
Ambientes Computacionais Orientados a Agentes para apoio à Aprendizagem Baseada em Simulação

Lívia Lopes Azevedo^{1,2}, Crediné Silva de Menezes^{2,5}, Beatriz Corso Magdalena³, Edilson Pontarolo^{4,5}

¹Instituto de Ciências e Letras do Médio Araguaia (ICLMA) – UFMT
Rod. MT 100, km 3,5 – CEP 78600-000 – Pontal do Araguaia – MT – Brasil
PPGEE – Universidade Federal do Espírito Santo (UFES)

²Centro Tecnológico – PPGEE – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, s/n – Campus de Goiabeiras
CEP 29060-900 – Vitória – ES – Brasil

³Laboratório de Estudos Cognitivos
Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – CEP 91501-970 – Porto Alegre – RS – Brasil

⁴Centro Federal de Educação Tecnológica do Paraná (CEFET-PR)
Via do Conhecimento, km 1 – CEP 85503-390 – Pato Branco – PR – Brasil

⁵Programa de Pós-Graduação em Informática na Educação (PGIE) – UFRGS
Av. Paulo Gama, 110 – prédio 12105 – sala 332
Caixa Postal 5071 – CEP 90041-970 – Porto Alegre – RS – Brasil

livia@cpd.ufmt.br, credine@inf.ufes.br, beamaq@terra.com.br,
edi@pb.cefetpr.br

Abstract: *The use of simulation has become an alternative for helping the understanding of our universe phenomena being particularly useful in the exploration of complex systems. Nowadays, one of the simulation domains to be highlighted is the one that aims at modeling the world through the interaction of the different dwelling agents. Simulation environments make up an important tool in supporting learning. In this article, the main learning concepts of this approach and the pedagogic use of the simulation as well as highlight NetLogo, as a support environment the learning based on simulation.*

Resumo: *O uso da simulação vem se tornando uma alternativa para ajudar a compreender os fenômenos de nosso universo, sendo particularmente útil na exploração de sistemas complexos. Uma das vertentes da simulação atualmente em destaque é aquela que busca modelar o mundo através das interações dos vários agentes que o povoam. Ambientes de simulação se constituem hoje em uma importante ferramenta de apoio à aprendizagem. Neste artigo, apresentamos os principais conceitos desta abordagem e o uso pedagógico da simulação bem como destacamos o NetLogo, como um ambiente de apoio à aprendizagem baseada em simulação.*

Palavras-chave: Ambiente de modelagem, sistemas complexos, ferramentas de aprendizagem, programação multi-agente.

1. Introdução

Apesar de não ser novidade, é importante repetir que vivemos numa era de grandes mudanças na educação, as quais implicam novas adaptações e reflexões sobre os modelos já cristalizados do processo educacional. Segundo Lévy (1993), com o advento das novas tecnologias da informação e comunicação (TIC), é possível criar um novo e amplo espaço de possibilidades para a educação.

O uso de ferramentas computacionais tem mostrado ser um forte aliado no processo educacional, despertando nos estudantes, que possuem conhecimento, informação, interesse, preferências e disponibilidade distintas entre si, uma maior interação e aproveitamento dos objetos trabalhados. Porém, para conseguir atender essa diversidade, o sistema educacional deve procurar adaptar-se e tirar proveito das novas formas de trabalho, entre as quais se sobressai a simulação.

A simulação vem se apresentando como uma alternativa interessante para ajudar a compreender os fenômenos de nosso universo, particularmente útil na exploração de sistemas complexos, nos quais há interdependência e, até mesmo, dependência entre uma multiplicidade de fatores e agentes. Assim, parece particularmente interessante uma das vertentes da simulação, atualmente em destaque, que busca modelar o mundo através das interações dos vários agentes que o povoam.

Frente a esse desafio, o que propomos aqui é fazer uma apresentação do uso pedagógico da simulação e destacar um ambiente de apoio à aprendizagem baseada nesta abordagem. Este ambiente pode ser utilizado de forma interdisciplinar, como uma ferramenta de integração, capaz de despertar nos estudantes um novo modo de pensar, entender e explorar sistemas descentralizados e sistemas complexos.

O artigo está organizado da seguinte forma: na seção 2, apresentamos a importância do uso de modelos, seguida da explicação sobre simulação em sistemas complexos e em sistemas descentralizados; na seção 4, abordamos alguns ambientes de modelagem de agentes, exploramos o ambiente Netlogo e apresentamos experiências de aprendizagens com a criação de modelos NetLogo e encerramos com as considerações finais.

2. A importância do uso de modelos

Desde os primeiros tempos, os homens têm criado modelos como uma forma de representar, explicar e entender o mundo ao seu redor. No decorrer da história, foram criados muitos modelos de madeira, papel, metal e expressões matemáticas e sistemas teóricos. Alguns deles tornaram-se exemplares: a) Leonardo DaVinci, construiu o modelo da máquina de voar, que muitos declaram ter sido inspirado pelo seu desejo de entender o voo dos pássaros; b) Sir Issac Newton, descreveu o comportamento do sistema físico com um conjunto de equações; c) Adam Smith, reconhecido pela sua obra “Uma Investigação sobre a natureza e as causas da riqueza das nações”, tratou do controle do governo sobre a economia; d) Charles Darwin, explicou a teoria da evolução; e) Henri Poincaré, observou a idéia da dinâmica do Chaos (teoria do Chaos); f) Herbert Simon, Ilya Prigogine, Herman Haaken e outros têm pesquisado para a teoria da complexidade.

Esses modelos não apenas ajudaram seus criadores a entender melhor o fenômeno que estavam estudando, mas também ajudaram a comunicar novas idéias que resultaram da aplicação dos mesmos.

2.1. O uso dos computadores para estudo de modelos

Recentemente, os computadores têm fornecido um novo meio para construir, analisar e descrever modelos mais sofisticados e complexos, incapazes de serem produzidos com os materiais postos à disposição pela natureza e pela ciência já construída. Assim, com o uso do computador, historiadores têm construído modelos de civilizações antigas; biólogos têm construído modelos de seres vivos com o DNA; engenheiros projetam obras, entre outros. Os computadores também são vistos, para os iniciantes, como uma ferramenta que possibilita analisar, construir e explorar seus próprios modelos e aprender novas idéias científicas.

Papert [apud Valente et al 1985] ao observar crianças que tinham aprendido como programar em computador constatou que elas criavam e usavam modelos computacionais bem concretos para aprender sobre o aprender. Dessa ação resultavam novas idéias e formas de perceber os fatos.

Abrahamson e Wilensky (2005), através das atividades desenvolvidas no ProbLab (laboratório de aprendizagem com uso do computador) afirmam que alunos que trabalham com simulações têm apresentado melhor desempenho em resolver questões que envolvam conhecimentos de matemática e probabilidade, aliado a um aumento de ações colaborativas. Na mesma perspectiva, muitas atividades e seus resultados são apresentadas por Colella, Klopfer e Resnick (2001), utilizando ambientes de simulação para desenvolvimento de modelos e apoio à aprendizagem.

2.2. Tipos de modelos

Como já vimos, a criação de modelos, em particular na sala de aula, pode ser uma poderosa aliada na experiência de aprendizagem. Através de um modelo, é possível mostrar um exemplo ou uma idéia que o aluno tenha pensado ou mostrar dados que ele tenha coletado em uma investigação particular. No entanto, para que o processo de construção de modelos e/ou aplicação dos mesmos seja efetivo, os alunos devem ser acompanhados e desafiados pelo orientador (professor) de modo que, juntos (em um processo interativo), possam assegurar uma seqüência de desenvolvimento capaz de ajudar a explorar um novo tópico ou entender um fenômeno que está em estudo.

Colella, Klopfer e Resnick (2001) apresentam uma classificação dos modelos científicos em três categorias: a) **Ilustrativos** – modelos que capturam algum tipo de visualização de um sistema ou processo científico, ajudando a entendê-lo de um novo modo. Exemplo: modelos do sistema solar mostrando as órbitas dos planetas, um corpo humano transparente com seus órgãos internos; b) **Analíticos** – modelos que são baseados em equações matemáticas e possibilitam explorar uma variedade de cenários. Esses modelos geram soluções que predizem o comportamento de um sistema, baseado em um dado conjunto de condições; c) **Simulação** – modelos que, usando eventos probabilísticos e randômicos, refletem importantes aspectos do mundo ao nosso redor.

Por suas propriedades, a simulação possibilita algumas explorações que são difíceis de realizar através de modelos analíticos e impossíveis de demonstrar com modelos ilustrativos, tais como a interdependência (equilíbrio) entre populações

diferentes, habitantes de um mesmo ecossistema, e a sua variabilidade em função de agentes, internos ou externos ao sistema, intervenientes no mesmo.

3. Simulação em sistemas complexos e em sistemas descentralizados

Em sistemas descentralizados, a ordem pode emergir sem controle centralizado. São sistemas cuja complexidade deriva da existência de um tecido interdependente, interativo e retroativo que torna inseparáveis as partes e o todo, exigindo, portanto, uma visão multidimensional do sistema para que o mesmo possa ser melhor entendido.

Nessa perspectiva, um bom exemplo é um formigueiro. Nele, dezenas de milhares de formigas andam e mudam de direção como numa coreografia, com um movimento que parece em perfeita consonância. Nele, apesar de cada indivíduo ter uma função específica, não há qualquer organização central controlando as ações de cada um dos indivíduos. A formiga rainha da colônia não faz nada além de gerar mais indivíduos, ela não controla o que cada operária deve fazer. Ninguém diz quantas operárias devem trabalhar na reforma e expansão dos túneis do formigueiro, quantas farão a limpeza, quantas e quais formigas irão buscar comida. Há, isso sim, um sistema de coordenações entre os diferentes níveis do sistema que o tornam um sistema complexo auto-organizativo e regulador. O comportamento das formigas não é determinado pela ordem de uma formiga rainha (controle centralizado), mas pelas interações locais entre as milhares de formigas trabalhadoras. Segundo Minsky e Pappert (1989), podemos dizer, então, que um formigueiro é um exemplo de sistema complexo, pois o comportamento de uma formiga sozinha é relativamente simples, porém a organização de todo o formigueiro descreve um comportamento mais inteligente e de alto nível..

Da mesma forma, Resnick (1991) ressalta que em colônias de bactérias, congestionamento de trânsito, bando de pássaros voando, reações do sistema imunológico não há controle centralizado e os modelos desse tipo de sistema são determinados pelas interações locais entre os componentes descentralizados. Ruthen (1993), salienta ainda que nestes exemplos, todos são sistemas de muitos agentes, cada um interagindo com seus vizinhos e, mais importante, adaptando-se às mudanças.

Segundo Colella et al (2001) e Ruthen (1993), idéias sobre descentralização e auto-organização estão se difundindo, principalmente na comunidade científica. Biólogos, físicos, economistas, cientistas da computação, matemáticos e pesquisadores de muitas outras áreas têm unido esforços para desvendar os princípios subjacentes a tais sistemas adaptativos complexos. Suas pesquisas têm possibilitado um melhor entendimento do que é complexidade, porque ela emerge, naturalmente dos dados e porque os sistemas vivos parecem evoluir para o limite entre a ordem e o randômico.

A teoria da complexidade é um novo campo da ciência que estuda até que ponto as partes de um sistema se separam do comportamento coletivo, e como o sistema interage com seu ambiente. Sistemas complexos são difíceis de entender porque as causas e os efeitos não estão relacionados de forma direta, ou seja, existem correlações entre eventos distantes no tempo e no espaço. O campo dos sistemas complexos fornece um número de ferramentas sofisticadas. Algumas delas abordam conceitos que auxiliam a pensar sobre esses sistemas, outras oferecem elementos de análise para estudá-la e ganhar profundidade, enquanto outras, baseadas em computador, podem ser utilizadas para descrever, modelar ou simular esses sistemas, em ambientes virtuais.

A simulação, tem sido considerada um modo de fazer ciência, que pode ser contrastada com os modos indutivos e dedutivos. Axelrod (1997a) salienta que esse novo campo tem crescido muito e sua utilização tem se estendido aos diversos campos do conhecimento, facilitando a interação e interdependência entre diferentes especialistas. Nesta mesma obra Axelrod utiliza e comenta a definição de simulação dada por Bratley, Fox e Schrage, segundo a qual “(...) *simulação significa dirigir um modelo de um sistema com entradas satisfatórias e observar as correspondentes saídas*”. No entanto, ele ressalva que apesar desta definição ser útil, ela esconde a explicitação dos diversos propósitos para os quais a simulação pode ser utilizada: predição, desempenho, treinamento, entretenimento, educação, prova e descoberta.

Afirma ainda que a simulação como metodologia de pesquisa é um novo modo de conduzir a pesquisa científica, além da indução – descoberta de padrões em dados empíricos – e da dedução – conjunto de axiomas e a prova das conseqüências que podem ser derivadas dessas suposições.-

A simulação, semelhante a dedução, inicia com um conjunto explícito de suposições, mas difere dela por não provar teoremas. Ao contrário, gera dados que podem ser analisados intuitivamente. Diferente da indução típica, a simulação dos dados vem de um rigoroso e especificado conjunto de regras, antes que diretamente mensurados no mundo real. Enquanto a indução pode ser usada para encontrar padrões nos dados, a dedução pode ser usada para gerar conseqüências das suposições, a modelagem da simulação pode ser usada como um auxílio na intuição.

3.1. Simulação baseada em agentes

Um importante tipo de simulação é a modelagem baseada em agentes (“*agent-based modeling*”). Esse tipo de simulação é caracterizado pela existência de muitos agentes interagindo uns com os outros, com pouca ou nenhuma coordenação centralizada. A propriedade emergente do modelo baseado em agentes é que ela vem do baixo para cima (*botton-up*) e não do sentido de cima pra baixo (*top-down*).

Alexrod (1997b) afirma que a modelagem baseada em agentes, ainda que use a simulação, não necessariamente visa fornecer uma acurada representação de uma aplicação empírica particular. Ao invés disso, a meta da modelagem baseada em agentes é enriquecer nosso entendimento acerca de processos fundamentais que podem aparecer em uma variedade de aplicações.

Através da modelagem baseada em agentes é possível observar como os agentes individuais (pessoas, bactérias, insetos, nações, ou organizações) interagem entre si e com seu ambiente. A simulação no computador é então usada para descobrir propriedades do modelo e, assim, ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões.

Muitos pesquisadores têm demonstrado os benefícios da aprendizagem, numa ampla variedade de fenômenos científicos, através da abordagem multi-agente. Ressaltamos os trabalhos realizado nessa área através do “Media Lab” do Massachusetts Institute of Technology – MIT, do “Advanced Project Research Agency” através do Institute Santa Fe Institute, do “Center for connected learning and computer-based modeling” da Northwestern University, do “Advanced Learning Center” do Mesa State College, entre outros, todos usando ambientes de modelagem multi-agente.

Existem vários ambientes de programação especificamente projetados para modelagem baseada em agentes, entre eles: StarLogo, Microworlds, Swarm, RePast e NetLogo. Esses ambientes de modelagem multi-agente estão revolucionando a prática científica, até então utilizada para entender sistemas complexos do Universo. (Abrahamson e Wilensky 2005, Blikstein, Abrahamson e Wilensky 2005, Blikstein e Wilensky 2005, Levy, Kim e Wilensky 2004, Levy e Wilensky 2005). Da mesma forma, as perspectivas de trabalhar com sistemas complexos mediante o uso de simulações multi-agente estão ganhando importância para os educadores em todos níveis de ensino, que estão voltando sua atenção para essas novas ferramentas, que têm influenciado mudanças nas salas de aula (Tissue e Wilensky 2004). Para tirarmos proveito desta abordagem é importante dispor de ferramentas capazes de criar, no computador, simulações de fenômenos complexos e modelos descentralizados.

4. O ambiente NetLogo

O NetLogo é um ambiente de modelagem programável, para explorar e modelar comportamento de sistemas descentralizados e sistemas complexos, desenvolvidos no decorrer do tempo. O NetLogo fornece uma modo fácil de começar a modelagem baseada em agentes, mesmo para aqueles que não tem avançada habilidade em programação e matemática. O ambiente NetLogo é composto de três tipos de agentes: *turtles*, *patches* e *observer*. Os modeladores podem dar, simultaneamente, instruções a centenas ou milhares de “agentes” independentes que trabalham paralelamente, tornando possível explorar as conexões entre o comportamento de micro-níveis e os de modelos de macro-níveis que emergem das interações de muitos indivíduos. NetLogo usa uma linguagem de modelagem própria, derivada da linguagem de programação Logo do Media Lab – MIT. O Netlogo é uma ferramenta de acesso e uso gratuito, podendo ser acessado por: <http://ccl.northwestern.edu/netlogo/> .

Membros da comunidade NetLogo têm colocado a “tartaruga” em forma de pássaros, animais, carros, bactérias, células, robôs, planetas, formas geométricas, insetos, e muitos outros, uma vez que, além dos modelos disponíveis na biblioteca, é possível esboçar seu próprio modelo. Os “*patches*”, grade onde vivem as tartarugas, têm sido usadas para representar árvores, muros, casas, terrenos, céu, areia, etc. As tartarugas e os *patches* estão sendo usados para visualizar e estudar abstrações matemáticas, desenhos artísticos, jogos, ecossistemas e outros sistemas.

Com as versões tradicionais do Logo [Valente e Valente 1998] é possível criar formas geométricas e animações através de comandos atribuídos a uma tartaruga gráfica na tela do computador, que pode ser usada para representar muitos objetos do mundo: uma formiga, uma molécula, um carro. Seus comandos são muito próximos da linguagem natural, o que a torna uma linguagem de alto nível e de fácil aprendizado. Porém, o fato de dispor de poucas tartarugas dificulta o trabalho com fenômenos de maior complexidade, que envolvam um número maior de elementos e inter-relações.

O NetLogo estende a idéia da linguagem original permitindo o controle de milhares de tartarugas e *patches* simultaneamente, o que possibilita a interação entre eles, tornando o mundo da tartaruga computacionalmente ativo. O ambiente NetLogo é completamente programável e apresenta uma interface gráfica com botões de controle, sistema de gráficos, exportação e importação de modelos, entre outros.

4.1. Programação em NetLogo

A programação em NetLogo consiste em atribuir comportamentos a três grupos de entidades: O observador (*observer*), as criaturas (*turtles*) e o ambiente (*patches*). O observador, é um “criador” que especifica as condições de funcionamento e dá controle ao demais elementos. São os agentes as turtles (tartarugas) e os patches (grade do ambiente), que interagem no ambiente.

A linguagem de programação apresenta vários recursos, tais como: definir diferentes espécies (*breeds*) para os agentes (*turtles*), definir o comportamento a uma “espécie” de agentes, atribuir diferentes formas (*shapes*) para uma “espécie” de agentes (os modelos podem ser selecionados na biblioteca do ambiente ou criados pelo próprio usuário através de uma janela de edição de modelos). Por exemplo, definidas as espécies denominadas “lobo” e “ovelha”, através da programação é possível escrever uma regra que faça o “lobo” comer as “ovelhas”. Essa diferenciação das espécies faz, também, com que o modelo fique visualmente mais atraente e esclarecedor.

O NetLogo dispõe de uma característica diferenciada dos demais ambientes da mesma categoria que é denominada de *Agentset*. O que é importante no conceito de *Agentset* é que ele possibilita construir apenas algumas agentes com regras específicas. Por exemplo, todas as tartarugas com ID divisível por cinco, ou todos os *patches* com *pxcor* (valor da coordenada x) par. Um *Agentset* pode conter tanto tartarugas quanto *patches*, mas não ambos ao mesmo tempo.

4.2. Um modelo presa / predador

O modelo que apresentaremos é uma adaptação de um modelo apresentado na biblioteca explora a estabilidade de um ecossistema através da relação sistêmica presa / predador. Esses sistemas apresentam dois comportamentos: instável – quando o resultado da interação tende à extinção de uma ou mais partes envolvidas; estável – quando tendem a se manter no decorrer do tempo, mesmo com flutuações no tamanho da população. No exemplo abaixo, exploraremos especificamente o primeiro caso.

Os agentes que compõem o modelo são lobos, ovelhas e grama. O modelo inicia com todos os *patches* caracterizados de grama, sobre a qual estão dispostos os lobos e as ovelhas. A quantidade inicial de lobos e ovelhas é determinada pelo *observador* e a distribuição destes sobre o ambiente é aleatória. As ovelhas e os lobos caminham randomicamente sobre o ambiente, sendo que os lobos se alimentam das ovelhas e essas da grama. Quando criados, os lobos e as ovelhas recebem, cada um, uma quantidade de energia. É estabelecido um custo para cada movimento, que é reduzido da quantidade de energia existente, bem como atribuído um ganho de energia cada vez que eles se alimentam. Para permitir que a população continue, tanto lobo quanto ovelha têm a possibilidade de reproduzir um clone e essa reprodução ocorre quando eles atingem um nível de energia pré-estabelecido. Nesse momento, cada um dos lobos ou das ovelhas fica com metade da energia existente. Por outro lado, se a quantidade de energia for reduzida a zero, eles morrem. Essa variação produz uma população dinâmica, mas instável.

A visualização do comportamento e de suas mudanças são apresentadas na interface gráfica do ambiente, sendo possível controlar a velocidade do movimento dos agentes para uma melhor visualização do seu comportamento. A Figura 1 é um exemplo do exposto. Na interface gráfica estão representados os *patches* (ambiente) e as espécies (*turtles*). Os botões de comando permitem que se acione o modelo computacional.

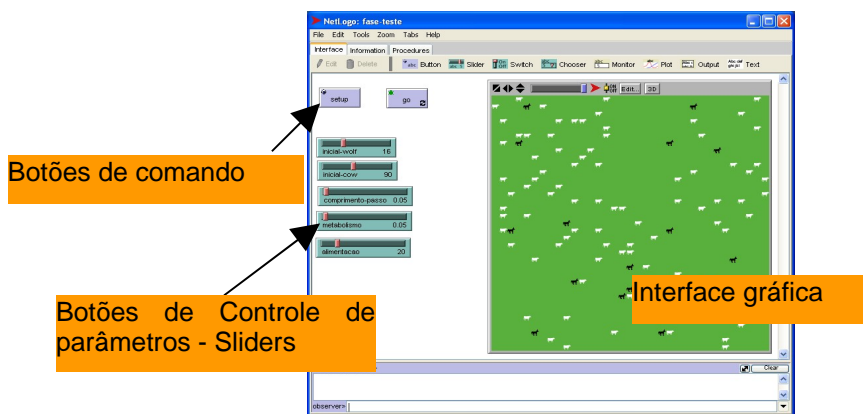


Figura 1. Interface do NetLogo – Exemplo presa / predador

O modelo é apresentado com valores iniciais para execução, bastando para isso acionar os botões de comando (setup e go) para iniciar o movimento dos agentes. O modelo dispõe também de botões de controle de parâmetros (sliders) que tem como objetivo permitir a mudança de parâmetros tais como: número inicial de ovelhas e lobos, ganho de energia com alimentação, perda de energia com o movimento (metabolismo), variação do movimento (comprimento-passo) que possibilitam observar as mudanças de comportamento e alterações no ambiente.

A visualização do comportamento e de suas mudanças são apresentadas na interface gráfica do ambiente, sendo possível controlar a velocidade do movimento dos agentes para uma melhor visualização do seu comportamento. A Figura 1 é um exemplo do exposto. Na interface gráfica estão representados os *patches* (ambiente) e as espécies (agentes). Os botões de comando permitem que se acione o modelo computacional.

4.2.1. Estudando com o modelo

Com esse modelo, pretendemos destacar o comportamento dos agentes (ovelha e lobo) na situação de presa / predador e suas relações de interdependência entre si e com a grama. Regras simples definem o movimento realizado por ambos, a disposição aleatória, e seus objetivos são a busca por alimento e manutenção da espécie. Através das variações de parâmetros é possível observar as alterações ocorridas no ambiente e a partir daí fazer várias conjecturas relacionadas aos comportamentos observados. Como nesse modelo não foi programada a renovação da grama – alimento das ovelhas – em algum momento as espécies se extinguirão. Porém é possível conjecturar sobre os valores iniciais dos agentes, a partir de algumas das questões que podem ser levantadas com o modelo: as variações do metabolismo levariam a modificações em seus comportamentos ou a extinção antecipada de uma das espécies? a quantidade de energia ganha com alimentação e o gasto de energia permitiria a reprodução? quais seriam os valores iniciais dos agentes e dos parâmetros para manter o sistema o maior tempo possível em funcionamento ou extinguir mais rápido? que papel teria a grama na estabilidade das espécies? que mudanças teriam que aparecer no sistema para que ele se torne estável? Que relações de interdependência precisam ser feitas para que haja equilíbrio entre as espécies lobo, ovelha e grama?

Cada um desses valores pode levar a uma série de discussões sobre o comportamento oscilatório no número dos indivíduos das populações, à medida que se processam as interações alimentares entre eles. A base dessas interações é a predação.

Assim, quando aumentam as chances de alimentação do agente ovelha pela mudança na quantidade de grama (nova regra que determina a reposição por crescimento nos *patches*) aumentam as chances de mais alimento e reprodução para o agente lobo. Como consequência, mais lobos determinam a diminuição nas ovelhas (nível trófico anterior) seguida de aumento dos produtores (grama). Essas alterações de quantidade entre os elementos da simulação podem levar a se alcançar um sistema estabilizado e a um entendimento do que significam relações de equilíbrio. Todas essas mudanças podem ser feitas com ou sem alteração no código do programa.

4.3. Um modelo de autômato celular - fractal

Estamos trabalhando com o ambiente NetLogo na exploração significativa de modelos relacionados a conceitos complexos, considerados de difícil compreensão. Pretendemos demonstrar aqui, a partir do exemplo de uma experiência particular com o conceito de *dimensão fractal no tempo* [Oliveira 1999], como a aprendizagem de tais conceitos pode ser explorada através da construção de modelos e simulações em NetLogo.

O conceito discutido a seguir baseia-se na idéia de autômato celular – um agente simples que armazena um único valor lógico por vez. Criamos um programa NetLogo que permite a criação de um determinado número de autômatos celulares (agentes *turtles*) que evoluem no tempo segundo determinadas regras lógicas simples e locais. O valor lógico de um autômato em um dado passo i depende apenas da aplicação de uma operação lógica entre os valores de seus vizinhos à esquerda e à direita no passo anterior $i - 1$. A representação visual do objeto gerado pela evolução iterativa e (localmente) interativa dos autômatos se dá por meio de um gráfico (*plot*), de forma que a cada passo de tempo discreto (ao longo do eixo x), cada autômato criado (ao longo do eixo y) marca um ponto no gráfico apenas se estiver “ligado” (*true*), e não marca nada caso contrário (*false*).

Neste caso, para demonstrar o conceito de um objeto de *dimensão fractal em relação ao tempo de evolução*, inicia-se a iteração ($i=0$) com apenas um autômato “ligado”. A cada passo na evolução do sistema, cada autômato assume o valor resultante da operação lógica **XOR** (“*eXclusive OR*”) entre seus vizinhos no passo anterior. Oliveira (1999) conclui que a massa de pontos m gerada por esse modelo varia em função do tempo t conforme a equação: $m = t^d$, onde $d = \log 3 / \log 2 = 1,5849625$

O expoente fracionário d demonstra a dimensão fractal do espaço ocupado pelo objeto gerado (massa de pontos) em função do tempo. Não satisfeitos com este conceito “incômodo e difícil de ser aceito”, fizemos em nosso programa o cálculo inverso, ou seja, calculamos o logaritmo da massa na base do tempo: $d = \log_t m$

Ao executar esta simulação observamos que a dimensão calculada oscilava por valores ao redor do valor 1,5849625 esperado. Isso ocorre porque esta dimensão fractal somente é válida para números de passos da seqüência de base 2, ou seja, somente para

$$t = 1, 2, 4, 8, 16, 32, 64, 128, \dots \quad t = 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, \dots$$

A Figura 2 demonstra a execução da simulação “autômatos celulares” após 512 passos, tendo sido gerado um objeto fractal de 19.683 pontos. Nota-se também claramente outra característica importante dos fractais, qual seja, a repetição de padrões entre o objeto e suas partes constitutivas.

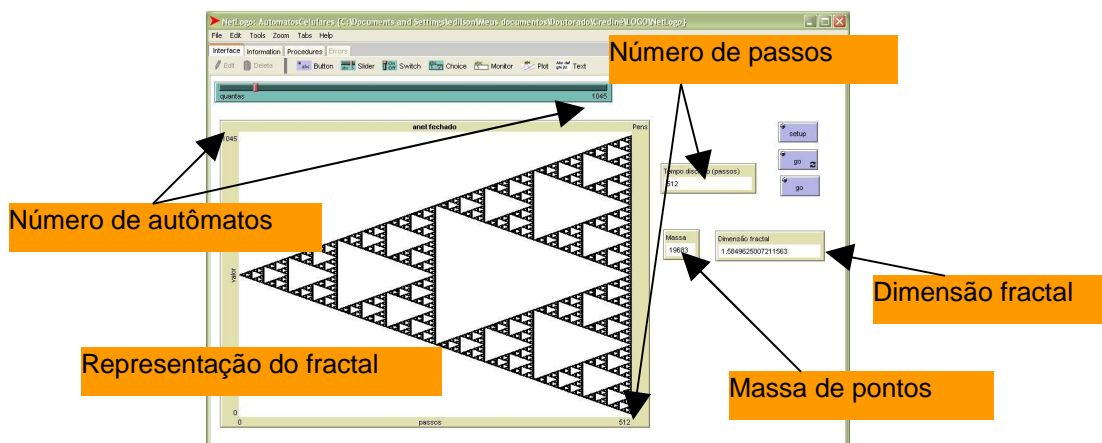


Figura 2. Simulação de um fractal por evolução de autómatos celulares em NetLogo.

5. Considerações finais

Este texto teve como objetivo ressaltar a importância do uso de ambientes computacionais na educação para que a aprendizagem dos alunos seja para buscar e encontrar respostas para se adaptarem a um mundo em constante mutação, a partir de ações em que sejam necessários processos de cooperação entre todos na sala de aula.

Nessa perspectiva, reforçamos a importância do uso de modelos na aprendizagem, dando atenção especial aos modelos de simulação, destacando aqueles especificamente projetados para modelagem de sistemas complexos e sistemas dinâmicos (modelagem baseada em agentes), capazes de simular situações complexas que compõem o universo de vida dos alunos, até então impossíveis de serem desenvolvidas e controladas em laboratórios escolares. Assim, nessas simulações é possível por em ação as mais variadas hipóteses sobre a realidade e, mediante processos ativos e construtivos de aprendizagem cooperativa, analisar os dados coletados para que seja determinada a viabilidade das mesmas como respostas aos problemas em questão.

Apesar de apenas tangenciar o tema, esperamos ter levantado o desafio da mudança da sala de aula, na qual: as grades programáticas sejam rompidas, diminuindo a quantidade e dispersão e aprofundando a qualidade das informações e conteúdos selecionados; o trabalho seja centrado no lançamento e testagem de hipóteses que são levantadas a partir de problemas tirados do mundo real; o processo de aprendizagem seja desenvolvido mediante processos ativos e construtivos de aprendizagem cooperativa e ações práticas de simulação no ambiente, físico ou virtual, de laboratório.

Referências

- Abrahamson, D., Wilensky, U. (2005) *ProbLab goes to school: Design, teaching, and learning of probability with multi-agent interactive computer models*. Proceedings of the Fourth Conference of the European Society for Research in Mathematics Education. San Feliu de Guixols, Spain. (Manuscript in press)
- Axelrod, R. (1997a) *Advancing the Art of Simulation in the Social Sciences*. In: Rosario Conte, Rainer Hegselmann and Pietro Terna (eds.), *Simulating Social Phenomena*. Berlin: Springer, 1997. pp. 21-40. Disponível em:
- Axelrod, R. (1997b) *Resources for Agent-Based Modeling*. Published as Appendix B of Robert Axelrod, *The Complexity of Cooperation: Agent-Based Models of*

-
- Competition and Collaboration Princeton, NJ: Princeton University Press.
<http://www.modelingcomplexity.org/pols490/Resources.pdf>
- Blikstein, P., Abrahamson, D., Wilensky, U. (2005) *NetLogo: Where We Are, Where We're Going*, In M. Eisenberg & A. Eisenberg (Eds.), Proceedings of the annual meeting of Interaction Design and Children 2005. Boulder, Colorado, disponível: <http://ccl.northwestern.edu/papers/>
- Blikstein, P., Wilensky, U. (2005) *Less is More: Agent-Based Simulation as a Powerful Learning Tool in Materials Science*. Proceedings of the IV International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, Holland. (Manuscript in press) Disponível em: <http://www.blikstein.com/paulo/documents/papers/BliksteinWilensky-LessIsMore-AAMAS2005.pdf>
- Colella, V. S., Klopfer, E., Resnick, M. (2001) *Aventures in modeling: exploring complex, dynamic system with StarLogo*, Teachers College Press, NY.
http://www.modelingcomplexity.org/pols490/adapting_to_complexity.pdf
<http://www.modelingcomplexity.org/pols490/AdvancingArtofSim.pdf>
- Lévy, P. (1993) *As tecnologias da inteligência: o futuro do pensamento na era da informática*. Rio de Janeiro: Ed. 34.
- Levy, S. T., Wilensky, U. (2005) *An analysis of student' patterns of exploration with NetLogo models embedded in the Connected Chemistry environment*. The annual meeting of the American Educational Research Association, Montréal, Quebec, Canada, April 11 - 15. Disponível em: <http://ccl.northwestern.edu/papers/AERA05-LevyWilensky.pdf>
- Levy, S. T., Kim, H., Wilensky, U. (2004) *Connected Chemistry - A study of secondary students using agent-based models to learn Chemistry*. In: J. Gobert (Chair) and N. H. Sabelli (Discussant) Modeling Across the Curriculum (MAC): Technology, Pedagogy, Assessment & Research. The annual meeting of the American Educational Research Association, San Diego, CA, April 12 - 16, 2004.
- Minsky, M., Pappert, S. (1989) *A Sociedade da Mente*. Rio de Janeiro: Francisco Alves.
- Oliveira, P. M. C. (1999) Autômatos Celulares. In: Nussenzveig, H. M. (org.). *Complexidade & Caos*. Rio de Janeiro: Editora UFRJ/COPEA.
- Resnick, M. (1991) *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. The MIT Press, Massachusetts, USA.
- Ruthen, R. (1993) *Adapting to Complexity*. Scientific American, January 1993. Disponível em:
- Tisue, S., Wilensky, U. (2004) *NetLogo: Design and Implementation of a Multi-Agent Modeling Environment*. Proceedings of Agent 2004, Chicago, October, 2004.
- Valente, J. A., et al 1985, LOGO: Computadores e Educação, Editora Brasiliense, São Paulo 1985
- Valente, J. A., Valente, A. B. (1998) *LOGO: conceitos, aplicações e projetos*. São Paulo: McGraw-Hill.