

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

Antônio José da Cunha Rodrigues

Orientador: Tiago Garcia de Senna Carneiro

Departamento de Computação  
UFOP

22 de julho de 2011



Universidade Federal  
de Ouro Preto

## 1 Introdução

- Visualização Científica
- Justificativa
- Objetivos

## 2 Metodologia

- Ambiente TerraME
- Padrões de Projeto
- Análise de Complexidade

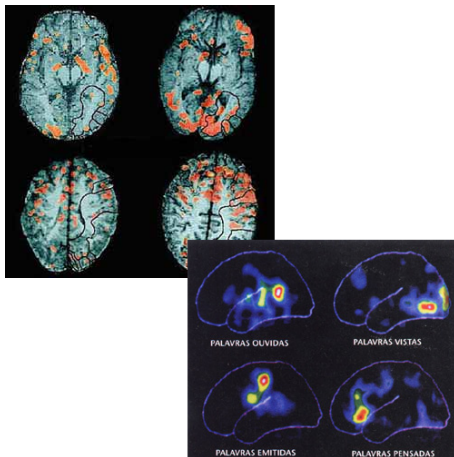
## 3 Experimentos

- Ambiente de testes
- Resultados

## 4 Conclusões

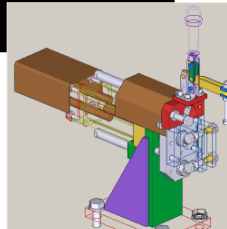
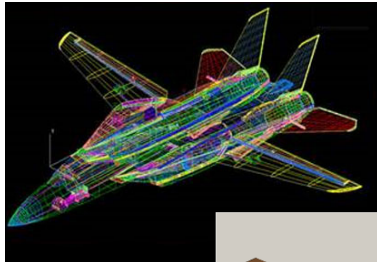


# Visualização Científica - Na Medicina



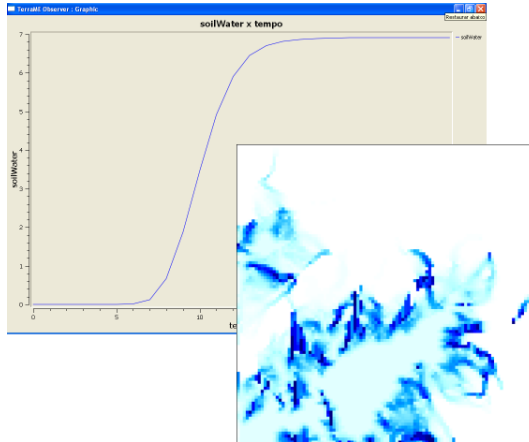
(Fonte: [www.google.com](http://www.google.com))

# Visualização Científica - Na Engenharia



(Fonte: [www.google.com](http://www.google.com))

# Visualização Científica - No estudo de fenômenos ambientais



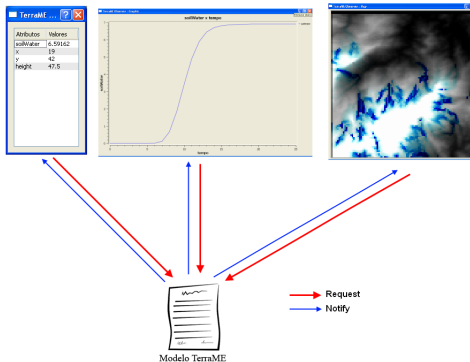
# Justificativa

- Um grande volume de dados são gerados em simulações e necessitam ser analisados de forma eficiente
- O monitoramento das variáveis de estado de um modelo pode reduzir o tempo de depuração desse modelo



# Objetivos

- Desenvolver algoritmos e estrutura de dados que viabilize o monitoramento das entidades do modelo.
- Analisar a complexidade do algoritmo proposto



## 1 Introdução

- Visualização Científica
- Justificativa
- Objetivos

## 2 Metodologia

- Ambiente TerraME
- Padrões de Projeto
- Análise de Complexidade

## 3 Experimentos

- Ambiente de testes
- Resultados

## 4 Conclusões



# Ambiente de modelagem TerraME

- Múltiplas escalas
  - Resolução espacial, analítica e temporal
- Múltiplos paradigmas
  - Teoria geral de sistemas
  - Teoria agentes
  - Teoria de autômatos celulares
- Interfaces gráficas são implementadas na forma do Padrão *Observer*
  - Monitoram as variáveis de estado de um modelo (Célula, Espaço Celular, entre outros)

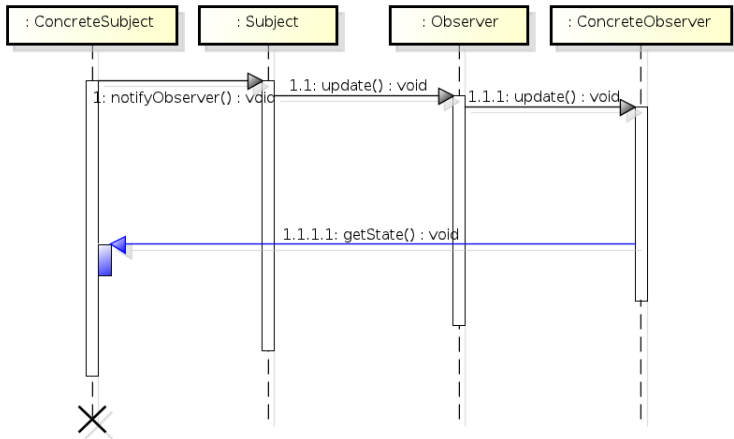


# Padrões de Projeto

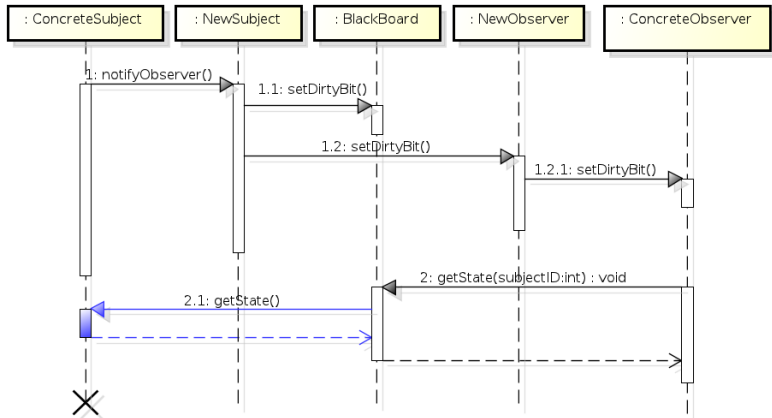
- Padrão *Observer* (Gamma *et al.*, 1995)
  - Padrão comportamental
  - Manter a consistência entre objetos
- Padrão *BlackBoard* (Buschmann *et al.*, 1996)
  - Padrão arquitetural
  - Pode ser visto como um repositório de dados. No entanto, possui um estrutura interna de controle



# Comunicação Atual



# Nova Versão



# Algoritmos

- VERSÃO ATUAL

**Algorithm** *Subject::getState(data)*

```
1 for each cell in cellSpace do
2     for each attributes in cell do
3         data+ = [attributeType, attributeName, attributeValue]
4 ...
5 retorna data
```

- Complexidade de tempo
  - Para todos os casos:  $O(n \times m)$
- Complexidade de espaço
  - A alocação é temporária
- $n$  é a quantidade de células no espaço celular e
- $m$  é a quantidade de atributos em cada célula.



# Algoritmo - Nova versão

- NOVA VERSÃO

**Algorithm** *BlackBoard::getState(subject)*

```
1 ...  
2 if (data[subjectId] NÃO É VÁLIDO) then  
3     data[subjectID] = subject->getState(aux)  
4 retorna data[subjectID]
```



# Análise de Complexidade

- Complexidade de tempo
    - Melhor Caso:  $O(1)$
    - Pior Caso:  $O(n \times m)$
  - Complexidade de espaço
    - Pior Caso:  $O(k \times n \times m)$
  - Complexidade no canal de comunicação
    - Melhor Caso: não há comunicação, os dados são estão no *BlackBoard*
    - Pior Caso:  $O(k \times n \times m)$
- $n$  é a quantidade de células no espaço celular,
  - $m$  é a quantidade de atributos em cada célula e
  - $k$  é o número de *observers* associados a células distintas.



## 1 Introdução

- Visualização Científica
- Justificativa
- Objetivos

## 2 Metodologia

- Ambiente TerraME
- Padrões de Projeto
- Análise de Complexidade

## 3 Experimentos

- Ambiente de testes
- Resultados

## 4 Conclusões



# Ambiente de testes

- Processador: Pentium D 2,8GHz
- Memória: 3,2GB
- Sistema Operacional: Windows XP SP3

Cinco baterias de teste, avaliando:

| Implementações | Padrão Utilizado      | Estrutura de dados | Gerenciado |
|----------------|-----------------------|--------------------|------------|
| Versão 1       | Observer              | nenhuma            | não        |
| Versão 2       | Observer + BlackBoard | Hash e Vector      | não        |
| Versão 3       | Observer + BlackBoard | Hash e Vector      | sim        |

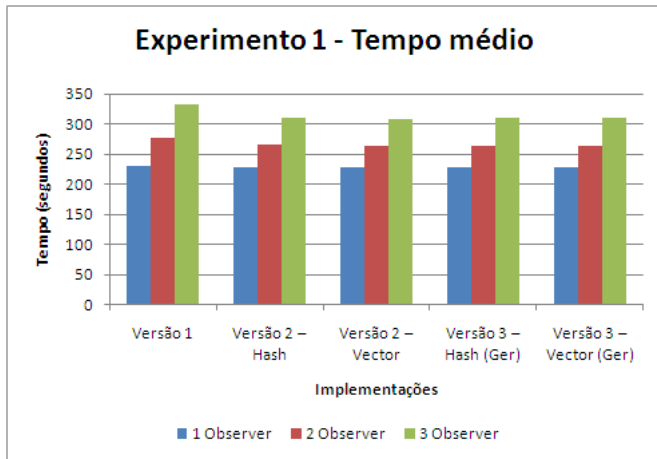


# Experimento 1

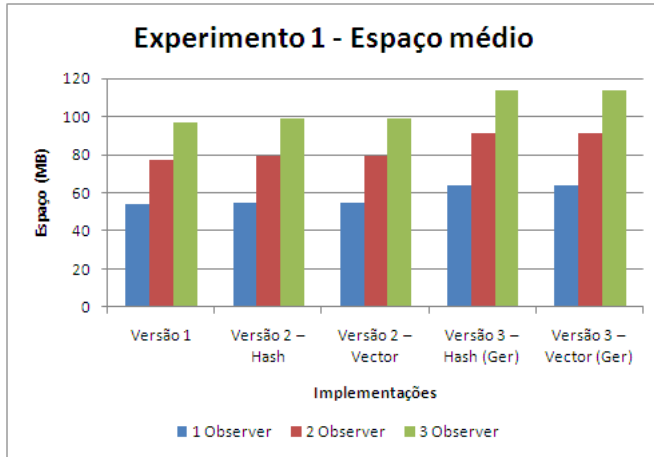
| Experimentos | Número de Observadores |
|--------------|------------------------|
| Teste 1      | 1                      |
| Teste 2      | 2                      |
| Teste 3      | 3                      |



# Experimento 1 - Tempo



# Experimento 1 - Espaço

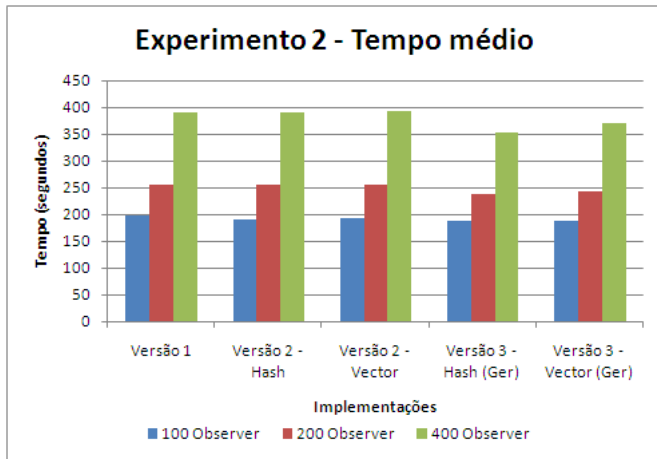


## Experimento 2

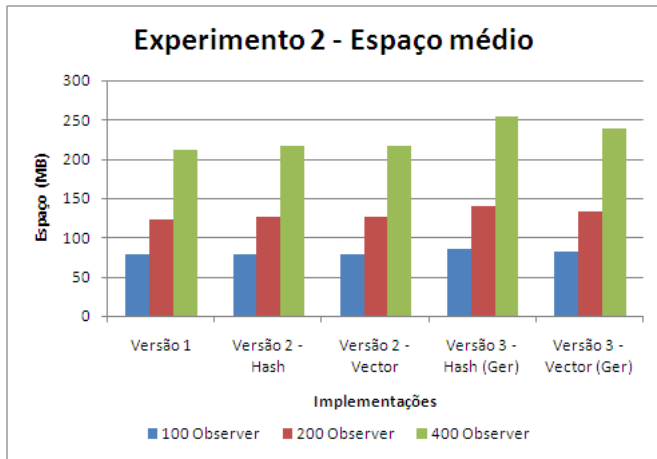
| Experimentos | Número de Observadores |
|--------------|------------------------|
| Teste 1      | 100                    |
| Teste 2      | 200                    |
| Teste 3      | 400                    |



## Experimento 2 - Tempo



## Experimento 2 - Espaço



## 1 Introdução

- Visualização Científica
- Justificativa
- Objetivos

## 2 Metodologia

- Ambiente TerraME
- Padrões de Projeto
- Análise de Complexidade

## 3 Experimentos

- Ambiente de testes
- Resultados

## 4 Conclusões



# Conclusão

- Aumento no desempenho da simulação
- Maior consumo de memória



# Trabalhos Futuros

- Aprimorar a gestão do *BlackBoard*, baseando-se em políticas de *cache*.
- Considerar a recuperação apenas dos dados modificados.
- Avaliar a possibilidade de utilizar programação paralela para recuperação dos dados do modelo.
- Investigar a perda de desempenho ao utilizar um grande número de *observers*.

