

Ferramentas de monitoramento e análise de modelos espaciais dinâmicos em tempo-real

Antônio José da Cunha Rodrigues, Orientador Tiago Garcia de Senna Carneiro
PPGCC - Programa de Pós-Graduação em Ciência da Computação
UFOP - Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais, Brasil
email: aj.rodrigues@ymail.com, tiagogsc@gmail.com

Resumo—A Visualização Científica se refere a qualquer técnica que utiliza visualização e dados científicos. Neste trabalho, utilizaremos mecanismos de visualização para acompanhar e depurar, em tempo-real, as entidades de modelos ambientais. Essa visualização deverá ser executada de maneira eficiente, reduzindo o tempo de processamento e consumo de memória. Para tanto, deverá ser implementado um protocolo e estruturas de dados que viabilizem esse monitoramento. Como intuito de avaliar a eficiência dessas estruturas, serão feitos experimentos comparativos entre as versões e análises de complexidade em termos de tempo e espaço de memória.

Keywords—protocolo de comunicação, modelagem e simulação, sistemas terrestres

I. INTRODUÇÃO

A Visualização Científica se refere a qualquer técnica que utiliza visualização e dados científicos. Segundo DeFanti *et al.* [1], é uma forma de comunicação que transcende a aplicação e os limites tecnológicos. As técnicas de visualização sintetizam várias informações de maneira eficaz [2] e são usadas em diversas áreas, por exemplo, Medicina, Engenharia e Ciência. Na Medicina, a visualização científica é aplicada em diversos casos, por exemplo, ressonância magnética e tomografia. Na Engenharia, também é usada em várias situações, como, na indústria automobilística e em ferramentas *Computer-aided design* (CAD). Na Ciência, é aplicada em inúmeros casos, por exemplo, estudos da dinâmica populacional do mosquito da dengue e em estudos de fenômenos geográficos. Essa última aplicação será investigada neste trabalho. No qual, aplicaremos a Visualização Científica para monitorar a evolução de simulações espacialmente-explicitas de processos naturais e sociais, ou simplesmente ambientais.

A modelagem computacional permite realizar experimentos controlados que aplicados ao estudo de fenômenos geográficos simulam os impactos ambientais provocados pelo homem que, por sua vez, condizionarão suas ações futuras. Tais simulações, geram uma enorme quantidade de dados que necessitam ser analisados de maneira eficiente. Antes da execução dos experimentos, os modelos devem ser verificados e quaisquer falhas detectadas devem ser corrigidas. O cérebro humano tem maior facilidade em reconhecer

imagens do que letras ou números [3]. Geralmente, padrões invisíveis quando exibidos em tabelas ou em conjunto de dados são imediatamente percebidos quando apresentados de maneira sintetizada, na forma de gráficos e imagens [1]. Desta maneira, interfaces gráficas contendo mapas e gráficos que ilustram em tempo-real a evolução das variáveis de estado de um modelo ambiental podem tornar o processo de depuração desse modelo mais fácil e eficaz.

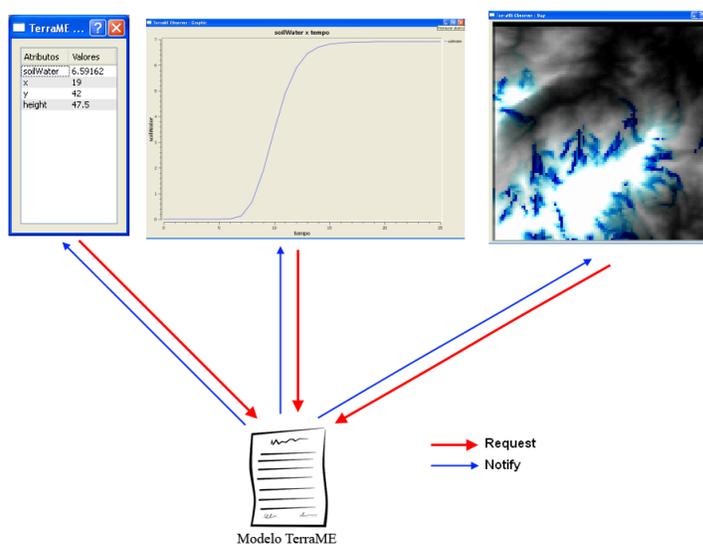


Figura 1. Padrão *Observer*, protocolo de comunicação e os observadores do tipo Tabela, Gráfico e Mapa (Adaptado de [4])

No ambiente de modelagem TerraME, as interfaces gráficas para monitoramento e análise dos modelos (*observer*) são implementadas na forma do Padrão *Observer* [4] onde cada interface gráfica, doravante chamada *observer*, é responsável por apresentar os dados de um *subject*. Um *subject* é qualquer entidade presente no modelo, cujos valores de suas propriedades determinam o estado corrente da simulação.

O TerraME¹ é um sistema para o desenvolvimento de modelos ambientais espacialmente-explicitos em múltiplas

¹<http://www.terrame.org>

escalas que permite o uso combinado de múltiplos paradigmas de modelagem, entre eles, teoria de agentes, teoria geral de sistemas ou teoria de autômatos celulares [5]. Ele foi desenvolvido em parceria pela Universidade Federal de Ouro Preto (UFOP) e pelo Instituto Nacional de Pesquisas Espaciais (INPE). Nesse sistema, vários *observers* podem ser acoplados a um único *subject*, para que o estado de uma entidade seja analisado de diferentes maneiras. Um *subject* ao ter seu estado alterado, notifica automaticamente todos os *observers* a ele acoplado. Um *observer* após receber essa notificação, requisita o novo estado do *subject* e passa a apresentá-lo. A Figura 1 apresenta a interação entre um *subject* e seus *observers*. Cada *observer* apresenta de maneira diferente os mesmos dados provenientes de um *subject*, neste caso, um componente de um modelo TerraME.

Na versão atual do Padrão *Observer* implementado na plataforma TerraME, o estado completo de um *subject* é serializado e transmitido a cada *observer* que o esteja monitorando, sem que nenhum processamento seja reaproveitado mesmo quando o mesmo estado precisa ser comunicado a diversos *observers*. Isto é, quando há dois ou mais *observers* associados a um mesmo *subject*, cada *observer* recupera o estado do *subject*, o utiliza e em seguida o descarta. Este processamento é realizado tantas vezes quanto for o número de *observers* lançados pelo usuário TerraME.

Neste trabalho, integraremos o padrão de projeto *BlackBoard* [6] às estruturas de dados existentes no sistema TerraME. O *BlackBoard* age como uma área de memória *cache*, na qual os estados recuperados a partir dos *subjects* são armazenados temporariamente e podem ser reutilizados por outros observadores. Desta forma, buscaremos reduzir o tempo de processamento envolvido na serialização dos estados, assim como a taxa de utilização do canal de comunicação entre *subjects* e *observers*.

Esta proposta está organizada da seguinte forma. A Seção II apresenta a justificativa para o desenvolvimento deste trabalho. Os objetivos são apresentados na Seção III. Na Seção IV demonstramos a metodologia utilizada nesta produção. Ao final, Seção V, apresentamos os resultados esperados.

II. JUSTIFICATIVAS

Uma imagem é mais facilmente compreendida pelo cérebro do que palavras ou números[2]. Desta forma, a utilização de observadores na depuração e na análise das simulações de sistemas geográficos facilitará o entendimento das interações homem-natureza e reduzirá o tempo gasto na análise de dados pois falhas serão detectadas com maior velocidade.

III. OBJETIVOS

Desenvolver um protocolo de comunicação e estruturas de dados que conferirão eficiência à comunicação entre *subjects* e *observers* TerraME. Avaliar e comparar a nova e a atual

versão desse protocolo quanto ao desempenho e usa de memória.

IV. METODOLOGIA

Nesta seção, apresentaremos alguns sistemas computacionais e métodos que serão empregados na execução deste trabalho. Além desses, será feita uma revisão da literatura. Através da qual, buscaremos métodos que conferirão eficiência em Visualização Científica.

Sistemas Computacionais

TerraME: TerraME é um *framework* de modelagem e foi criado para avaliar o modelo de Autômatos Celulares Aninhados - Nested-CA [5], utiliza múltiplas escalas onde, para cada escala há uma resolução espacial, analítica e temporal.

Métodos

Padrões de Projeto: De acordo com Gamma *et al.* [4], padrões de projeto são soluções genéricas que podem ser adaptadas e aplicadas a diversas situações. A ideia principal é a reutilização, ou seja, problemas diferentes podem ser resolvidos através da adaptação e aplicação de uma mesma solução.

- *Observer*: Para visualizar de forma automática a evolução do modelo foi utilizado o Padrão *Observer*. Gamma *et al.* [4] o descreve como sendo um padrão comportamental e é usado para manter a consistência entre objetos. Quando algum atributo de um objeto *Subject* muda, ele notifica todos os objetos *Observer* associados a ele. Cada *observer*, após requisitar ao *subject* seu estado corrente, se atualiza e apresenta esse novo estado.

- *BlackBoard*: Segundo Buschmann *et al.* [6], é um padrão arquitetural. Pode ser visto como um repositório de dados, onde programas independentes trabalham cooperativamente sobre uma mesma estrutura de dados e montam sobre essa estrutura suas bases de conhecimento. Cada programa é especialista em resolver apenas uma parte do problema e juntos buscam encontrar uma solução. Essa solução pode ser parcial ou aproximada. O nome “BlackBoard” foi dado baseando-se em um exemplo real: especialistas trabalham juntos em frente a um quadro negro, cooperando e utilizando suas habilidades para resolver um problema. A cada instante, um deles se desloca até o quadro e utiliza suas habilidades em busca de uma solução para esse problema. A diferença entre o BlackBoard e um repositório de dados é que o primeiro, possui um estrutura interna de controle [6]. Essa diferença será explorada neste trabalho pois, o uso de uma estrutura de controle permitirá uma melhor gerência do consumo de memória.

V. RESULTADOS ESPERADOS

Ao término deste trabalho, esperamos uma versão eficiente do protocolo e das estruturas de dados estejam

implementados, em comparação com a atual versão, e que viabilize monitoramento das variáveis de estado do modelo em estudo seja feito de maneira eficaz consumindo o menor quantidade de memória.

REFERÊNCIAS

- [1] T. A. DeFanti, M. D. Brown, and B. H. McCormick, "Visualization: Expanding scientific and engineering research opportunities," *IEEE Computer Society Press*, vol. 22, pp. 12–25, 1989. [Online]. Available: <http://portal.acm.org/citation.cfm?id=72885.72886>
- [2] C. Kelleher and T. Wagener, "Ten guidelines for effective data visualization in scientific publications," *Environmental Modelling & Software*, vol. 26, no. 6, pp. 822 – 827, 2011.
- [3] K. Cukier, "A special report on managing information: Data, data everywhere," *The Economist*, feb 2010. [Online]. Available: http://www.economist.com/node/15557443?story_id=15557443
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional, 1995.
- [5] T. G. S. Carneiro, "Nested-ca: um fundamento para a modelagem de uso e cobertura do solo em múltiplas escalas," PhD Thesis, INPE - Instituto Nacional de Pesquisas Espaciais, Brazil, Computação Aplicada, Jun. 2006.
- [6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., 1996.
- [7] R. Ierusalimschy, "A linguagem lua," 2007, <http://www.inf.puc-rio.br/roberto/talks/ufrj-2007.pdf>, visitado em 05/07/2011.
- [8] —, *Programming in Lua, Second Edition*. Lua.Org, 2006.