

# LINGUAGEM ESPECÍFICA DE DOMÍNIO PARA SIMULAÇÃO ESPACIAL

Luís de Sousa<sup>1</sup>; Alberto Rodrigues da Silva<sup>1</sup>

<sup>1</sup>Instituto Superior Técnico – Av. Rovisco Pais 1049-001 Lisboa

(luis.a.de.sousa@gmail.com; alberto.silva@acm.org)

---

## RESUMO

Este artigo descreve uma linguagem específica de domínio orientada para modelos de Simulação Espacial na área dos Sistemas de Informação Geográfica (SIG), designada DSL3S. Técnicas como os autômatos celulares e os modelos de agentes têm sido usadas na área para capturar e simular a dinâmica da mudança da informação espacial. Invariavelmente, as ferramentas existentes para o efeito requerem conhecimentos de programação o que tem limitado a sua utilização por parte de utilizadores de SIG. A DSL3S pretende sintetizar os conceitos relevantes de uma simulação espacial no contexto dos SIG através de um perfil UML. Este perfil será suportado na plataforma ProjectIT-Studio (uma ferramenta de engenharia orientada por modelos) para, através de mecanismos de geração de código, se traduzir modelos UML em simulações quase prontas a executar em diferentes plataformas computacionais.

---

## 1. Introdução

Nos sistemas de informação é recorrente existir uma dependência dos dados em relação ao tempo, sendo estes usualmente válidos apenas durante determinados períodos. Em tais casos é comum a compilação de um histórico das variáveis armazenadas. Em certas aplicações o conhecimento de como as variáveis mudam pode não ser suficiente. De forma a planejar o futuro pode ser necessário conhecer também porque razão mudam, a forma como mudaram e eventualmente antever como evoluirão (Samuelson & Macal, 2006). Na área dos Sistemas de Informação Geográfica (SIG) esta necessidade tem sido satisfeita através de processos de Modelação e Simulação no contexto espacial. A técnica mais antiga para este fim recorre a Autômatos Celulares (AC) (Wuensche & Lesser, 1992) nos quais o mundo é discretizado numa grelha de células de igual dimensão que evolui de acordo com um conjunto fixo de regras. Também Simulações baseadas em Agentes (Weiss, 1999) têm adquirido popularidade e sido aplicadas com sucesso na área dos SIG, com especial relevância nas seguintes aplicações:

- *Planeamento Urbano* – estudo previsão de mudanças no tecido urbano para alocar novas infraestruturas (Batty, 2007);
- *Uso do Solo* – compreensão da dinâmica do uso solo, como as transições entre coberto vegetal, agricultura e urbe (Messina & Walsh, 2000);
- *Fogo/Florestas* – compreensão do crescimento das Florestas e antecipação da propagação do fogo (Li & Magill, 2001);

- *Biologia* – modelação a evolução de habitats e estudo da dinâmica de populações (Ermentrout & Edelstein-Keshet, 1993).

Dada a natureza multidimensional da informação espacial e a sua origem ecléctica, os modelos de simulação espacial desenvolvidos no contexto dos SIG tendem a ser muito específicos à aplicação a que se destinam. Por esta razão, os pacotes de SIG continuam em grande medida a não apresentar ferramentas dedicadas a este tipo de técnicas, o que resulta no desenvolvimento *ad-hoc* de modelos. Por estas razões, a utilização de ferramentas de Simulação Espacial tem permanecido dependente de programadores experientes, largamente inacessível aos peritos no campo de aplicação.

Este artigo descreve o projecto DSL3S, um esforço para aproximar a Simulação Espacial dos utilizadores finais de SIG através de uma linguagem específica de domínio e independente de plataformas. Pretende-se o desenvolvimento de modelos de simulação espacial dispensando programação, através de uma abordagem mais alto nível, orientada por modelos (Saraiva & da Silva, 2008). A Secção 2 revê um conjunto de ferramentas de simulação espacial no contexto dos SIG. A Secção 3 introduz o perfil UML que captura os conceitos fundamentais da DSL3S. A Secção 4 apresenta uma aplicação exemplificativa da linguagem ao caso clássico da simulação de fogo florestal. A Secção 5 sumariza o artigo e discute o trabalho futuro.

## **2. Ferramentas de Simulação Espacial**

Usando uma linguagem de programação generalista (e.g. C, C++, Java) para desenvolver uma simulação espacial, para além da implementação lógica, o programador tem que controlar o fluxo de execução, gerir recursos do sistema e manipular estruturas de dados. Estas tarefas adicionais podem levar a diversos problemas, nomeadamente (Fall & Fall, 2001): (i) dificuldade em verificar a correcta implementação do modelo; (ii) dificuldade em modificar a implementação, limitando a sua generalização; (iii) modelos diferentes são comparáveis apenas pelas suas entradas e saídas; e (iv) integração difícil com outros modelos ou ferramentas. Para além destas linguagens existe hoje espectro de ferramentas de simulação que vão desde as que dão suporte directo via programação àquelas baseadas em modelos parametrizáveis pré-construídos.

### **2.1 Ferramentas de Nível-Programação**

Estas ferramentas estendem as funcionalidades das linguagens de programação generalistas, tipicamente fornecendo bibliotecas de código dedicadas à simulação. Estas soluções reduzem substancialmente o tempo de desenvolvimento e aumentam a fiabilidade dos modelos. São tipicamente apoiada em linguagens de programação de alto nível orientadas por objectos, onde o programador especifica a estrutura e comportamento de objectos pré-definidos. Exemplos destas ferramentas são Swarm (Mina et al., 1996), RePAST (Samuelson & Macal, 2006) e MASON (Luke et al., 2005).

**Swarm** foi a primeira destas ferramentas, proporcionando um conjunto de primitivas para criar simulações espaciais e funções para a visualização de resultados. Ganhou popularidade, mas a integração com os SIG é ainda fraca, resumindo-se a dados *raster*.

**RePAST** é uma biblioteca Java mais recente, que evoluiu para um pacote eclético suportando diferentes técnicas de simulação que vão muito para além da simulação espacial. O que lhe granjeou grande popularidade. Hoje será possivelmente a mais útil destas ferramentas e aquela com a melhor integração com os SIG.

**MASON** é também uma biblioteca Java para modelação com agentes, mas com o objectivo de ser leve, rápida e portátil. É uma ferramenta moderna e muito compacta, com menos funcionalidades mas já abrangendo dados espaciais vectoriais e *raster*.

## 2.2 Ferramentas de Nível-Modelo

Neste grupo estão as ferramentas que possibilitam a utilização de simulações espaciais sem recorrer à programação. São altamente específicas, não indo muito além de um modelo pré-codificado ao qual utilizador pode ajustar um conjunto de entradas e de parâmetros. Permitem obter resultados rapidamente mas invariavelmente dependem de plataformas específicas quer na execução quer na integração com os SIG. Exemplos são TELSA (Merzenich & Frid, 2005), LANDIS (Mladenoff, 2004) e SLEUTH (Yi & He, 2009).

**TELSA** é um programa especializado em ecossistemas, a típica ferramenta comercial de simulação espacial que permite o estudo de diferentes cenários de gestão. É parametrizável através de uma linguagem diagramática, desenvolvida pelo vendedor, a ESSA.

**LANDIS** é um projecto conjunto dos serviços florestais dos EUA e diversas universidades desse país para a simulação do coberto florestal à larga escala. O utilizador providencia um conjunto de variáveis de entrada para as quais o comportamento está pré-definido.

**SLEUTH** é um dos mais antigos modelos do género, datando do início da década de 1990. O seu domínio é o desenvolvimento urbano, com apenas seis temas espaciais de entrada para os quais um conjunto de comportamentos pode ser ajustado. Tornou-se numa solução popular no domínio, aplicado com sucesso a diferentes partes do mundo.

## 2.3 Linguagens Específicas de Domínio

A meio caminho entre as ferramentas do Nível-Modelo e as de Nível-Programa, apresentam funcionalidades ao nível do modelo mas dirigidas a classes de aplicações. Tomam menos assumpções estruturais, permitindo o desenvolvimento de comportamentos *ad hoc*. Exemplos de linguagens que implementam o conceito são SELES (Fall & Fall, 2001), MOBIDYC (Ginot *et al.*, 2002) e NetLogo (Sakellariou *et al.*, 2008).

**SELES** é uma linguagem declarativa para a modelação de dinâmica da paisagem, tentando balancear a flexibilidade da programação com a facilidade de utilização de modelos pré-definidos, garantindo interoperabilidade com dados espaciais *raster*.

**MOBIDYC** é uma linguagem de programação dedicada à dinâmica de populações que permite a construção de modelos complexos a partir de primitivas simples, próximas da

linguagem natural. É na realidade uma biblioteca de código na linguagem orientada por objectos Smalltalk. Não permite a integração a integração directa de informação espacial.

**NetLogo** é a última de uma geração de linguagens que evoluiu da linguagem funcional Logo, especializada em simulações com Agentes. De código fechado, tem sido usada sobretudo como ferramenta de ensino. Permite a utilização de dados vectoriais e *raster*.

### 2.3 Dificuldades

As ferramentas do Nível-Programa aliviam algum do esforço imposto pelas linguagens de programação generalistas mas requerem ainda assim boas competências de programação (Tobias & Hofmann, 2004). O total conhecimento de uma destas bibliotecas é algo atingível apenas com vários meses de prática (Samuelson & Macal, 2006). Tipicamente de código aberto, usualmente alcançam diferentes plataformas computacionais. Congregam largas comunidades de utilizadores que prestam apoio informal mas extenso.

No Nível-Modelo estão ferramentas muito específicas, cujo comportamento e implementação são total ou parcialmente opacos ao utilizador, não sendo modificáveis para outros domínios aplicativos. O utilizador pode de facto dispensar competências de programação mas fica restrito na funcionalidade da ferramenta. A evolução e generalização pode por vezes ser tão cara que as condena à extinção. Tradicionalmente tiram partido de nichos de mercado, satisfazendo um conjunto de utilizadores específicos, daí a natural comercial de muitas destas ferramentas. O suporte, a existir, é comercial.

As linguagens específicas facilitam a construção de modelos e reduzem o tempo de desenvolvimento, mas não dispensam competências de programação. Tal como para linguagens generalistas, é necessário conhecer palavras chave e saber compôr um conjunto de instruções num programa. Linguagens deste tipo têm sido utilizadas especialmente para fim académicos. As comunidades de utilizadores tendem a ser fortes, mas a dependência de plataformas computacionais particulares é muitas vezes um problema.

Relativamente ao desenvolvimento gráfico, as ferramentas do Nível-Programa parecem liderar, ao apoiar-se em ambientes de programação modernos, por seu lado as linguagens específicas têm tomado sobretudo um formato declarativo. Em contraste, ferramentas para a simulação de dinâmica de sistemas sem componente espacial, como Stella ou Modelica (Cellier, 2008) há muito que apresentam linguagens gráficas de desenvolvimento.

### 3. A linguagem DSL3S

A especificação do UML 2.0 permite a extensão das suas primitivas base (elementos gráficos, ligações, etc) por especialização para domínios particulares (OMG, 2010). Tal é conseguido pela definição de um Perfil UML, uma coleção de estereótipos, etiquetas e restrições. Estereótipos são especializações de elementos existentes do UML, criando novos elementos de abstracção mais estrita. Um conjunto semanticamente relacionado de estereótipos, especificado e enriquecido com anotações e restrições, pode ser usado para especializar o UML numa nova linguagem dedicada a determinado domínio.

A DSL3S (*Domain Specific Language for Spatial Simulation Scenarios*) é um perfil UML que define um conjunto de estereótipos que capturam as noções abstractas subjacentes à simulação espacial. Estes estereótipos refletem os termos usados na explicação de uma simulação usando o discurso natural, tal como descrevendo o fogo como um agente ou a topografia como uma variável espacial.

### 3.1 O Meta-modelo

Foram identificados três elementos principais como os conceitos fundamentais de uma simulação espacial: Variável Espacial, Animata e Comportamento. **Variáveis Espaciais** são temas espaciais que produzem algum tipo de impacto na dinâmica da simulação, e.g. topografia que impede o desenvolvimento urbano ou a biomassa que alimenta um fogo. **Animata** é um termo proposto por Wilson (1991) significando *animal artificial*, aqui empregue de forma mais lata, representado todos os agentes da mudança como um fogo, áreas urbanas ou espécies num modelo predador-presa. **Comportamento** associa os dois conceitos anteriores, definindo como reagem os animatas à sua envolvente e estado interno. Para além destes conceitos fundamentais outros elementos podem fazer parte de uma simulação, particularmente variáveis de contexto. Poderão existir variáveis **Globais**, representando informação que é constante em todo o espaço da simulação, tal como a direcção do vento. Poderá também o **Estado** do animata que pode ser encarado como um conjunto de variáveis que descreve cada instância a dado momento do tempo.

A Figura 1 apresenta o modelo conceptual destes conceitos: uma **Simulação** é composta por variáveis Espaciais, Animatas e variáveis Globais. Os Animatas podem ser compostos por variáveis de Estado e ainda possuir Comportamento que determina a sua evolução.

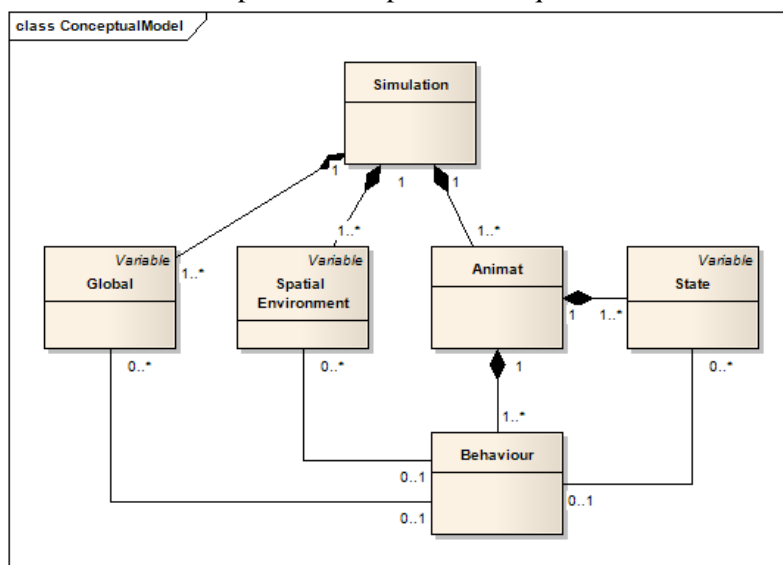


Figura 1 – Modelo Conceptual da DSL3S

O estereótipo Simulação é usado para capturar definições como o número de iterações a executar, o espaço cartográfico da simulação e o tipo de vizinhança a considerar. O estereótipo Global é um simples escalar que pode ser modificado estocasticamente a cada iteração ou ser associado ao modelo como uma série temporal. Variável Espacial é essencialmente um veículo de ligação de temas geo-referenciados ao modelo; não há qualquer alusão explícita a sistema de coordenadas, com a simulação a decorrer efectivamente no espaço cartesiano. Uma variável espacial também poderá ser gerada estocasticamente para a criação de cenários sintéticos. Para além do seu estado interno cada animata tem também associada uma posição espacial. O número, estado interno e posição inicial de cada animata podem também ser inicializados através de temas geo-referenciados alimentados à simulação.

### 3.2 Comportamentos

Os elementos apresentados até aqui representam apenas a informação necessária à execução de uma simulação espacial. Mais será necessário para explicitar a forma como actuam os agentes, para tal a DSL3S apresenta cinco especializações de Comportamento:

- **Iniciar** – Captura as condições sob as quais um novo Animata aparece na simulação; o acto de “nascer”. E.g. uma simulação de desenvolvimento urbano em que novos centros urbanos podem emergir em locais que obedeçam a certos critérios. Etiquetas desta classe permitem definir limiares na vizinhança relativos a variáveis espaciais ou outros Animatas, acima dos quais a emergência de um novo Animata se torna possível.
- **Mover** – Relaciona o Animata com variáveis espaciais ou outros Animatas, determinando os locais onde será mais ou menos favorável estar. As etiquetas deste estereótipo permitirão dar pesos a cada uma destas relações. Numa simulação predador-presa o movimento de uma “ovelha” poderá ser pesado positivamente numa relação com a classe “erva” e negativamente numa relação com a classe “lobo”. Uma relação particular com o estado do Animata permite definir o impacto interno do movimento, tal como o consumo de energia.
- **Replicar** – Captura comportamentos em que um Animata se replica para uma localização adjacente, tal como um fogo que se propaga ou uma área urbana que se estende. Novamente o objectivo é definir as condições sob as quais tal poderá ocorrer, usando as etiquetas para pesar a influência de variáveis espaciais (e.g. alastrar de fogo) ou determinar limiares relativos ao estado interno (e.g. reprodução biológica). Também é possível definir com etiquetas o impacto da replicação no estado do Animata.
- **Colher** – Um comportamento que permite a um Animata modificar outros elementos na mesma localização. Poderá afectar uma variável espacial, como um fogo que consome biomassa, ou capturando outro Animata como num modelo predador-presa. As etiquetas desta classe permitem especificar os impactos no colectador e no colhido, ou ainda impor uma taxa de colheita.

- **Perecer** – Define as circunstâncias sob as quais um Animata deixa de existir durante a simulação (e.g. ficar sem energia). Na maioria dos casos são definidos limiares mínimos relativos ao estado interno do Animata, as condições para a perduração da sua existência. Relacionando a mesma classe de Animata, este comportamento pode definir condições em que o Animata perece por pressão social.

Outros tipos de comportamento são possíveis, em especial em áreas como a Economia ou a Sociologia (Epstein & Axtell, 1996), mas que, apesar de interessantes não são comuns em aplicações de SIG.

### 3.2 Vistas e Ícones Propostos

Os modelos construídos com a DSL3S podem tornar-se visualmente complexos se representados num único diagrama. De forma a mitigar tais dificuldades é proposto com a DSL3S um conjunto de vistas que organiza o modelo de forma a isolar cada aspecto num diagrama específico; nomeadamente: a **Vista Simulação** é um simples diagrama de pacotes que agrega as restantes vistas. Desta vista partem três outras: Vista Global, Vista Espacial e Vista Animata. A **Vista Global** congrega os elementos que não têm representação espacial directa, tais como as variáveis globais. Na **Vista Espacial** são representadas as entradas geográficas do modelo: as variáveis espaciais. Todas as classes destas duas vistas devem ser ligadas a uma classe simulação que é única no modelo. A Figura 2 resume este esquema.

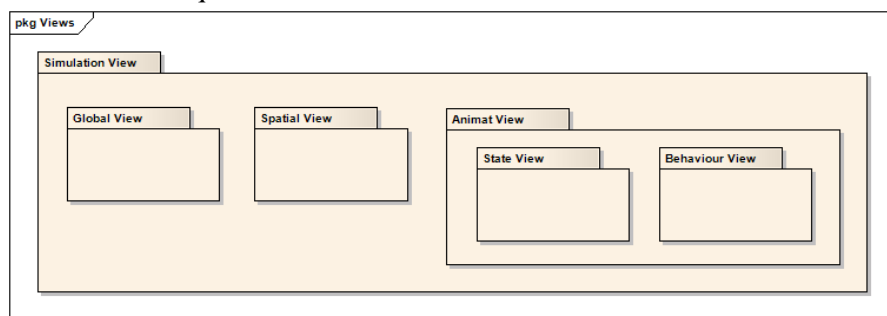


Figura 2 – Esquema de vistas proposto

A **Vista Animata** é também um diagrama de pacotes, incluindo vistas do tipo Estado e Comportamento, que caracterizam cada classe de Animata em particular. Na **Vista Estado** são definidas as propriedades que compõe um Animata, na **Vista Comportamento** são definidas as classes comportamento e respectivas relações, tendencialmente existindo uma destas vistas para cada comportamento diferente do Animata.

Para além destas vistas é também proposto um conjunto de ícones para tornar a linguagem mais explícita, conforme sugerido na Figura 3. Para cada estereótipo foi criada uma imagem que de forma pictórica pretende capturar a semântica base de uma simulação.

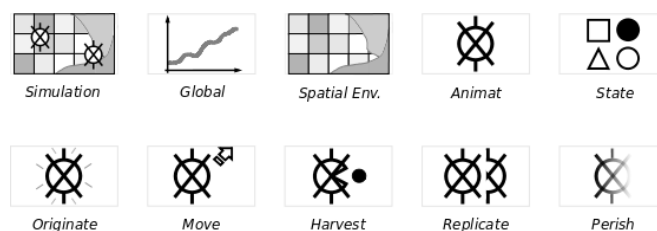


Figura 3 – Ícones propostos

#### 4. Exemplo de Aplicação

A DSL3S contém os conceitos suficientes para descrever uma simulação tradicional num contexto SIG. A título de exemplo de aplicação da DSL3S refere-se o caso clássico da propagação de fogos florestais. Li & Magill (2001) publicaram um dos trabalhos de referência na área, cujo modelo pode ser resumido em três comportamentos (i) propagação do fogo a localizações adjacentes; (ii) consumo de biomassa em locais onde o fogo lavra; e (iii) extinção do fogo em locais onde a biomassa foi totalmente consumida.

Os primeiros elementos a identificar são as variáveis da simulação. Existe apenas uma variável global, o Vento, um vector com direcção e intensidade. Quanto às variáveis espaciais existem três: Biomassa, Inclinação e Humidade. Animatas há só um, o Fogo, que representa os locais onde a biomassa é consumida. A Figura 4 apresenta estes elementos.

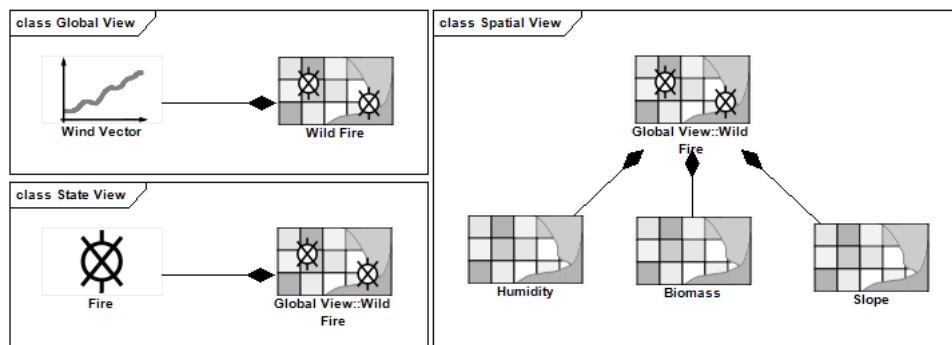


Figura 4 – Vista Global, Vista Estado e Vista Espacial do modelo Li & Magill (2001)

A queima da biomassa é modelada por um comportamento do tipo Colher que parametriza a taxa de consumo. A extinção do fogo é modelada por um comportamento Perecer, determinando um limiar de biomassa abaixo do qual o fogo se apaga. Finalmente a propagação do fogo a localizações adjacentes é modelada por um comportamento Replicar que é influenciado pela Humidade, Inclinação, Biomassa e Vento. Este é um modelo típico de dinâmica espacial em SIG, em que diversos factores afectam em paralelo as acções de um Animata; o objectivo é conhecer o peso de cada um desses factores no processo. A Figura 5 apresenta os diagramas de comportamento com o léxico da DSL3S.



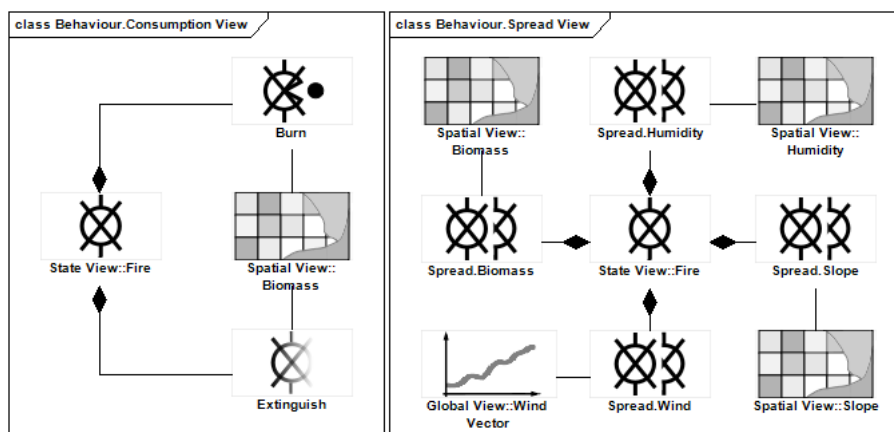


Figura 5 – Vistas de Comportamento do modelo Li & Magill (2001)

## 5. Sumário e Trabalho Futuro

A Simulação Espacial é uma importante técnica no domínio dos SIG, dotando estes sistemas de capacidades preditivas. Existe actualmente um grande número de ferramentas dedicadas ao desenvolvimento de modelos de simulação descrevendo um espectro que vai das ferramentas que requerem maiores competências mas de grande abrangência até aos modelos pré-codificados, mais simples de usar mas que limitam o domínio de aplicação. É também identificada uma tendência no que diz respeito à interoperabilidade com os SIG, com as ferramentas mais específicas a apresentarem maiores dificuldades nesse âmbito.

É neste contexto que surge a DSL3S, com o intuito de possibilitar o desenvolvimento e parametrização de modelos a partir de elementos gráficos, mais fáceis de definir e de entender por peritos do domínio, não técnicos de informática. O perfil UML, da linguagem DSL3S, apresentado neste artigo, captura de forma abstracta os conceitos básicos da dinâmica espacial. Estes conceitos são suficientes para descrever modelos tradicionais como a propagação do fogo.

O próximo passo da investigação será criar a infra-estrutura que permita traduzir modelos descritos pela DSL3S em simulações prontas, ou quase-prontas, a executar. Para este fim está em estudo a plataforma ProjectIT (Saraiva & da Silva, 2008), um ambiente de desenvolvimento que pretende suportar o processo de produção de software sob o paradigma da Engenharia Orientada por Modelos (MDE – *Model Driven Engineering*). Dada a sua capacidade para desenvolver e aplicar perfis UML e gerar código fonte automaticamente a partir de diagramas, esta plataforma apresenta-se como um ambiente apropriado para acolher a DSL3S. Os modelos finais serão codificados usando uma ferramenta do Nível-Programação como MASON ou RePAST.

## Referências Bibliográficas

- Batty, M. (2007). "Cities and Complexity". MIT Press.
- Cellier, F. (2008). "World3 in modelica: Creating system dynamics models in the Modelica framework". 6th International Modelica Conference.
- Epstein, J. M. & Axtell, R. (1996). "Growing Artificial Societies". MIT Press.
- Ermentrout, G. B. & Edelstein-Keshet, L. (1993). "Cellular automata approaches to biological modeling". *Journal of Theoretical Biology* 160, 97–133.
- Fall, A. & Fall, J. (2001). "A domain-specific language for models of landscape dynamics". *Ecological Modelling* 141, 1–18.
- Ginot, V., Le Page, C. & Souissi, S. (2002). "A multi-agents architecture to enhance end-user individual based modelling". *Ecological Modelling* 157, 23–41.
- Li, X. & Magill, W. (2001). "Modeling fire spread under environmental influence using a cellular automaton approach". *Complexity International* 8, 1–14.
- Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). "MASON: A Multi-Agent Simulation Environment". *Simulation*. 82(7):517-527.
- Merzenich, J. & Frid, L. (2005). "Projecting landscape conditions in southern Utah using VDDT". *Systems Analysis in Forest Resources: Proc. of the 2003 Symp. Dep. of Agriculture, Forest Service*.
- Messina, J. P. & Walsh, S. J. (2000). "The application of a cellular automaton model for predicting deforestation". In: *Proceedings of the 4th International Conference on Integrating GIS and Environmental Modelling: Problems, Prospects and Research Needs*. Banff, Canada.
- Minar, N., Burkhart, R., Langton, C. & Askenazi, M. (1996). "The Swarm simulation system: a toolkit for building multi-agent simulations". <http://www.santafe.edu/projects/swarm>.
- Mladenoff, D. (2004). "Landis and forest landscape models". *Ecological Modelling* 180(1), 7 – 19.
- OMG (2010). "UML 2.0 specification". <http://www.omg.org/spec/UML/2.0/>.
- Sakellariou, I., Kefalas, P., Stamatopoulou, I. (2008). "Teaching Intelligent Agents using NetLogo". *ACM-IFIP Informatics Education Europe III Conference, IEEEIII 2008, Italy, December 4-5, 2008*.
- Samuelson, D. & Macal, C. (2006). "Agent-based simulation comes of age". *OR/MS Today* 33(4).
- Saraiva, J. & da Silva, A. R. (2008). "The projectit-studio umlmodeler: A tool for the design and transformation of UML models". In: *Actas 3º CISTI*. Espanha.
- Tobias, R. & Hofmann, C. (2004). "Evaluation of free java-libraries for social-scientific agent based simulation". *Journal of Artificial Societies and Social Simulation* 7(1).
- Weiss, G. (ed.) (1999). "Multiagents Systems: a Modern Approach to Distributed Artificial Intelligence". MIT Press.
- Wuensche, A. & Lesser, M. (1992). "The Global Dynamics of Cellular Automata". Addison-Wesley.
- Yi, W. & He, B. (2009). "Applying SLEUTH for simulating urban expansion of Beijing". In: *2009 International Joint Conference on Artificial Intelligence*, vol. 2.