

Dinamica EGO, uma plataforma para modelagem de sistemas ambientais

Hermann Oliveira Rodrigues
Britaldo Silveira Soares-Filho
William Leles de Souza Costa

Centro de Sensoriamento Remoto, Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Belo Horizonte, 31270-900
{hermann, britaldo, william}@csr.ufmg.br

Abstract. Simulation models can be envisaged as a heuristic device useful for assessing, from long and short-terms, the outcomes from a variety of scenarios, translated as different socioeconomic, political, and environmental frameworks. A special class among them, spatially explicit models simulate the dynamics of an environmental system, reproducing the way its spatial patterns evolve, to project the probable ecological and socioeconomic consequences from the system dynamics. In this context, we have developed a spatially explicit simulation model of landscape dynamics – Dinamica, as a tool to investigate trajectory of landscapes and dynamics of spatial phenomena. In this paper, we present its new version called Dinamica EGO – acronym for Environment for Geoprocessing Objects. In this version, Dinamica was totally reengineered. The software environment holds a series of algorithms called *functors*. Each algorithm performs an algebra map algebra operator. To date, we have implemented the most common spatial analysis operators available in commercial GIS, plus a series of operators especially designed for spatial simulations, including calibration and validation methods. These functors are sequenced to establish a data flow in form of a graph. Through this graphical interface one can create models by simply dragging and connecting functors via their ports, which represent connectors to types of data, such as maps, tables, matrices, mathematical expressions, and constants. Thus models can be designed as a diagram and execution follows a data flow chain. This friendly interface allows the design from simple to very complex spatial models that are saved in a script language. As a result, Dinamica EGO software favors simplicity, flexibility, and performance, optimizing speed and computer resources, such as memory and parallel processing. To illustrate its application potential, we present a version of SimAmazonia deforestation model implemented on Dinamica EGO platform, describing the *functors* employed in the design of this model.

Palavras-chave: Dinamica EGO, spatial simulation model, environmental modeling, deforestation, SimAmazon, modelagem de sistemas ambientais, modelo de simulação espacial, desmatamento, SimAmazonia.

1. Introdução

Modelos de simulação espacial visam auxiliar o entendimento dos mecanismos causais e processos de desenvolvimento de sistemas ambientais, e assim determinar como que eles evoluem diante de diferentes cenários que se traduzem por quadros socioeconômicos, políticos e ambientais. Em efeito, modelos de simulação espacialmente explícitos têm recebido atenção por parte de pesquisadores de diversas áreas, sendo usados para avaliação de complexas questões ambientais, como no exemplo da influência das mudanças climáticas no turismo da ilha de Santa Lúcia (White *et al.*, 2000), determinantes espaciais do desmatamento nas Filipinas e Malásia (Verburg *et al.*, 2002) e resposta da paisagem a diferentes cenários de gerenciamento de bacias hidrográficas (Costanza *et al.*, 1992). Sob esse contexto, foi desenvolvido o software Dinamica como um instrumento para investigação de trajetórias de paisagens e dinâmica de fenômenos espaciais (Soares-Filho *et al.* 2002, Soares-Filho 2006). Dinamica foi concebido como um software genérico de modelagem de mudanças ambientais, por isso, ele tem sido aplicado a uma variedade de estudos que incluem a modelagem de expansão urbana e dinâmica intra-urbana (Almeida *et al.*, 2003; Godoy e Soares-Filho, no prelo), avaliação de propostas de zoneamento ecológico-econômico (Castro *et al.*, submetido) e modelagem de desmatamento tropical desde uma escala local a da bacia Amazônica (Soares-Filho *et al.*, 2002; 2004; 2006).

Neste artigo, é apresentada a nova versão do software Dinamica, intitulada Dinamica EGO – acrônimo para *Environment for Geoprocessing Objects* (ambiente para objetos de

geoprocessamento). Nessa versão, o Dinamica foi totalmente reescrito, acabando-se com a herança do fluxo estático de processamento da versão anterior. Essa nova versão muito mais complexa e flexível está sendo usada como plataforma do SimAmazonia 2, o qual integra uma variedade de modelos desenvolvidos por pesquisadores do projeto “Cenários para a Amazônia”, incluindo uma versão atualizada do modelo de desmatamento PanAmazônico (Soares-Filho *et al.*, 2006), integrada a modelos de rentabilidade da soja e da exploração madeireira e CARLUCC – um modelo que simula fluxos de carbono dentre os componentes da floresta e entre ela e a atmosfera (Hirsh *et al.*, 2004). Esses modelos são rodados simultaneamente trocando dados entre si. Para se ter uma idéia do nível de complexidade, todos os modelos juntos empregam mais de uma centena de mapas de entrada, cada um composto por uma matriz de 2103x1561 células. Este artigo descreve sucintamente a arquitetura do Dinamica EGO, ilustrando o seu potencial de modelagem através de sua aplicação para implementação do modelo de desmatamento SimAmazonia-1.

2. Arquitetura do Dinamica EGO

O ambiente de modelagem do Dinamica EGO envolve uma série de operadores chamados de *functores* (*functors*). Um *functor* pode ser entendido como um processo que atua sobre um conjunto de dados de entrada sobre o qual é aplicado um número finito de operações, produzindo como saída um novo conjunto de dados. Cada *functor* é responsável pela execução de uma tarefa específica, incluindo toda uma gama de operadores de álgebra cartográfica (Tomlim, 1990), tais como cálculo de mapa de custo, sorteio de pontos mais prováveis para a ocorrência de um determinado fenômeno, cálculo de caminho de menor custo e cálculo de mapas usando-se de expressões lógicas e aritméticas. A esse respeito, *functores* podem ser considerados como os elementos básicos de um modelo cartográfico. Até o presente, foram implementados os operadores de análise espacial mais comuns disponíveis em SIG comerciais, mais uma série de operadores especialmente desenhados para simulação espacial, incluindo métodos de calibração e validação (Soares-Filho, 2006).

Além dos operadores convencionais, chamados simplesmente de *functores*, existem os operadores de grupo, conhecidos como *containers*. Os *containers* são especiais, pois agrupam e determinam um comportamento para o conjunto de operadores contido nele. Exemplos de *containers* são os operadores “Repeat” que itera a execução do sub-modelo inserido nele, “Block” que simplesmente agrupa *functores* e “Region”, empregado para que uma determinada operação afete apenas uma região específica em um mapa. Os *functores* e *containers* recebem e enviam dados para outros *functores* e *containers* por meio de um conjunto de entradas e saídas chamados portos. Cada porto possui um tipo de dado associado, por exemplo: tabela, mapa, matriz, valor, etc. O tipo de porto determina seu modo de edição e visualização, sendo que para cada tipo há um editor e visualizador específico.

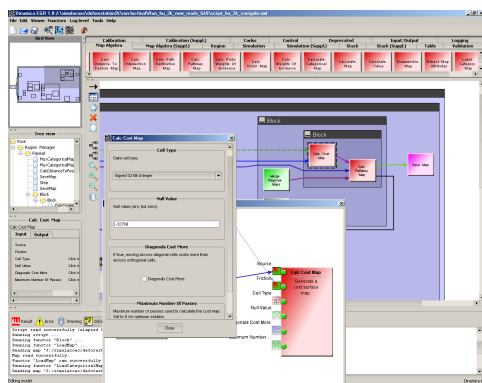


Fig. 1 – Interface gráfica do Dinamica - EGO

Para a criação de modelos, os *functores* e *containers* são seqüenciados em um fluxo de dados através da interface gráfica (Fig.1). Por essa interface, os modelos podem ser criados simplesmente, arrastando e conectando *functores* através de portos com tipos equivalentes. Os modelos são assim desenhados na forma de um diagrama e a execução segue a cadeia de fluxo de dados. Nessa interface, o fluxo de dados não é mais “oculto”, obrigando o usuário a construí-lo, o que torna sua flexibilidade direta-

mente proporcional à complexidade. Nos anos recentes, foram desenvolvidos vários softwares de programação baseados em linguagem de fluxo de dados, sobretudo para visualização de dados (ver exemplos em Abran *et al.*, 1996; Johnston *et al.* 2004), mas também para aplicações em geoprocessamento (Eastman, 2001). A premissa dessas linguagens é que elas deveriam ser suficientemente simples para permitir que pessoas com pouca ou nenhuma experiência em programação fossem capazes de construir aplicações. No entanto, um grande desafio ao desenvolvimento desse tipo de ambiente consiste em manter a sua simplicidade tradicional à medida que essa ferramenta se torna mais sofisticada e os modelos crescem em complexidade (Abran *et al.*, 1996). É nesse ponto que se faz necessária uma interface amigável e poderosa que permita ao usuário, quer seja iniciante ou avançado, implementar modelos de álgebra cartográfica de forma rápida e/ou intuitiva. Para tanto, a natureza da interface gráfica do Dinamica EGO foi baseada na teoria de grafos direcionados, nos quais a informação parte de um ponto, percorre um caminho guiado pelas conexões dos nós no grafo (*functores*) e chega a outro ponto no grafo, não necessariamente diferente do inicial. Assim, por meio de setas conectam-se os *functores* através de seus portos de entrada e saída, de acordo com a ordem de processamento desejada.

Portanto, a linguagem de fluxo de dados do Dinamica descreve o encadeamento e a relação entre os operadores. Um dos benefícios dessa representação interna é a simplificação da análise do tempo necessário que um resultado deve ser mantido na memória, uma vez que as dependências entre os operadores podem ser obtidas de forma bastante trivial, mesmo quando essa dependência envolve *containers* com um nível arbitrário de aninhamento. Isso faz com que Dinamica use os recursos de forma bastante racional, utilizando pouca memória mesmo em modelos bastante complexos. Além disso, a análise das dependências auxilia no cálculo da ordem de execução dos operadores. Essa abordagem é diferente da abordagem tradicional que usa uma tabela para determinar quando um operador pode ser executado (Wesley *et al.*, 2004). O cálculo prévio da ordem de execução permite que os operadores sejam agrupados de forma que um operador seja executado próximo daquele que utiliza o seu resultado, evitando-se assim que uma estrutura grande de dados, como um mapa, seja mantida na memória por mais tempo do que o necessário. Esse cálculo é compatível ainda com a execução condicional dos *functores*, permitindo que operadores que executam um ou mais *functores/containers*, com base em um resultado intermediário, sejam facilmente definidos.

Os modelos criados pela combinação desses operadores são gravados na forma de scripts textuais. Scripts podem ser criados em duas sintaxes distintas, a saber: XML ou EGO Script. Internamente, Dinamica não faz distinção da sintaxe usada na definição do script. EGO é uma linguagem específica que favorece a criação de modelos textuais, já o uso de XML é adequado quando os modelos não são criados diretamente, mas por meio da interface gráfica. Essa linguagem possui a vantagem de permitir que modelos possam ser especificados parcialmente e gravados, adiando a conclusão do modelo para um segundo momento.

O núcleo do Dinamica EGO, responsável pela criação e execução dos modelos, é escrito em C++, enquanto a interface gráfica é escrita em Java. O núcleo pode ser usado diretamente para executar simulações, sem que haja sobrecarga da presença de uma interface gráfica, por meio de uma versão em linha de comando. Atualmente vários operadores do Dinamica são paralelizados internamente e tiram proveito de ambientes de execução com múltiplos processadores ou núcleos, dividindo a tarefa aplicada, por exemplo, a um determinado mapa entre múltiplos processadores. Futuramente, prevê-se a paralelização externa dos operadores, permitindo que vários operadores sejam executados de forma concorrente em múltiplas máquinas, sem que haja intervenção do usuário para indicar quais operações serão executadas em paralelo e quais serão executadas sequencialmente. A esse respeito, alguns dos modelos atuais implementados na plataforma do Dinamica EGO processam vários gigabytes de dados,

envolvendo uma longa cadeia de algoritmos de álgebra de mapas. Para fins de comparação, um modelo implementado no Dinamica EGO que calcula uma rota de menor custo sobre um mapa 900x900 células é sessenta vezes mais rápido do que o mesmo modelo implementado sobre o *MacroModeler* do Idrisi (Eastman, 2003). Outra grande vantagem é que, enquanto outros ambientes de modelagem em geoprocessamento (ex: *MacroModeler* do Idrisi e *ModelBuilder* do ArcGis) utilizam-se de módulos que escrevem seus resultados em disco para serem novamente lidos em um passo posterior, penalizando desse modo sua performance, Dinamica utiliza um fluxo contínuo de dados que são mantidos em memória somente enquanto forem necessários à execução do modelo.

3. SimAmazonia

O ambiente de simulação do SimAmazonia-1 incorpora dois modelos acoplados com distintas estruturas espaciais: 1) uma configuração em sub-regiões definida a partir de uma regionalização socioeconômica da Amazônia e 2) e um mapa *raster*. Um modelo integrador de cenários projeta as taxas de desmatamento para as 47 sub-regiões da bacia, processando dados de desmatamento (taxa anual e derivada média anual calculada para o quinquênio 1997-2002), de estradas a serem asfaltadas e das extensões de remanescentes florestais e áreas protegidas atuais e planejadas em cada uma das sub-regiões (Soares-Filho *et al.*, 2006). As taxas regionais produzidas por esse modelo implementado no software Vensim (Ventana, 2006) são então passadas para o modelo espacial do Dinamica, o qual utiliza mapas de infra-estrutura, unidades administrativas, áreas protegidas e aspectos biofísicos para reproduzir os padrões de progressão

do desmatamento através do território amazônico. Cada sub-região possui um modelo singular com parâmetros personalizados. Não obstante, a integridade espacial entre as sub-regiões é assegurada através do cômputo anual de um subconjunto de variáveis espaciais (ex. distâncias ao desmatamento prévio e às estradas vicinais) de modo contínuo por toda a bacia. Isso faz com que a dinâmica de uma região afete as de suas vizinhas. Também, conectado ao simulador de mudanças, encontra-se um modelo para simular a expansão da rede de estradas vicinais e assim incorporar o efeito da abertura de estradas espontâneas (Souza *et al.*, 2004) na difusão do desmatamento.

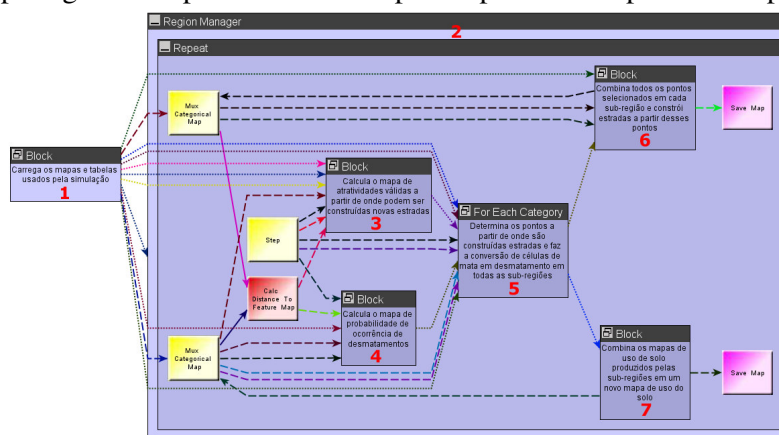


Fig. 2 – Visão geral do modelo

4. Implementação do modelo de desmatamento SimAmazonia na plataforma do Dinamica EGO

A seguir é mostrado de forma geral como o modelo de desmatamento SimAmazonia foi implementado na plataforma Dinamica EGO.

Existem basicamente dois sub-modelos principais no modelo de desmatamento SimAmazonia (Fig. 2). O primeiro, identificado como 1 (Fig. 2), é responsável pela carga das imagens (mapas) e tabelas de parâmetros. O segundo, identificado como 2 (Fig. 2), é responsável pela simulação propriamente dita.

Esse segundo sub-modelo é delimitado por um *container* “RegionManager” que é responsável pelo controle das sub-regiões. Dentro do “RegionManager”, existe um *container* “Repeat” responsável por executar os seus operadores por 50 iterações, correspondentes aos 50 anos abrangidos pelo intervalo da simulação. Uma vez que os limites das sub-regiões não sofrem alterações ao longo da execução, o uso do “RegionManager” externo ao “Repeat” evita que as fronteiras das sub-regiões tenham que ser recalculadas a cada iteração.

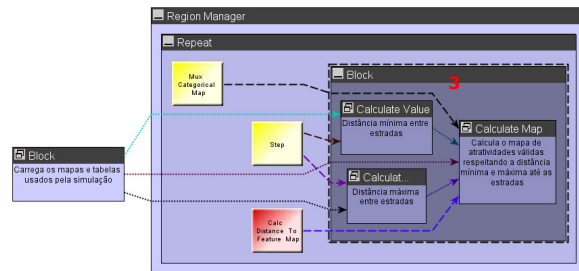


Fig. 3 – Candidatos a novas estradas

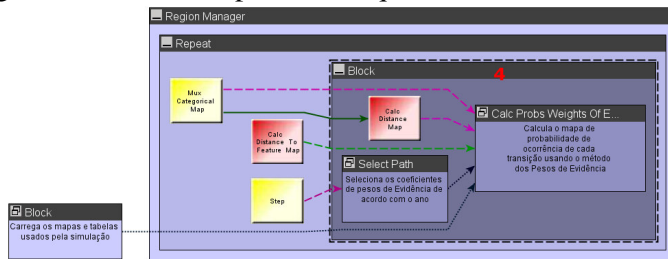


Fig. 4 – Cálculo de probabilidade de desmatamento

functores é responsável por re-injetar o mapa de uso de solo, o outro faz o mesmo com o mapa de estradas, ambos mapas dinâmicos.

Tanto o mapa de uso de solo quanto o de estradas são passados para um *functor* “CalcDistanceToFeatureMap”, usado para calcular o mapa de distâncias até as estradas principais e/ou vicinais mais próximas.

Em seguida são executados dois outros sub-modelos. O primeiro, identificado como 3 (Fig. 2 e 3), determina as áreas em um mapa de atratividade (favorabilidade à criação de estradas) candidatas a destinos de novas estradas. A Fig. 3 mostra o conteúdo desse sub-modelo e suas dependências. Como os critérios de rejeição de candidatos (distância máxima e mínima até à estrada prévia mais próxima, etc) variam de acordo com o ano corrente de uma forma pré-estabelecida, a iteração corrente da simulação capturada por um *functor* “Step” é usada para obter os dados dos parâmetros desse cálculo.

O segundo sub-modelo, identificado como 4 (Fig. 2 e 4), calcula o mapa de probabilidades de conversão de desmatamento em cada célula, usando o método de Pesos de Evidência (Soares-Filho, 2006). A Fig. 4 mostra o conteúdo desse sub-modelo e suas dependências.

Os resultados produzidos pelos sub-modelos 3 e 4 são usados como entradas para as análises específicas de cada sub-região. O *container* “ForEachCategory”, identificado como 5 (Fig. 2 e 5), executa os *functores/containers* contidos nele uma vez para cada sub-região definida no mapa de sub-regiões.

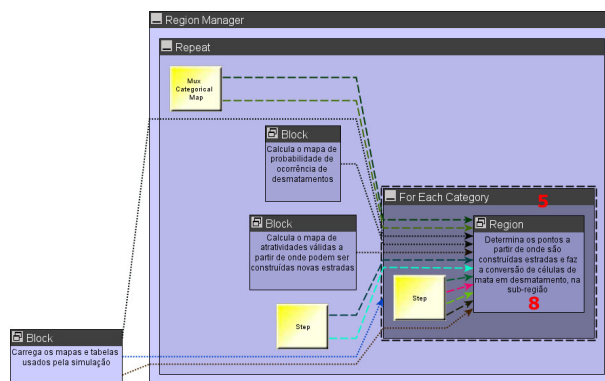


Fig. 5 – Análise de todas as sub-regiões

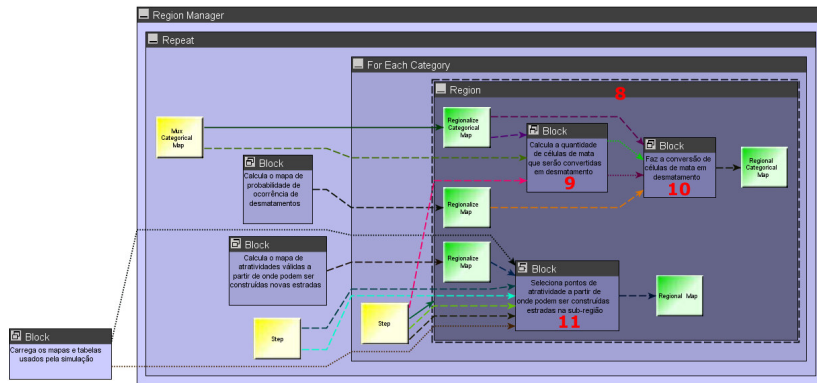


Fig. 6 - Análise individual de sub-região

Dentro do “ForEachCategory”, está definido um sub-modelo “Region”, identificado como 8 (Fig. 5 e 6). Ele consiste numa forma conveniente de se conectar o “RegionManager” externo com os *functores* internos, específicos de

manipulação dos dados das sub-regiões, sem que seja preciso tornar explícito um grande número de associações. A Fig. 6 mostra o conteúdo desse sub-modelo e suas dependências.

Os sub-modelos definidos no “Region” recuperam a taxa de desmatamento para cada sub-região calculada pelo modelo externo implementado no Vensim, fazem a conversão de células de mata em desmatado de acordo com a taxa calculada e com o mapa de probabilidade de desmatamento, e determinam as células de destino para a construção de novas estradas na sub-região dentre aquelas candidatas.

O sub-modelo de cálculo da taxa de desmatamento é indicado como 9 (Fig. 6 e 7). A Fig. 7 mostra o conteúdo desse sub-modelo e suas dependências. Ele usa o índice da sub-região e o *functor*

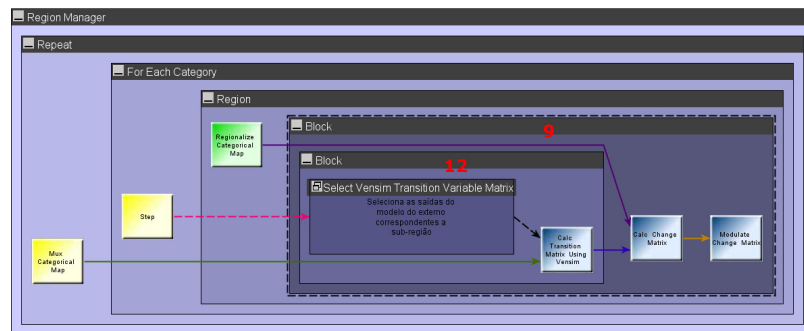


Fig. 7 – Cálculo da taxa de desmatamento usando o modelo externo do Vensim

“CalcTransitionMatrixUsingVensim” para perguntar ao modelo externo pela matriz com a taxa de desmatamento que deve ser usada. Essa matriz é passada para o “CalcChangeMatrix” que transforma a taxa em um número absoluto de mudanças (quantidade de células). A quantidade de células é então dividida entre as funções de transição (*Expander* e *Patcher*).

Em seguida, o sub-modelo indicado por 10 (Fig. 6 e 8), faz a conversão de desmatamento usando os *functores* das funções de transição “Expander”, responsável pela expansão de manchas prévias, e “Patcher”, responsável pela criação de novas manchas. A Fig. 8 mostra o conteúdo desse sub-modelo e suas

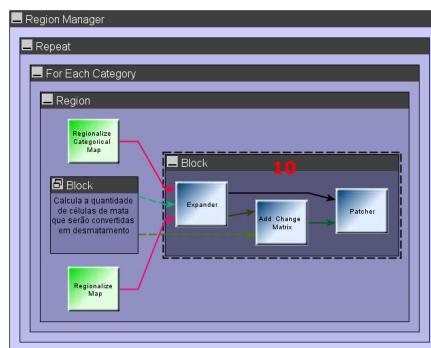


Fig 8 – Conversão de células de mata em desmatado

dependências.

A determinação dos destinos de novas estradas é feita pelo sub-modelo indicado por 11 (Fig. 6 e 9). A Fig. 9 mostra o conteúdo desse sub-modelo e suas dependências. Basicamente, o *functor* “CalcDestinationMap” divide o mapa de atratividade recortado por sub-região em um determinado número de quadrantes. Em seguida, ele escolhe uma célula de destino em cada quadrante. Os parâmetros de seleção desses pontos de destino (número de quadrantes

verticais e horizontais, distância mínima entre novos destinos, etc) variam de acordo com a sub-região e com a iteração da simulação de forma pré-determinada, sendo previamente estabelecidos e passados como dados armazenados em uma tabela.

Uma vez que as operações específicas das sub-regiões tenham sido executadas, é preciso que os mapas de uso de solo produzidos por cada sub-região sejam fundidos em um novo mapa de uso de solo e que novas estradas sejam construídas.

A combinação dos mapas de uso do solo é feita pelo sub-modelo identificado como 7 (Fig. 2 e 10). A Fig. 11 mostra o conteúdo desse sub-modelo e suas dependências. O mapa resultante é repassado para um *functor* “MuxCategoricalMap, o que permite que ele seja usado como mapa de uso de solo na iteração seguinte.

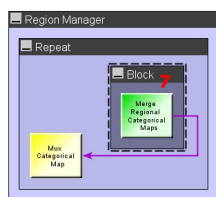


Fig. 10 – Fusão dos mapas de uso do solo

são construídas as novas estradas conectando cada destino escolhido à rede de estradas pré-existentes, usando o mapa de custo calculado para determinar o caminho de menor custo a ser seguido. Tal qual o mapa de uso de solo, esse mapa é repassado para um *functor* “Mux-CategoricalMap” para que ele seja usado como mapa de estradas na iteração seguinte. Para cada iteração, são produzidos mapas de desmatamento e da rede de estradas vicinais atualizada, como outros também que sejam especificados pelo usuário (ex. mapa de probabilidade de desmatamento).

5. Consideração final

Como resultado, o ambiente de modelagem do Dinamica EGO favorece simplicidade, flexibilidade e desempenho, otimizando velocidade e recursos computacionais, como memória e uso de processamento em paralelo. Dinamica EGO está disponível como *freeware* em www.csr.ufmg.br/dinamica/EGO. Além da sua aplicação nos nossos estudos de modelagem, é nosso objetivo disponibilizá-lo como um instrumento de investigação de dinâmica de sistemas ambientais para uso tanto de profissionais da área de planejamento, como no treinamento de estudantes em ciências ambientais.

6. Agradecimentos

Os autores agradecem ao Projeto LBA (Experimento em Larga Escala da Biosfera-Atmosfera da Amazônia) e Fundação Gordon & Betty Moore pelo financiamento. Segundo

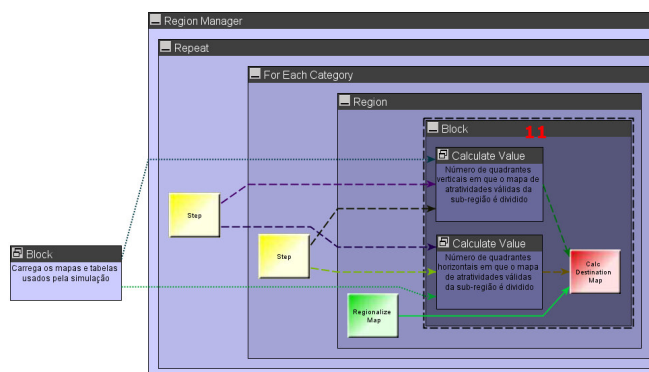


Fig. 9 – Sorteio de destinos para novas estradas

A criação de novas estradas é feita pelo sub-modelo identificado como 6 (Fig. 2 e 11). A Fig. 11 mostra o conteúdo desse sub-modelo e suas dependências. Após a combinação de todos os mapas de destinos de novas estradas produzidos para as sub-regiões, é calculado o mapa de custo pelo *functor* “CalcCostMap” a partir das estradas pré-existentes (vicinais e/ou principais) usando um mapa de fricção pré-definido. Por fim,

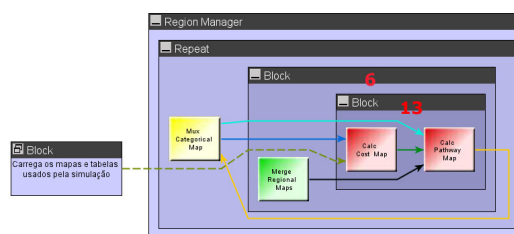


Fig. 11 – Construção de novas estradas

autor também recebe apoio do CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

Referências

- Abran, G.; Treinish, A. L.; Watson, T.J. An extended data-flow architecture for data analysis and visualization. In: **Proceedings of the 1996 IBM Visualization Data Explorer Symposium**. Disponível em <http://www.research.ibm.com/dx/proceedings/dx_paper/index.htm> Acesso em 3 set. 2006.
- Almeida, C.; Batty, M.; Monteiro, M.; Camara, G.; Soares-Filho, B.; Cerqueira, G.; Pennachin, C. L. Stochastic cellular automata modeling of urban land use dynamics: empirical development and estimation. **Computers, Environment and Urban Systems**, 27, pp. 481–509, 2003.
- Castro, F. V. F.; Soares-Filho, B.S.; Mendoza, E. Modelagem de cenários de mudanças na região de Brasília aplicada ao zoneamento ecológico-econômico do Estado do Acre. In: Simpósio Brasileiro de Sensoriamento Remoto (SBSR), 13, 2007, Santa Catarina. (submetido).
- Costanza, R.; H.C. Fitz; J.A. Bartholomew, and E. DeBellevue. **The Everglades Landscape Model : Summary Report of Task 1, Model Feasibility Assessment. Report to the Everglades Systems Research Division**, South Florida Water Management District, 3301 Gun Club Road, West Palm Beach, FL 33416.1992. 118 p.
- Eastman, J. R. **Idrisi Kilimanjaro**. Clark University, Massachusetts, USA 2003 200p.
- Eastman, R. The Evolution of Modeling Tools in GIS. **Directions Magazine** 2001. Disponível em <http://www.directionsmag.com/article.php?article_id=114&trv=1> Acesso em 3 set. 2006.
- Godoy, M. M.; Soares-Filho, B.S. Modelagem de dinâmica intra-urbana no Bairro Savasssi, Belo Horizonte, Brasil. In: Almeida C. M., Câmara G. N., Vieira, A. M. **Geoinformação em Estudos Urbanos**. INPE, no prelo.
- Hirsch, A.I.; Little, W.S.; Houghton, R.A.; Scott, N.A.; White, J.D. The net carbon flux due to deforestation and forest re-growth in the Brazilian Amazon: analysis using a process-based model. **Global Change Biology**, v. 10, n. 5, p. 908-924. 2004.
- Johnston, W. M.; Hanna, J.R.P; Millar, R. Advances in Dataflow Programming Languages. **ACM Computing Surveys**, v. 36, n. 1, p. 1-34, 2004.
- Soares Filho, B.S.; Pennachin, C. L.; Cerqueira, G. DINAMICA – a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. **Ecological Modelling**, v. 154, n. 3, p. 217-235, 2002.
- Soares-Filho, B.S.; Alencar, A.; Nepstad D.; Cerqueira, G.; Vera-Diaz, M.; Rivero, S.; Solórzano, L.; Voll, E.. 2004. Simulating the Response of Land-Cover Changes to Road Paving and Governance Along a Major Amazon Highway: The Santarém-Cuiabá Corridor. **Global Change Biology**. v. 10, n. 5, p. 745-764, 2004.
- Soares Filho, B.S. 2006. **Dinamica project**. Disponível em <<http://www.csr.ufmg.br/dinamica>> Acesso em 3 set. 2006.
- Soares-Filho, B. S.; Nepstad, D; Curran, L.;Voll, E.; Cerqueira, G.; Garcia, R. A.; Ramos, C. A.; McDonald, A; Lefebvre, P., Schlesinger, P. Modeling conservation in the Amazon basin. **Nature**, v. 440, p. 520-523, 2006.
- Souza, C.; Brandão, A.; Anderson A.; Veríssimo, A. **Avanço das Estradas Endógenas na Amazônia** Disponível <<http://www.imazon.org.br/upload/124466.pdf>> Acesso em 07 jul. 2004.
- Tomlin, C. D. **Geographic Information Systems and Cartographic Modelling**. Prentice-Hall, Inc. New Jersey, 1990. 249 p.
- Ventana. Vensim software – linking systems thinking to powerful dynamic models. <<http://www.vensim.com/software.html>> Acesso em 3 set. 2006.
- Verburg, P.H.; Soepboer, W.; Veldkamp, A.; Limpiada, R.; Espaldon, V.; Mastura, S.S. Modeling the spatial dynamics of regional land use: the CLUE-S model. **Environmental Management**, v. 30, p. 391–405. 2002.
- White, R.; Engelen, G.; Uljee, I. Modelling Land Use Change with Linked Cellular Automata and Socio-Economic Models: A Tool for Exploring the Impact of Climate Change on the Island of St. Lucia. In: Hill, M., Aspinnall, R (Eds.), **Spatial Information for Land Use Management**. Gordon and Breach, pp. 189-204. 2000.
- Wesley, M. J.; Hanna, J.R.P; Millar, R.J. Advances in dataflow programming languages. **ACM Computing Surveys**, v. 36, p.1-34, 2004.