

# Modelagem e Implementação de um Visualizador para Simulações Computacionais em Redes de Sensores Sem Fio

Jesimar da Silva Arantes

Orientador: Tales Heimfarth

Bacharelado em Ciência da Computação  
Universidade Federal de Lavras

Agosto – 2013

GRUBi 

# Sumário

- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

- 1 **Introdução**
  - Contextualização
  - Motivação
  - Objetivos
- 2 Referencial Teórico
- 3 Metodologia
- 4 Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

### Ideia Básica

- Este trabalho tem como foco principal apresentar a modelagem e implementação de um software de visualização de simulações em RSSF
- Este visualizador tem como entrada um arquivo de *log*, gerado por um simulador (*framework*)
- O visualizador deve apresentar de forma visual e com interface amigável a simulação, para auxiliar a programação da RSSF

### RSSF

- 1 Uma RSSF é composta por dispositivos denominados nós sensores
  - São sistemas embarcados simples
  - Possuem um ou mais sensores conectados a ela
- 2 RSSFs são aplicadas a diversos problemas
  - Detecção de intrusos
  - Vigilância e detecção de desastres
- 3 Simulações computacionais em RSSFs
  - Baixo custo
  - Facilidade de implementação
  - Interpretação aproximada da realidade
  - Simulador utilizado: GrubiX

### Motivação

- O antigo visualizador do *framework* GrubiX (*Visual ShoX*) apresenta algumas deficiências
- RSSF é uma tecnologia chave para o futuro
- Existência de poucos softwares de simulação especializados em RSSF
- Trabalhar com simulações computacionais são mais fáceis de implementar e testar que as simulações reais e são menos onerosas

# Introdução

## Objetivos

### Objetivo Geral

Modelar e implementar um software para visualização de simulações computacionais

### Objetivo Específicos

1. Elaboração do Projeto:
  - Pesquisa em bibliografia sobre visualizadores em RSSF
  - Elaboração de um software visualizadores de RSSF
  - Comparação com outros softwares de visualização em RSSF
2. Desenvolvimento do Projeto:
  - Levantamento de requisitos
  - Diagramas de classes
  - Diagramas de pacotes
  - Definição do modelo de desenvolvimento de software
  - Implementação

- 1 Introdução
- 2 Referencial Teórico
  - Redes de Sensores Sem Fio
  - GrubiX
  - Visualizadores de RSSF
- 3 Metodologia
- 4 Desenvolvimento
- 5 Resultados
- 6 Considerações Finais



### Definição

RSSF é um tipo de rede que consiste em um grande número de nós sensores autônomos que cooperam entre si. Estes são implantados sobre um determinada área [Aboelaze 2005].

- Podem ser compostas por dezenas até milhares de nós sensores
- Podem monitorar, medir e coletar dados do ambiente
- São autônomos

# Referencial Teórico

## Redes de Sensores Sem Fio

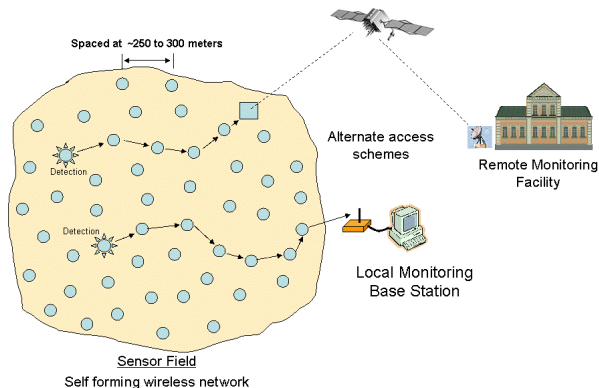


Figura: Exemplo de cenário: Fonte [Alico Systems 2011] GRUBi<sup>(S))</sup>

# Referencial Teórico

## Redes de Sensores Sem Fio

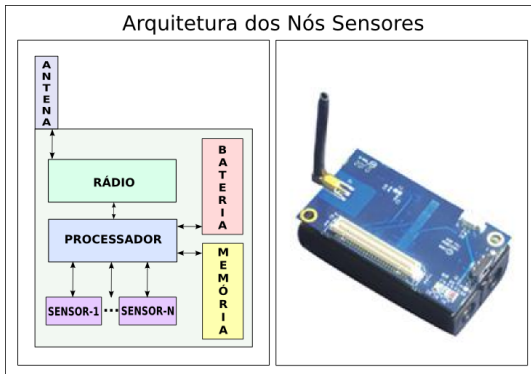


Figura: Componentes de um nó sensor

### Visão Geral

- É um *framework* para RSSF
- Desenvolvido em Java
- O GrubiX segue o paradigma orientado a objetos
- Permite criar facilmente aplicações ligadas a RSSF
- Após executar a aplicação desenvolvida um arquivo de *log* é gerado
- Este arquivo será o elemento principal que o *Visual GrubiX* utilizará
- É um *framework* orientado a eventos discretos
- Foi modelado especificamente para RSSF, ao contrário da maioria dos outros simuladores

# Referencial Teórico

GrubiX

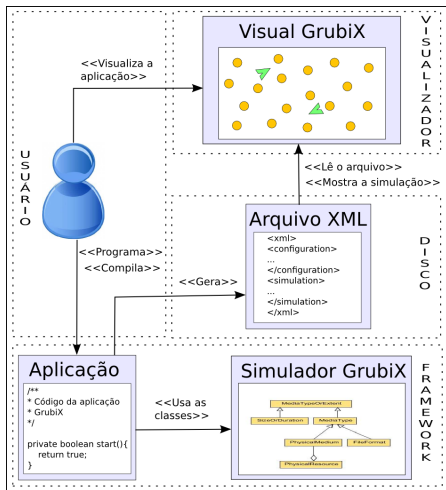


Figura: Etapas envolvidas no uso do simulador GrubiX

GRUBIX

### Visão Geral

- São software capazes de mostrar de forma visual o funcionamento de uma RSSF
- Os dados coletados pelas RSSFs são salvos, em geral, usando algum tipo de arquivo
- Os visualizadores auxiliam a exibir estes dados
- Alguns exemplos de visualizadores são: *Visual ShoX*, *Nam* e *TinyViz*

### Classificação da Visualização de Eventos

- Online
  - Os eventos são exibidos enquanto a simulação está executando
  - Permite ver o comportamento da rede imediatamente
  - Economia de tempo
- Offline
  - Os eventos são registrados em um arquivo de *log* e são exibido após a simulação terminar
  - Pode-se visualizar quantas vezes desejar
  - Pode-se selecionar algum ponto específico que se deseja visualizar
  - Alterar a velocidade da animação

### Visual ShoX

- É um visualizador desenvolvido para o ShoX
- Visualizador *offline*
- A arquitetura segue o paradigma MVC
- Carrega arquivos XML e Compact para exibir a simulação
- Controles iniciar, pausar, continuar e reiniciar a simulação
- Contém também alguns níveis de velocidade e *zoom*.



# Referencial Teórico

## Visualizadores de RSSF

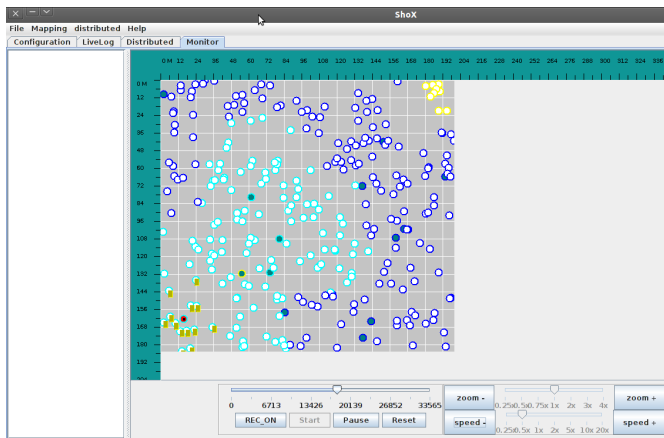


Figura: Tela do visualizador do ShoX. Fonte: [Lessmann and Heimfarth 2008].

GRUBi)))

### Visualizador Nam

- Utilizado no *framework* NS-2
- Visualizador *offline*
- Os nós podem ter três formas possíveis - círculos, retângulos e hexágonos
- A forma de visualização dos nós não pode ser alterada em tempo de execução

# Referencial Teórico

## Visualizadores de RSSF

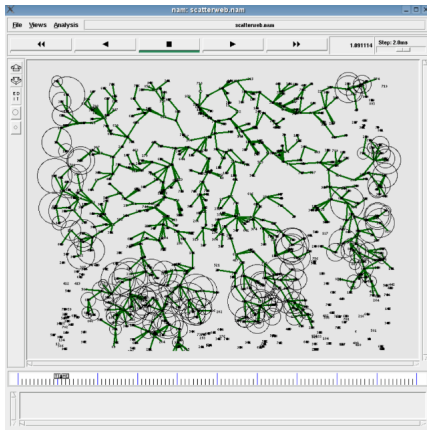


Figura: Tela do visualizador *Nam*

### Visualizador TinyViz

- É uma ferramenta utilizada no TOSSIM
- TOSSIM é um dos primeiros simuladores puramente orientada para RSSF
- Visualizador *online*
- Possui a capacidade de executar a visualização em tempo real
- Visualiza as leituras dos sensores, estados dos LEDs e ligações do rádio

# Referencial Teórico

## Visualizadores de RSSF

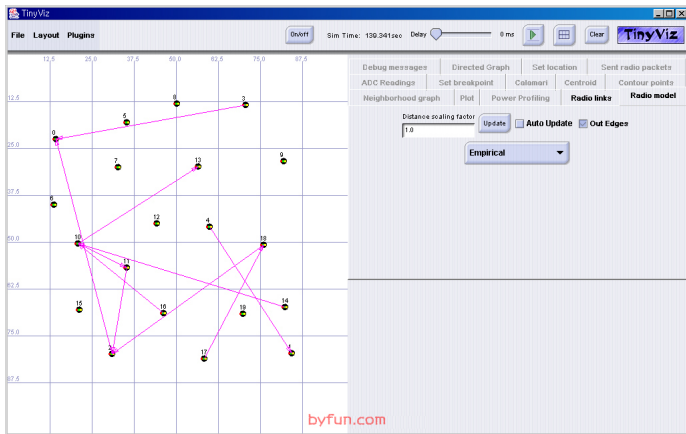


Figura: Tela do visualizador TinyViz

- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia**
  - Ferramenta Utilizada
  - Arquitetura do Software
  - Ciclo de Vida do Software
- 4 Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

### Linguagem Java

- Suporta os principais conceitos de orientação a objetos
- Favorece extensibilidade e reusabilidade de código
- É altamente portátil, ou seja, é independente de plataforma
- Suporta aplicações concorrentes fazendo uso de *multithreads*
- Grande quantidade de APIs disponíveis

# Metodologia

## Arquitetura do Software e Ciclo de Vida

### Model-View-Controller

Este padrão de projeto foca na completa separação entre os elementos da interface gráfica, regras de negócio e estrutura de dados.

### Ciclo de Vida do Software

- Levantamento de requisitos
- Análise
- Projeto
- Implementação
- Testes
- Aceitação
- Implantação
- Manutenção do software



- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Desenvolvimento**
  - Estrutura do Arquivo de Log
  - Arquitetura do Visualizador
  - Principais Recursos Desenvolvidos
- 5 Resultados
- 6 Considerações Finais

### Arquivo de Log

- O arquivo de saída do simulador GrubiX está no formato XML
- O XML é armazenado em estruturas hierárquicas em formato de árvore
- Após carregado pelo visualizador, será criado uma lista de eventos a serem exibidos na tela

# Desenvolvimento

## Estrutura do Arquivo de Log

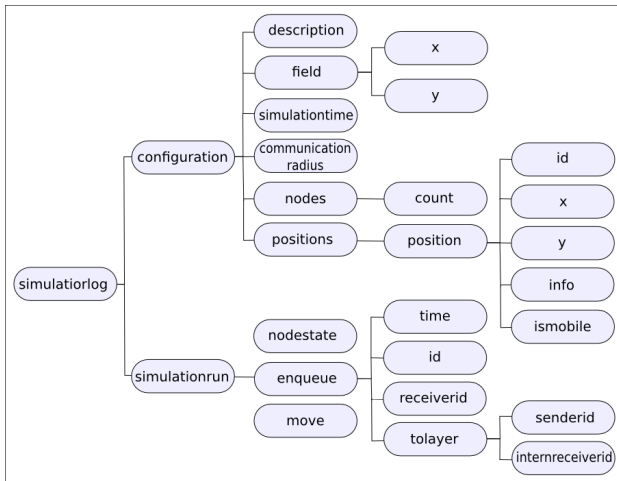


Figura: Estrutura do Arquivo de Log.

### Arquitetura de Multi-Layer

- Primeira camada - Ambiente
- Segunda camada - Nós
- Terceira camada - Eventos
- Quarta camada - Informações

# Desenvolvimento

## Arquitetura do Visualizador

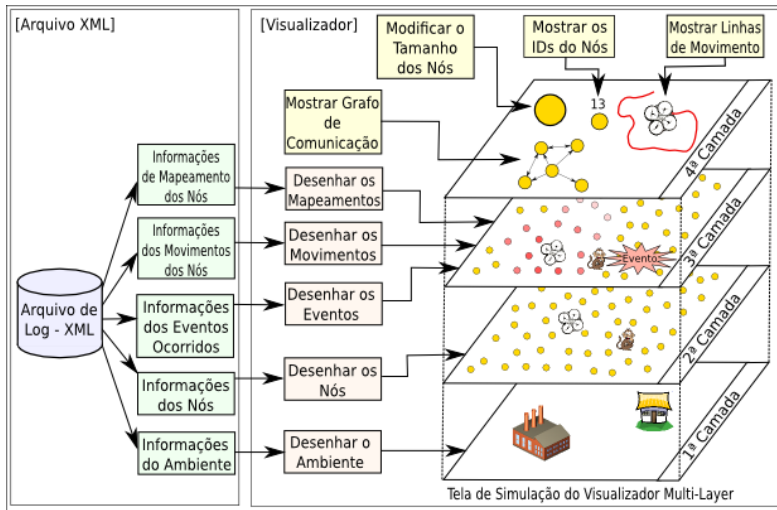


Figura: Arquitetura do Visualizador.

# Desenvolvimento

## Principais Recursos Desenvolvidos

### Recursos

- Transmissão de Pacotes
- Movimentação dos Nós
- *Mapping*
- Grafo de Conectividade
- Linhas de Movimentação

# Desenvolvimento

## Grafo de Conectividade

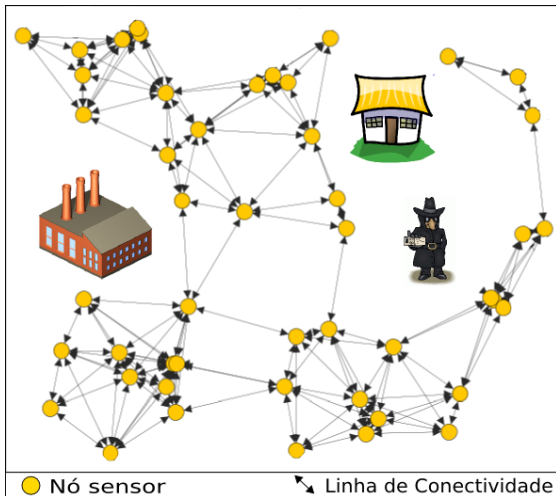


Figura: Grafo de Conectividade.

# Desenvolvimento

## Linhas de Movimentação

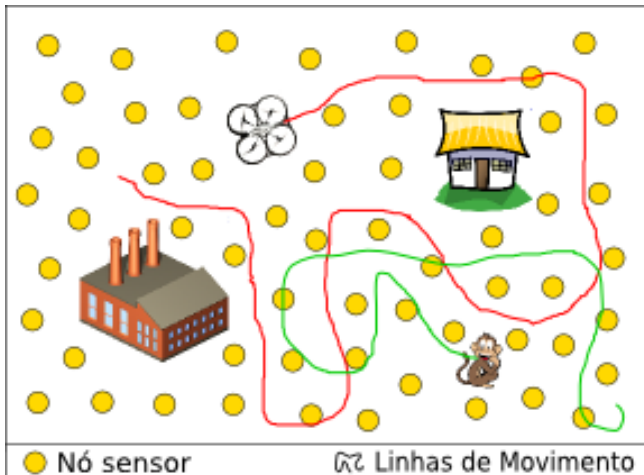


Figura: Linhas de Movimentação.



- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Desenvolvimento
- 5 Resultados**
  - Software Desenvolvido
  - Comparação
- 6 Considerações Finais

# Resultados

## Software Desenvolvido

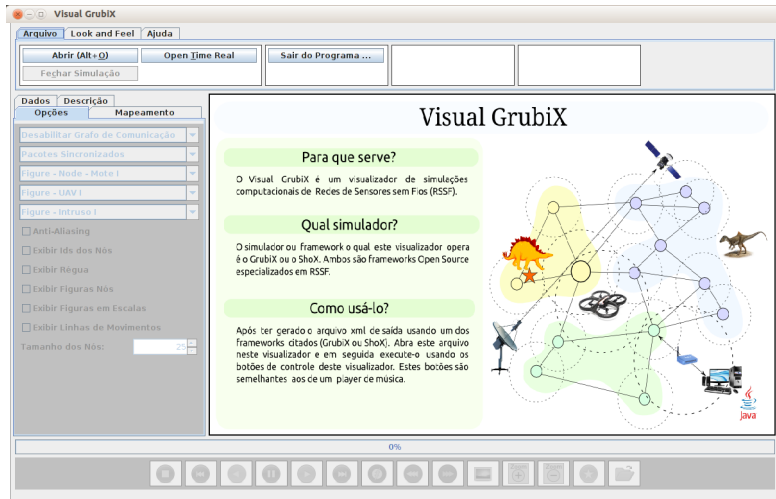


Figura: Interface do *Visual GrubiX*.

GRUBiX

# Resultados

## Software Desenvolvido

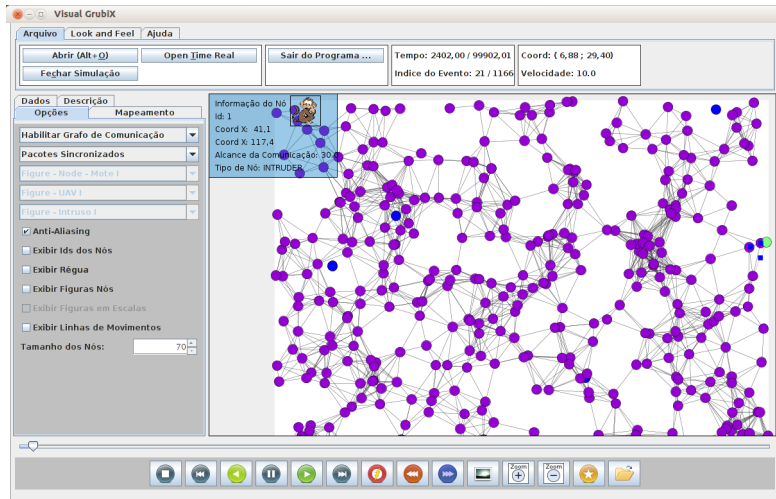


Figura: Interface do *Visual GrubiX*.

GRUBi<sup>(3)</sup>

# Resultados

## Comparação

Recursos \ Visualizador	<i>Visual GrubiX</i>	<i>Visual ShoX</i>
Multiplataforma	Sim	Sim
Possui mapeamento de elementos	Sim	Sim
Nós da rede selecionáveis	Sim	Não
Mostra grafo conectividade	Sim	Não
Mostra linhas de movimentação	Sim	Não
Mostra imagens nós	Sim	Não
Possui análise estatística	Não	Sim
Definir as configuração da rede	Não	Sim
Rapidez carregar arquivo	Rápido	Lento
Controle da animação	± velocidade	± velocidade
Idiomas	Inglês / Português	Inglês
Abrir arquivos <i>log</i>	XML	XML / <i>Compact</i>
Carregamento dos arquivos	Memória / <i>Buffer</i>	Memória
<i>ProgressBar</i> simulação	Dinâmica	Estática
Forma de zoom	Usa mouse ou botões	Usa <i>slider</i>
Forma de translação	Usa mouse ou botões	Usa <i>slider</i>

Tabela: Comparação do *Visual GrubiX* e do *Visual ShoX*.

GRUBi<sup>(u))</sup>

- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Desenvolvimento
- 5 Resultados
- 6 Considerações Finais**
  - Conclusão
  - Trabalhos Futuros

# Considerações Finais

## Conclusão

### Conclusão

- Este trabalho desenvolveu o *Visual GrubiX* para visualização de simulações em RSSF
- O objetivo de conseguir uma visualização de maneira fácil e intuitiva foi atingido
- Fácil migração entre o *Visual ShoX* e *Visual GrubiX*
- O *Visual GrubiX* possui um visual rico e permite depuração em rede

# Considerações Finais

## Conclusão

### Conclusão

- O *Visual GrubiX* possui mais recursos que o *Visual ShoX*
- O software proposto já é amplamente utilizado no laboratório GRUBi
- Um dos focos foi desenvolver um software com alta manutenibilidade e legibilidade de código

# Considerações Finais

## Trabalhos Futuros

### Trabalhos Futuros

- Integrar o visualizador com o *Google Maps*
- Produzir vídeos da simulação corrente
- Dar suporte a novos formatos de arquivo de saída como o *Compact*
- Poder visualizar a parte do *script* XML que está sendo executada



Obrigado a Todos!!!



Lessmann, J. and Heimfarth, T. (2008).

Flexible offline-visualization for mobile wireless networks.

In *Proceedings of the Tenth International Conference on Computer Modeling and Simulation*, UKSIM '08, pages 404–409, Washington, DC, USA. IEEE Computer Society.