esclarecê-lo melhor. Além do fato físico apontado, é possível perceber que, nos momentos em que uma saída de dados alimenta uma entrada de dados, há *loopings* condicionais alinhados ou, ainda, circularidades interdependentes: o **2** está contido no **1**. E, naturalmente, o motor a vapor não opera por sua própria vontade. Ele recebe a sua 'meta' de fora. Quem define a velocidade da roda é uma pessoa, ajustando a duração do vínculo entre o regulador centrífugo e a válvula de vapor; note-se, aqui, que neste caso a função de transferência é alterada – o que é entrada de dados acaba tornando-se saída, e vice-versa.

Esse modelo de motor a vapor tem a mesma estrutura do modelo clássico de interação descrito anteriormente do elevador! Ambos são sistemas de autorregulação, sistemas cibernéticos de primeira ordem. Apesar disso, não apenas os ligamos (liberando energia) ou os desligamos (interrompendo o fluxo de energia); pois, depois de ligados, os sistemas se mantém estáveis, autorregulando suas próprias ações, mantendo uma meta desejada e corrigindo os desvios de suas metas que possam ocorrer automaticamente. Mas ações automáticas não significam ações autônomas.

## 3. Tipos de Sistemas

Os autores do texto-base deste trabalho começam então a determinar tipos de sistemas:

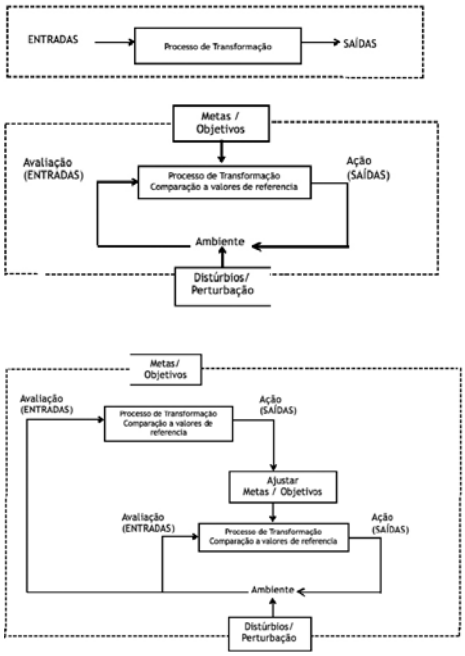
1. Temos aqueles que apenas **reagem** e aqueles que **interagem**, conforme já apresentado anteriormente. E, dentre eles, podemos distinguir, então, os sistemas que são lineares com circuitos abertos, dos sistemas de circuitos-fechados;
2. A meta definiria a relação entre o sistema e o ambiente. Esta

relação é o que regula o sistema e o que pretende mantê-lo constante em face das forças externas. Um sistema de autorregulação simples (com apenas um *looping* de mão única) não pode ajustar a sua própria meta. Seu objetivo só pode ser ajustado por algo que esteja fora do sistema;

3. Para termos um sistema de segunda ordem, efetivamente, temos que ter estes ajustes precisam ser realizados por algo que seja interno ao sistema (princípio da segunda cibernética), e é preciso também que o sistema seja um sistema que aprende – chamam de "aprendizagem" a modificação de metas, por parte do próprio sistema, com base no efeito de suas ações. O que significa também que metas podem ser modificadas (ajustadas) durante o processo. Temos que ressaltar que, no mecanismo a vapor apresentado, as metas eram estabelecidas por alguém de fora do sistema, e as mudanças nestas metas, movidas pelo efeito das ações do sistema, eram também realizadas por alguém fora do sistema. Então, quem na verdade estava "aprendendo", era aquele que estava tomando essas decisões e orientando o mecanismo, mas não o mecanismo em si. Sistemas que aprendem devem ter autonomia nesse processo. Somente assim os mecanismos deixariam de serem apenas *autômatos*, passando a ser também *autônomos*.

Alguns sistemas de aprendizagem agem sobre sistemas múltiplos, e isto a partir de sistemas de autorregulação de primeiro nível. Para atingir seus objetivos, o *sistema de segunda ordem* escolhe quais *sistemas de primeira ordem* deve ativar a cada momento. O *sistema de segunda ordem* persegue o seu objetivo dentre as opções de comparação de outros *sistemas de primeira ordem* contidos nele mesmo, enquanto um todo, que aprende como sua ação estará afetando todo o resto. "Aprender", neste caso, significa saber quais os *sistemas de primeira ordem* podem ajustar metas e corrigir os distúrbios que venham a ocorrer, e isto ocorre porque, de algum modo, o *sistema de segunda ordem* é capaz de "lembrar" quais foram os *sistemas de primeira ordem* tiveram sucesso no passado. Ou seja: "aprender" significa saber quais *os sistemas de primeira ordem* que podem melhor resolver os distúrbios levando-se em comparação àqueles que alcançaram sucesso na solução do mesmo problema anteriormente. Os autores do texto-base apresentam alguns tipos de sistemas, facilmente compreensíveis a partir de diagramas de suas ideias:

a)Sistema Linear
b)Sistema de autorregulagem
c)Sistema de aprendizagem




O que estes diagramas nos fazem perceber é que um *sistema de segunda ordem* contém aninhado outro sistema de autorregulação, que é *de primeira ordem*. O mesmo pode ocorrer em níveis adicionais, em diversas camadas e classes. Se retomarmos aos *scripts* apresentados anteriormente, poderemos rapidamente estabelecer correlações entre os sistemas e a linguagem de programação.

## 4. Combinação de Sistemas

É claro que essa divisão não encerra a questão. Os autores vão mais além, e propõem a seguinte combinação sistêmica. Se chamarmos os sistemas dinâmicos de a) lineares de ordem "0", b) de autorregulagem de ordem "1" e c) de aprendizagem de ordem "2", nós poderemos combiná-los em seis pares: "0-0", "0-1", "0-2", "1-1", "1-2" e "2-2". Vamos apresentar alguns exemplos, os mais significantes para nossos propósitos dentre aqueles citados pelos autores do texto-base.

0-0 – Reação

No caso da Reação, a saída de um sistema linear fornece insumos para outro, por exemplo, o sinal de um sensor que abre uma porta

do supermercado, por exemplo, que parte do princípio de uma ação que provoca uma reação. O primeiro sistema empurra o segundo, de modo que o segundo sistema não tem escolha em sua resposta. Esse tipo de interação é limitado e pode ser observado em muitos casos, como, por exemplo: a) ligando-se um vídeo com um controle remoto; b) um botão de "ir para a próxima página" numa consulta na internet; c) um sensor de presença que dispara um vídeo etc. Cabe ressaltar que nenhum julgamento de valor está sendo feito à obra em si, quando nos referimos a um tipo de 'interação limitado'. Muito pelo contrário, não há como avaliar uma obra de arte através de critérios como esse. Apenas tentamos nos aproximar desses dispositivos lógicos para melhor entender os meandros desse universo.

0-1 – Regulação (Correção)

A saída de um sistema linear fornece uma entrada para um sistema de autorregulação. O sistema de autorregulação funciona principalmente para combater distúrbios ocorridos no processo. No caso do motor a vapor, um aumento de resistência ao giro da roda pode ser uma perturbação controlável pelo sistema. Este aumento de resistência faria diminuir a velocidade do giro e, consequentemente, faria com que a válvula acoplada ao mecanismo liberasse mais calor, corrigindo sua velocidade, superando a resistência e voltando, assim, aos seus valores de referência. Nesse tipo de combinação, o papel do sistema linear pode ser visto como parte da autorregulação do sistema, um tipo de ligação entre o que vai dar o "início" ao sistema de correção até que este atinja seus valores de funcionamento. É também o caso do modo como um elevador, de que falávamos anteriormente, corrige seu destino sempre que chega a um andar: O elevador, até chegar ao andar solicitado, é um sistema linear; depois, o mecanismo de correção de desvios é acionado, e é ele o responsável por se certificar de que o elevador esteja na exata posição que deveria estar naquele momento.

Podemos pensar trabalhos que se utilizam do computador também a partir deste caso de Regulação. Se ligarmos um vídeo com controle remoto, e, sempre que o vídeo acabe, acionarmos novamente o "*play*", produziremos um estado em *looping,* embora bastante limitado. Mas, podemos criar um recurso que deixe o vídeo em eterno retorno – o que significa criarmos um mecanismo que automaticamente retorne o vídeo ao início assim que ele chegue ao final. A segunda possibilidade parece mais interessante, mas continua mesmo assim ainda muito limitada. Vamos aos próximos exemplos.

0-2 e 1-2 – Aprendizagem

A saída de um sistema linear fornece uma entrada para um

sistema de aprendizagem, no primeiro caso (0-2). Se, além disso, o sistema de aprendizado também conferir insumos para o sistema linear, fechando o ciclo, poderemos formar um sistema de regulação onde o sistema de aprendizagem avalie o impacto de suas ações e "aprenda", sendo capaz de ajustar suas metas (1-2).

Hoje, grande parte da interação humano-computador é caracterizada por um sistema de aprendizagem que interage com um processo linear. Você (o sistema que aprende) carrega um aplicativo em seu computador (o processo linear) que responde, e você reage. Depois que você trabalhou com este programa inúmeras vezes você acaba criando, em sua mente, um modelo de como trabalhar com aquela máquina. A partir daí, é claro que você aprendeu o sistema e passará a corrigir os distúrbios que vierem a ocorrer, contando com sua memória dos processos que obtiveram sucesso no passado... Mas o computador não "aprendeu você". Essa forma de interação (e os autores já começam a falar em interação, neste nível) é ainda assim limitada, pois, efetivamente, apenas um dos *interatores* "aprende"; mas já começa a se diferenciar das demais.

Agora, pensemos num programa de edição de texto (como o *Microsoft Word*) que "*corrija nossos erros"* através de um recurso de autocorreção. Este recurso pode ser uma ferramenta muito útil, principalmente quando bem definidas as alterações automáticas desejadas – ou seja, mudanças automáticas podem ser ativadas e desativadas conforme as preferências de quem escreve. Correções ortográficas e de digitação facilmente serão executadas; erros comuns (como digitar "masi", querendo dizer "mais", ou "qeu", quando na verdade pretende-se escrever "que") serão corrigidos imediatamente. Regras gramaticais também podem ser revisadas, e o programa chegará até a sugerir correções. Podemos ainda acrescentar palavras, sugestões, e também armazenar novos itens de autocorreção. Mas, apesar disso, o programa estará apenas articulando ajustes de primeira ordem. Poderíamos ter então uma mudança significativa de 0-2 para 1-2, mas isto não configura o *sistema-Word* como se fosse capaz de adquirir aprendizagem.

Serviços de busca funcionam da mesma maneira. O *Google* obtém a resposta para uma consulta de pesquisa, mas trata nossa milésima consulta do mesmo modo que tratou nossa primeira. Ele pode até mesmo gravar nossas ações, mas ainda assim não "aprendeu", pois não tem metas para modificar. Isto é verdadeiro mesmo com a adição de dados comportamentais para modificar *ranking* dos resultados, porque aí só existe inferência estatística e nenhum *feedback* direto que confirmaria se os objetivos foram ou não alcançados. Podemos citar, por exemplo, os agentes da empresa "*Amazon.com"*, que identificam nosso perfil e acabam enviando

sugestões de leituras adicionais àquelas que nós estávamos procurando, baseados em nossos dados armazenados. Os programas que realizam tais funções são chamados "agentes inteligentes"[8]. Esses agentes trabalham com estatísticas e não recolhem, por exemplo, formas de avaliar se os objetivos foram alcançados ou não – condição necessária para a modificação de suas metas (a comparação entre objetivos desejados e alcançados). Ou seja, quando os autores do texto-base citam o exemplo de agentes na rede, eles consideram, por exemplo, que tais agentes só "aprenderiam", no caso de compras realizadas por usuários, se fossem capazes de avaliar se a partir das sugestões apresentadas, outras compras foram feitas, e em qual proporção; se fossem capazes de comparar esses resultados e, então, adquirir uma estratégia para as novas sugestões. Somente deste modo, insistimos, tais agentes estariam aprendendo, pois as sugestões não seriam efetuadas somente a partir da estatística de "*todos os que compraram este livro compraram estes outros também*"'.

8. A esse respeito, ver: JOHNSON, Steven. **Cultura da Interface.** Rio de Janeiro: Jorge Zahar Editor, 2001.

2-2 Conversação

Considerarmos a saída de um sistema de aprendizagem tornar-se entrada para outro sistema de aprendizagem seria a combinação sistêmica mais complexa dentre as citadas. É o caso, conforme os autores sugerem, de sistemas de aprendizagem organizados em equipes, de redes sociais e sistemas de aprendizagem organizados em comunidades. Tais sistemas aprendem não apenas descobrindo ações que possam manter suas metas sobre circunstâncias específicas, mas também por meio da troca de informações de interesse comum. Reconhecemos aqui um diferencial importante, mas não foram citados exemplos mais significativos do que aquele em que o computador (a rede) participa como mediador de uma conversação entre duas ou mais pessoas (equipes ou comunidades), o que valeria, em última instância, até mesmo a comparação com uma conversa telefônica. Neste caso, reconhecemos sistemas inteligentes (as duas pessoas conversando) que interagem e aprendem, mas acreditamos que o computador também possa ser um agente capaz de adquirir aprendizagem, e não apenas de desempenhar o papel de extensão semiótica do aprendizado do homem [9].

9. LAURENTIZ, Silvia. Processos computacionais evolutivos na arte. In: **Ars 2 – Revista do Departamento de Artes Plásticas ECA/USP**, Departamento de Artes Plásticas, São Paulo, 2003, p. 45-55.

## Algumas Considerações

*"Sistemas que aprendem"*, como se apresentou, possuem autonomia nos processos de interação, e isto já identifica uma grande diferença entre eles e os sistemas autômatos. Além disso, "*sistemas que aprendem"* remetem a uma estrutura complexa, que traz amalgamada em si mesma as anteriores – sistemas lineares, reagentes e de primeira ordem –, incorporando-as em

sua complexidade sistêmica. Melhor dizendo: sistemas autônomos precisam de sistemas autômatos para existir. O que surge como grata surpresa é a imediata relação com o pensamento dos signos de Charles Sanders Peirce. Está clara a relação entre causação final, inteligência, semiose e aprendizagem em sistemas autônomos, bem como sua dependência com a causação eficiente e mecânica dos sistemas autômatos.

Tudo isso nos permite reconhecer que a maioria dos sistemas que estão sendo utilizados atualmente pela computação ainda é sistema autômato: aqueles que funcionam por equilibração, comparação e ajustes de padrões de referência, correção de desvios e distúrbios, regulação automática, ajustes por médias comparadas, eliminação de ruídos, mudanças através de *ranking* de resultados e inferência estatística. É claro que neste momento estamos ampliando o conceito de "autômato", que é tratado pela ciência da computação como aquela teoria que estuda modelos matemáticos em máquinas de estados finitos. Mas, mesmo nesses casos, há autômatos que já utilizam linguagens mais complexas.

Quanto àqueles sistemas considerados "inteligentes" pelo mercado[10], eles não passam de sistemas que atuam dessa maneira. Assim, é preciso notar que, quando nos deparamos com termos como "edifícios inteligentes"', "agentes inteligentes", "máquinas inteligentes", "dispositivos inteligentes", estamos diante, na verdade, de sistemas ainda muito aquém de possuírem autonomia e aprendizagem nos moldes que foram colocados neste texto.

Então, se Júlio Plaza[11] alertou para os graus de abertura de uma obra, analisando as relações autor-obra-recepção, apresentando 3 graus de abertura (subjetiva, participativa, interativa) relativos à recepção e associados diretamente às fases produtivas da arte; se Edmond Couchot[12] orientou-nos para a relação do interator no sistema, já apontado pela cibernética de segunda ordem, que sugeriria uma segunda interatividade, posicionando o sujeito dentro da obra; se Janet Murray[13] demonstrou o potencial de agência de sistemas interativos, aquele em que o sujeito-interator age sobre a obra com ações de fato significantes, onde agência não seria simplesmente "livre arbítrio", mas "*a satisfação de ter o poder de uma ação significante e de ver os resultados de nossas decisões e escolhas*"; se Andy Lippman[14] disse ainda que retroalimentação (*feedback*) genuína, aquela da interatividade de fato, requer "*atividade mútua e simultânea por ambos os participantes*"; e se Milton Sogabe[15] apresenta sua perspectiva histórica da pintura medieval às instalações, apontando uma passagem da obra se relacionando com o *corpo do artista* até aquela que se relaciona com o *corpo do público*; apresentamos neste artigo um potencial em se criar obras que sejam efetivamente autônomas, em que os diferentes ti-

10. Hoje encontramos inúmeras possibilidades de automação industrial, residencial, predial, etc, que apresentam seus produtos como 'sistemas inteligentes'. O mercado ainda anuncia eletrodomésticos inteligentes: geladeiras, televisão, torneiras...

11. PLAZA, Júlio. Arte e interatividade: Autor--obra-recepção. In: **Revista do Mestrado em Arte e Tecnologia da Universidade de Brasília,** VIS, v. 3, n.3, UnB, p. 29-42, 2001.

12. COUCHOT, E.; TRAMUS, M.; BRET, M. A segunda interatividade - em direção a novas práticas artísticas. In: DOMINGUES, Diana (Org). **Arte e vida no século XXI**. UNESP, São Paulo, 2003, p. 27-38.

13. MURRAY, Janet H. **Hamlet on the Holodeck – The Future of Narrative in Cyberspace**. Massachusetts: MIT Press, 2001.

14. LIPPMAN, Andrew. Op.Cit.

15. SOGABE, Milton. O corpo do observador nas artes visuais. In: **ANPAP, 16° Encontro Nacional da Associação Nacional de Pesquisadores de Artes Plásticas Dinâmicas Epistemológicas em Artes Visuais** – 24 a 28 de setembro de 2007 – Florianópolis.

pos de interação é que estabelecem os limites, graus de liberdade e ações significantes.

Tomando-se procedimentos que evoluem no tempo, definidos por uma coleção de valores *X,* indexados por um parâmetro de tempo (*t*), e sendo o conjunto de valores que *Xt* pode assumir seu *Espaço de Estados*; perceber essa dimensão de um projeto não definirá a obra do artista, mas dará um salto em seu campo de possibilidades. E mais: pensar a função de *Transição de Estados* a partir de diferentes interações entre os elementos de um sistema, definida por metas, relação entre sistema e ambiente, aprendizagem a partir da modificação de metas com base no efeito das ações do sistema, requestiona a própria condição sígnica da obra. Pensar esses processos é uma tremenda reviravolta para o artista, que nunca esteve tão próximo a essas atividades lógicas, mas é um caminho possível para desmembramentos estéticos. Não estamos reivindicando um *Tratado de Lógica*, nem um conhecimento especializado em outra área, mas entender dos procedimentos tecnológicos e deixando de olhar para noções comuns passando a um conhecimento mais rigoroso, o que aumentará as possibilidades de ações criativas, o potencial expressivo dos projetos. Em contrapartida, aumentam o conhecimento, habilidades e sensibilidades de quem destes compartilham.

Silvia Laurentiz é professora do Departamento de Artes Plásticas e do Programa de Pós-Graduação em Artes Visuais da ECA-USP. Coordenadora do Grupo de Pesquisa *Realidades–das realidades tangíveis às realidades ontológicas* (www.eca.usp.br/realidades). Artista Multimídia com trabalhos em realidade virtual, aumentada, ambientes interativos, multimídia e *web art*.

## Referências Bibliográficas

JOHNSON, Steven. **Emergência – a dinâmica de rede em formigas, cérebros, cidades e softwares.** Rio de Janeiro: Jorge Zahar Editor, 2003.

LAURENTIZ, Silvia. Poesia Digital: uma estreita relação entre a poética e os códigos da lógica computacional. In: **16° Encontro Nacional da Associação Nacional de Pesquisadores de Artes Plásticas Dinâmicas Epistemológicas em Artes Visuais** – 24 a 28 de setembro de 2007 – Florianópolis.

LAURENTIZ, Silvia. Cibernética na Arte-da teoria às experimentações artísticas, 2007. Disponível em: ‹http://www.upgradesaopaulo.com.br/e-magazine/pdf/laurentiz_cibernetica.pdf›

PEIRCE, Charles Sanders. **The electronic edition of The collected Papers of Charles Sanders Peirce**. Utah: Folio Corporation (Vol. I-VI edited by Charles Hartshorne e Paul Weiss; vol. VII-VIII edited by Artur W. Burks); Harvard University Press, 1994.

***Dyn*LAB project**. Course on Dynamics of multidisplicinary and controlled Systems - DynLab - Pilot Project No: CZ/02/B/F/PP/13400. Leonardo da Vinci Programme Pilot Project No: CZ/02/B/F/PP/134001 by C. Schmid, Lehrstühl für Automatisierung und Prozeßsinformatik, Ruhr-Universität Bochum, in ‹http://virtual.cvut.cz/dynlabmods/syscontrol.pdf›, ‹http://virtual.cvut.cz/dynlabcourse/› by C. Schmid, Lehrstühl für Automatisierung und Prozeßsinformatik, Ruhr-Universität Bochum.