

Desenvolvimento de uma Plataforma Gráfica para a Descrição de Modelos de Sistemas Ambientais

Tiago F. M. Lima¹, Tiago G. S. Carneiro², Sérgio D. Faria³

¹Programa de Pós-Graduação em Análise e Modelagem de Sistemas Ambientais Instituto de Geociências – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG – Brasil

²Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
Ouro Preto, MG – Brasil

³Instituto de Geociências – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG – Brasil

tiagofml@yahoo.com.br, tiago@iceb.ufop.br, fariamaracai@yahoo.com.br

Abstract. *This paper presents the actual stage of development of TerraME Graphical Interface for Modeling and Simulation – TerraME GIMS, a graphical interface to building dynamic spatial-temporal models to the platform TerraME. The development of models through graphical components improve the productivity of the users and become more intuitive the process of building models, contributing to expand the user community of the environment TerraME.*

Resumo. *Este trabalho apresenta o estágio atual de desenvolvimento do TerraME Graphical Interface for Modeling and Simulation – TerraME GIMS, uma plataforma gráfica para a construção de modelos dinâmicos espaço-temporais para o ambiente de modelagem e simulação TerraME. A construção de modelos através de componentes gráficos além de aumentar a produtividade dos usuários torna o processo de construção de modelos mais intuitivo, contribuindo para ampliar a comunidade de usuários do ambiente TerraME.*

1. Introdução

Muitos problemas de natureza complexa exigem um tratamento interdisciplinar, envolvendo pessoas das mais diversas áreas do conhecimento. A criação de um modelo para representar o processo de mudança de uso e cobertura do solo na região Amazônica poderia envolver biólogos, sociólogos, economistas, geógrafos, geólogos, tecnólogos etc. A representação de um modelo desta natureza deve ser de comum entendimento para todas as pessoas envolvidas, sendo específica o suficiente para facilitar a representação do conhecimento que cada um tem a respeito do sistema e geral o suficiente para não restringir essa representação a um domínio de aplicação específico. A escolha da linguagem a ser utilizada na representação de um modelo é, portanto, fator crucial para o sucesso de um projeto de modelagem que envolva a colaboração de profissionais de diferentes formações.

O *TerraME* (Carneiro 2003) é um ambiente de modelagem e simulação de processos ambientais com representação explícita no espaço que permite aos usuários criar as estruturas, dados e regras, que irão definir o comportamento de um modelo, utilizando a linguagem de programação *TerraME*. Apesar de sua fácil utilização se comparada a linguagens de programação como Java e C++, a assimilação de seus conceitos ainda apresenta acentuada curva de aprendizado para profissionais e pesquisadores não familiarizados com algoritmos e técnicas de programação, conforme constatado em avaliações realizadas por participantes de cursos e apresentações sobre o ambiente *TerraME* (Câmara et al. 2007, Câmara et al. 2008). Entretanto, são esses os profissionais mais interessados no desenvolvimento de modelos para sistemas ambientais e que detêm o conhecimento a respeito do domínio de aplicação dos modelos. Desta maneira, o uso direto e obrigatório de uma linguagem de programação para representação de modelos dinâmicos se apresenta como a principal barreira para a difusão do ambiente *TerraME* e para a ampliação de sua comunidade de usuários.

Para permitir que os usuários do ambiente *TerraME* se concentrem na resolução dos problemas pertencentes ao domínio de aplicação dos modelos e não nos problemas da sua representação computacional, um novo e mais alto nível de abstração é necessário. A representação dos modelos através de componentes gráficos, como diagramas, ao invés de algoritmos, além de aumentar a produtividade dos atuais usuários do *TerraME*, diminuirá a curva de aprendizado de novos usuários.

Este trabalho apresenta o atual estágio de desenvolvimento do *TerraME Graphical Interface for Modeling and Simulation – TerraME GIMS*, uma interface gráfica com o usuário (GUI – *Graphical User Interface*) que permite a descrição e simulação de modelos de sistemas ambientais para a plataforma *TerraME* por meio de metáforas visuais que representam graficamente o modelo.

2. TerraME

O *TerraME* (*TerraLib Modelling Environment*), desenvolvido por Carneiro (2003), é um componente da família de soluções *TerraLib* (Câmara et al. 2000) para a implementação e simulação de modelos ambientais que envolvam a representação explícita do espaço. O *TerraME* provê mecanismos para a representação e simulação de modelos espaciais dinâmicos integrados a um Sistema de Informações Geográficas (SIG). Este ambiente permite a criação de modelos de múltiplas escalas espaciais e temporais.

O *TerraME* foi construído baseado na arquitetura em camadas, onde as camadas inferiores fornecem funcionalidades sobre as quais as camadas superiores são implementadas. Os componentes de sua arquitetura permitem aos usuários experientes a implementação de modelos utilizando diretamente o código fonte do ambiente através da linguagem de programação C++, enquanto aqueles que possuem apenas o conhecimento básico sobre algoritmos e modelagem computacional podem utilizar a linguagem de programação de alto nível *TerraME Modeling Language* – uma extensão da linguagem de programação LUA (Jerusalimschy et al. 1996), que permite a fácil escrita, leitura e alteração dos modelos (Carneiro 2003).

3. Eclipse

Desenvolver um sistema de computação e um modelo computacional para fenômenos sócio-ambientais são atividades essencialmente similares. O uso de um ambiente de desenvolvimento integrado (ou IDE, do inglês *Integrated Development Environment*) cujas características e funcionalidades contribuem para agilizar o processo de desenvolvimento de software é, portanto, também aplicável ao desenvolvimento de modelos para sistemas ambientais. Entretanto desenvolver um ambiente desta natureza possui um custo muito elevado. Uma alternativa à implementação de um ambiente totalmente novo é a utilização de ambientes, ferramentas e *frameworks* já existentes.

O Eclipse (<http://www.eclipse.org>) é um IDE *open source* que fornece uma plataforma comum para diversos produtos baseados em IDE e facilita a integração destes, servindo de base para a construção de ferramentas e aplicações diversas (Rivieres 2004). Seu grande diferencial é a sua capacidade de integração obtida por meio da sua arquitetura baseada em *plug-ins*. Um *plug-in* é a menor unidade funcional passível de ser desenvolvida e distribuída separadamente. A Plataforma Eclipse é construída num mecanismo de descobrir, integrar, e executar *plug-ins*. Desenvolver uma aplicação sobre a plataforma *Eclipse* permite que ela seja integrada a outras aplicações que também foram escritas sobre esta plataforma. (Eclipse 2006).

A Figura 1 mostra os principais componentes do *Eclipse Software Development Kit (Eclipse SDK)* e sua arquitetura baseada em *plug-ins*. O *Eclipse SDK* inclui a Plataforma Eclipse além de duas ferramentas úteis ao desenvolvimento de *plug-ins*: a *Java Development Tools (JDT)*, que implementa um ambiente de desenvolvimento Java, e o *Plug-in Developer Environment (PDE)* que adiciona ferramentas específicas ao desenvolvimento de *plug-ins* e extensões. Novas aplicações são desenvolvidas estendendo-se o sistema através de *plug-ins* (Eclipse 2006).

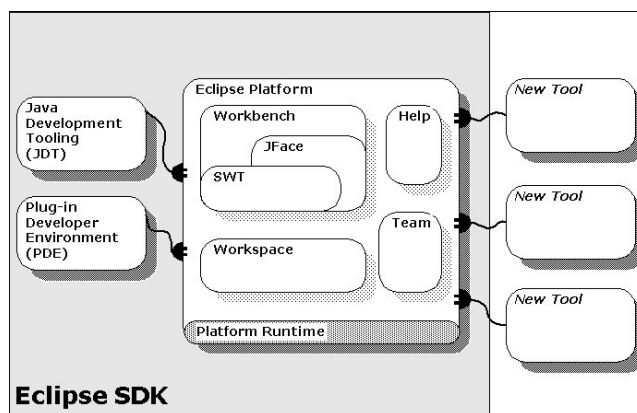


Figura 1. Visão geral do Eclipse SDK (Eclipse 2006)

4. TerraME GIMS

O *TerraME GIMS – TerraME Graphical Interface for Modeling and Simulation*, é uma interface gráfica que permite a construção de modelos dinâmicos espaço-temporais para o ambiente *TerraME* através da interação com componentes gráficos. Está sendo desenvolvido sobre a plataforma Eclipse, implementado e distribuído portanto como um conjunto de *plug-ins* para o Eclipse.

Em conformidade com as plataformas sobre as quais está sendo desenvolvido, *TerraME* e *Eclipse*, o *TerraME GIMS* apresenta uma arquitetura de software em camadas. O *TerraME GIMS* irá compor uma nova camada, entre o *TerraME* e o usuário final, sendo a plataforma *Eclipse* uma camada intermediária entre o *TerraME* e o *TerraME GIMS*, conforme ilustrado na Figura 2. Desta forma, não há restrição para a criação de modelos diretamente sobre a linguagem *TerraME* quando utiliza-se o *TerraME GIMS*.

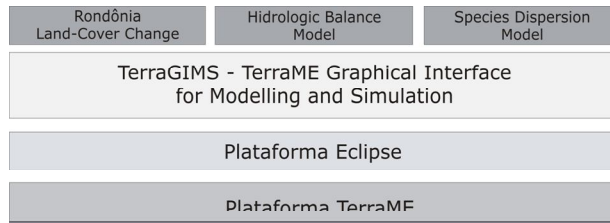


Figura 2. Visão geral da Arquitetura do *TerraME GIMS*

A construção de modelos é feita através do *Eclipse* e do conjunto de *plug-ins* que constituem o *TerraME GIMS*. A Figura 3 ilustra o ambiente de desenvolvimento dos usuários do *TerraME GIMS*.

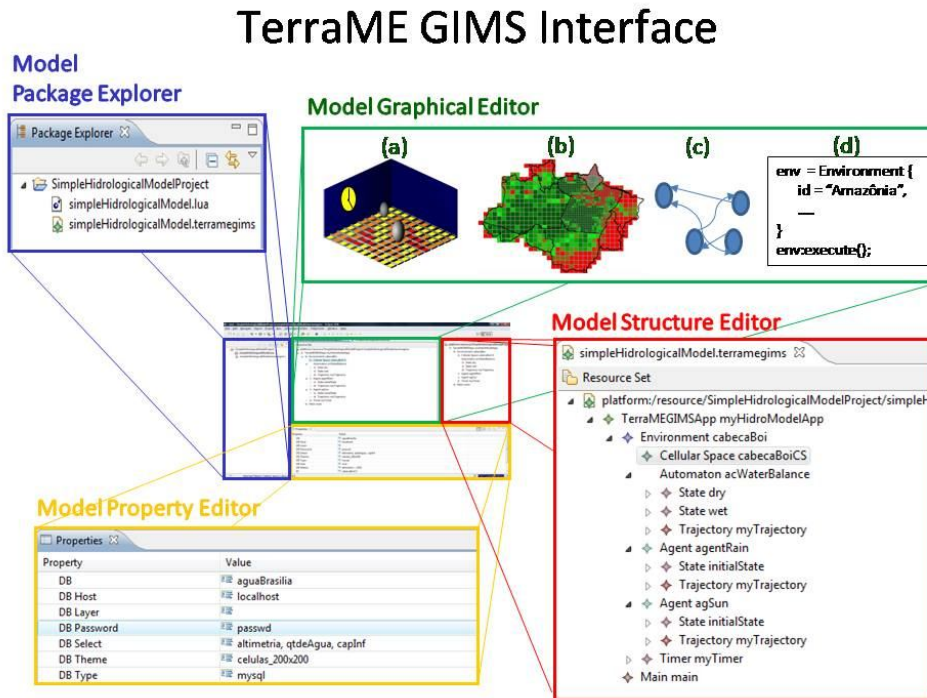


Figura 3. *TerraME GIMS* incorporado ao *Eclipse*

Através do *Model Package Explorer* os usuários podem acessar os diversos arquivos que fazem parte do projeto. O *Model Structure Editor* permite navegar pelo modelo estruturado hierarquicamente na forma de uma árvore. Uma vez selecionado um elemento do modelo, este poderá ser editado a partir do *Model Graphical Editor* e do *Model Property Editor*.

No *Model Graphical Editor* serão apresentadas ao usuário representações gráficas correspondentes aos diversos elementos do modelo, como por exemplo de um *Environment* em (a), contendo a representação de seus agentes, espaço e tempo; de um *CellularSpace* em (b), exibindo a representação do espaço celular carregado a partir de um banco de dados; de um *Agent* em (c), representado internamente como um autômato, constituído de estados e transições de estados; e o código *TerraME* correspondente em (d), gerado automaticamente em conformidade com o modelo criado pelo usuário a partir da interação com as diversas representações gráficas de seus elementos. As propriedades dos elementos podem ser visualizadas e editadas a partir do *Model Property Editor*, por meio de *widgets* (componentes gráficos de interação com o usuário) como campos de texto e caixas de seleção.

5. Resultados

O trabalho apresenta como resultados parciais de desenvolvimento uma versão do *TerraME GIMS* que permite a criação e visualização dos elementos do modelo representados na forma de uma árvore (*Model Structure Editor*), e a edição das propriedades dos elementos do modelo por meio de campos de texto e menus (*Model Property Editor*). Uma vez criado o modelo, o código *TerraME* correspondente é gerado automaticamente.

Um modelo *TerraME GIMS* é estruturado visualmente para os usuários na forma de uma árvore. A estrutura hierárquica na forma de árvore contribui para a percepção dos usuários na construção de modelos aninhados, com múltiplas escalas. A Figura 4 apresenta um exemplo simples de balanço hídrico, que permite ilustrar a visualização hierárquica do modelo, a visualização das propriedades do elemento *CellularSpace* e o código *TerraME* correspondente.

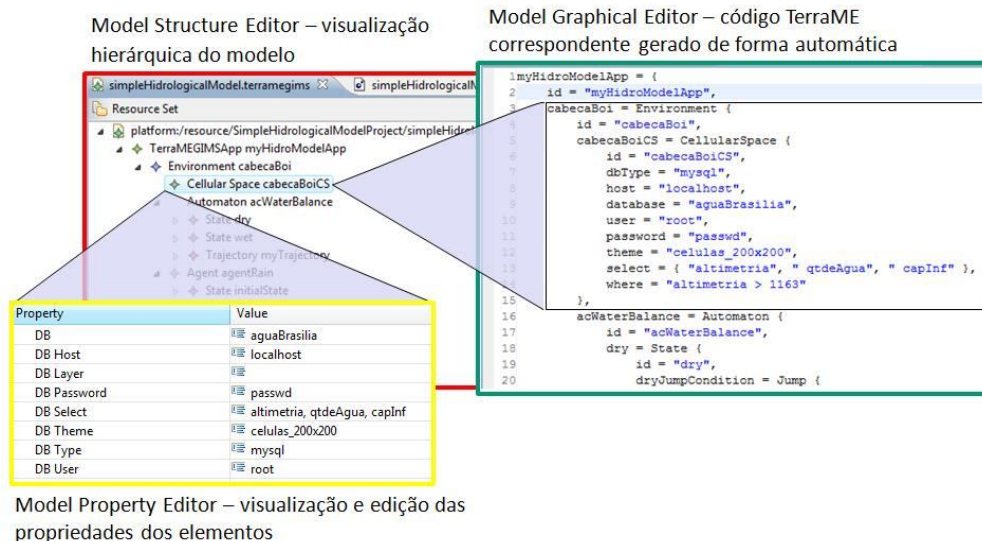


Figura 4. Visualização do hierárquica de um modelo *TerraME GIMS*

Um modelo *TerraME GIMS* é armazenado como um documento XML (<http://www.w3c.org/xml>), e o código *TerraME* correspondente é gerado (ou atualizado) automaticamente sempre que o modelo *TerraME GIMS* é salvo. A utilização

de documentos XML facilita a troca de informações entre diferentes aplicações, contribuindo para integrações futuras de outras ferramentas ao ambiente.

6. Conclusões

A partir dos resultados parciais obtidos com o desenvolvimento do *TerraME GIMS* já é possível vislumbrar seu grande potencial em ampliar a comunidade de usuários do *TerraME*, aumentando a produtividade dos atuais usuários e reduzindo a curva de aprendizado de novos usuários.

Além disto, a partir da utilização do Eclipse como plataforma base de desenvolvimento, uma série de recursos já existentes na plataforma, tais como facilidades para o desenvolvimento colaborativo e controle de versão, poderá ser utilizada na construção de modelos por equipes multidisciplinares e dispersas geograficamente. A arquitetura baseada em *plug-ins* do Eclipse também contribui para futuras extensões ao *TerraME GIMS* e sua reutilização.

As próximas etapas do trabalho incluem incorporar ao *Model Graphical Editor* um editor para construção de autômatos celulares e agentes a partir de diagramas de autômatos (contendo estados e transições de estados), o desenvolvimento de modelos baseados em agentes, e avaliação e testes da interface junto a usuários da aplicação.

6. Agradecimentos

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

References

- Carneiro, Tiago G. S. (2003) "Nested-CA: a foundation for multiscale modeling of land use and land change". São José dos Campos: INPE, 2003.
- Câmara, Gilberto. Carneiro, Tiago G. S., Bevilacqua, Luís (2007). I Curso de Verão Geoma 2007 – Modelagem Dinâmica com *TerraME*. http://www.dpi.inpe.br/geoma/curso_verao/index.php. Acessado em Agosto de 2008.
- Câmara, Gilberto. Carneiro, Aguiar, Ana P. D., Tiago G. S., Neto, Pedro R., Feitosa, Flávia. (2008). Course on Environmental Modeling – LUCC. http://www.dpi.inpe.br/cursos/environmental_modelling/index.html.
- Câmara, G., R. Souza, B. Pedrosa, L. Vinhas, A. Monteiro, J. Paiva, M. Carvalho and M. Gattass (2000). TerraLib: Technology in Support of GIS Innovation. II Workshop Brasileiro de Geoinformática, GeoInfo2000, São Paulo.
- Ierusalimschy, R., Figueiredo, L. H., Celes, W. (1996) "Lua - an extensible extension language", *Software: Practice & Experience* 26 #6. 635-652.
- Rivieres, J., Wiegand, J. (2004) Eclipse: A platform for integrating development tools. *IBM Systems Journal*, 43(2), 2004.
- Eclipse (2006) Eclipse Technical Platform Overview. <http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>. Acessado em Agosto de 2008.