Desenvolvimento de um software *open source* para a modelagem e simulação do Sistema Imune Humano

Exame de Qualificação

Mestrado em Ciência da Computação

Brenno Lemos **Orientador: Prof. Alexandre B. Pigozzo**

24 de março de 2023

Universidade Federal de São João del-Rei

UFSJ

Sumário

1 Introdução

Motivação

- **Objetivo**: desenvolver um software para facilitar a construção e simulação de modelos matemáticos e computacionais;
  - Romper a barreira envolvida: não-programadores também devem conseguir usufruir do software;
- **Como**: utilizando uma representação fácil de entender, flexível para manutenção e eficiente em computação;
  - E mais: ofertar ao usuário a opção de exportar o modelo como código. Assim, usuários mais avançados (ou da área) poderão usufruir de modificar o modelo a seus gostos, enquanto usuários iniciantes poderão aprender a construir seus próprios;

Referencial Teórico

1 Introdução

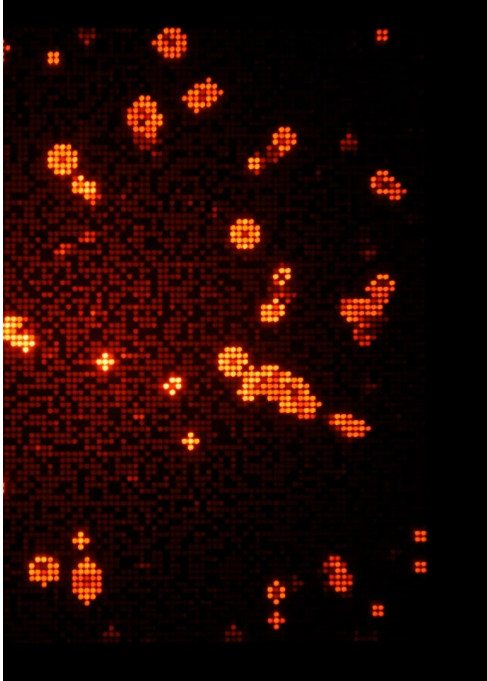
Universidade Federal
de São João del-Rei

UFSJ

Autômatos Celulares

1 Introdução

- Consiste em uma grade de células, cada uma em um estado de um conjunto finito;
- As células transitam entre estados utilizando uma função que recebe os estados de seus vizinhos como entrada;
- O objetivo é estudar a evolução do sistema com o tempo;
- Com poucas regras é possível criar comportamentos complexos;



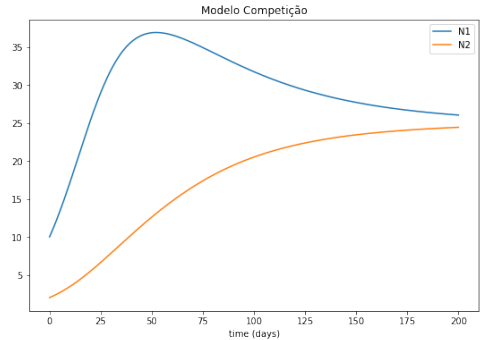
Equações Diferenciais Ordinárias (EDOs)

1 Introdução

- Usadas para estudar o comportamento populacional ao longo do tempo;
- Diversas aplicações em várias áreas do conhecimento;
- Cada equação descreve a concentração de uma população diferente;

$$\frac{dN_1}{dt} = r_1.N_1(1 - W_{11}.N_1 - W_{21}.N_2)$$

$$\frac{dN_2}{dt} = r_2.N_2(1 - W_{22}.N_2 - W_{12}.N_1) \quad (1)$$



EDO — Modelo Predador-Presa

Um modelo clássico da literatura é o modelo Predador-Presa. Este modelo descreve o comportamento de duas populações, H e P, que possuem uma uma relação de predação entre si.

$$\begin{aligned} \frac{dH}{dt} &= r.H - a.H.P \\ \frac{dP}{dt} &= b.H.P - m.P \end{aligned} \tag{2}$$

Na equação, temos que

H	Presa
P	Predador
r	Taxa de reprodução da presa
m	Taxa de mortalidade dos predadores
a	Taxa de predação
b	Taxa de reprodução dos predadores

Sumário

2 Trabalhos Relacionados

Trabalho Anterior — AC Designer

2 Trabalhos Relacionados

O trabalho que levou diretamente a este, AC-Designer, envolveu o desenvolvimento de um software para a criação e simulação de Autômatos Celulares. O software conta com várias funcionalidades que foram trazidas para este, como exportação de código e simulação *onsite*.

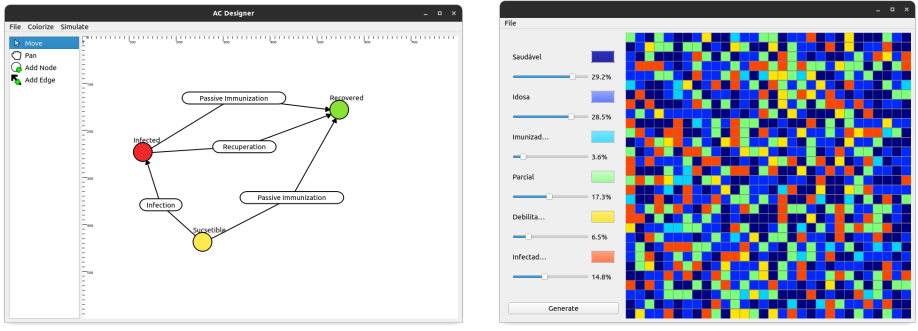
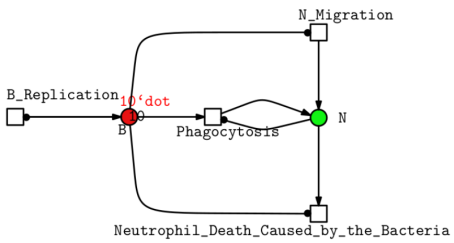


Figura 1: Tela principal e de simulação do software.

- Permite a modelagem de Redes de Petri;
- Utiliza uma representação baseada em grafos;
  - Círculos representam estados (*places*) e quadrados representam transições;
- Foi uma grande inspiração às interfaces deste e do trabalho anterior, por ser simples de entender e representar;
- Assim como este trabalho, possui simulação *onsite*;



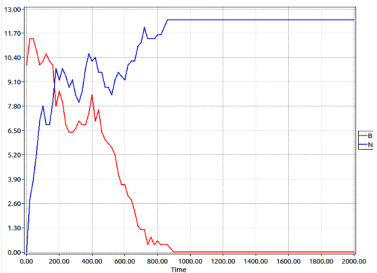


Figura 2: Exemplo de um modelo na interface e o resultado da simulação.

VCell e InsightMaker

2 Trabalhos Relacionados

- O VCell é uma plataforma de modelagem de sistemas biológicos celulares;
- Os modelos são construídos com base em regras;

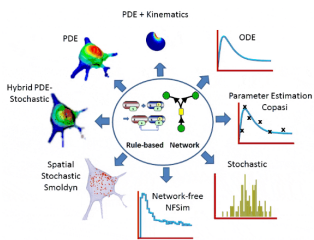


Figura 3: Modelos suportados pelo VCell.

- O InsightMaker permite a construção de modelos utilizando diagramas de Dinâmica de Sistemas e modelos baseados em agentes;
- Internamente, usa um modelo de EDOs;

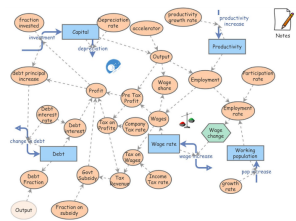


Figura 4: Exemplo da GUI do InsightMaker.

UFSJ

Sumário

3 Metodologia

Fluxograma de uso simplificado

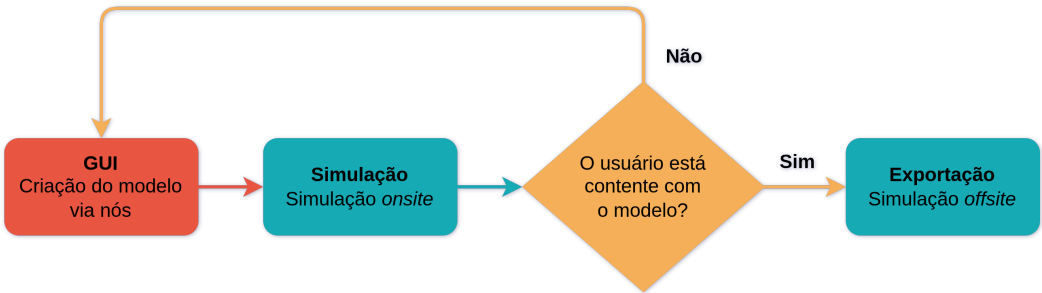


Figura 5: Fluxograma da experiência do usuário.

Funcionalidades do Software

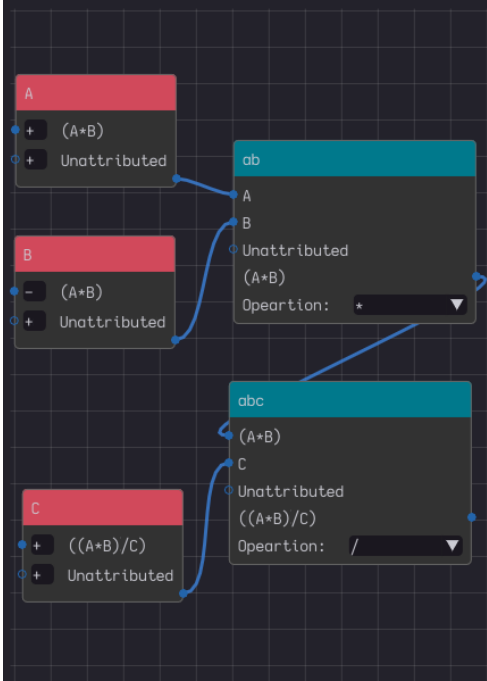
O software deve entregar as seguintes funcionalidades:

- Criação de modelos pela interface gráfica;
- Simulação do modelo e exibição dos resultados na interface;
- Exportação de PDF/Imagens com os resultados das simulações;
- Exportação de código equivalente ao modelo implementado;

Representação do Modelo

3 Metodologia

- O software necessita de uma interface simples de ser usada;
  - Mas também deve naturalmente relembrar uma EDO;
- Após diversas iterações, chegamos numa interface baseada em programação visual;
  - Estas interfaces existem desde 1963, mas tiveram uma renascença com o avanço dos computadores e a necessidade de softwares de edição de imagem/áudio/vídeo;



Fluxo de passagem de informações na interface

3 Metodologia

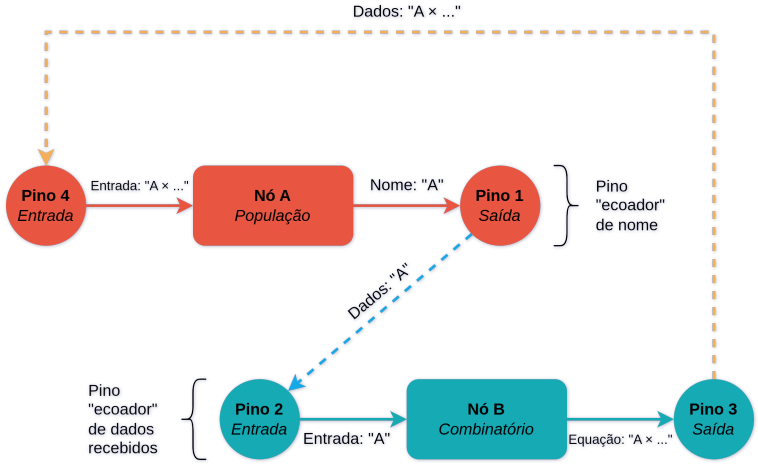


Figura 6: Relações entre nós e pinos.

GRaIL — 1968

3 Metodologia

Representação Intermediária

3 Metodologia

Universidade Federal
de São João del-Rei

Representação Intermediária

- Com o objetivo de realizar tantas transformações, torna-se necessário a utilização de uma Representação Intermediária (RI);
- Inspirados nas arquiteturas de compiladores modernos (GCC, baseados em LLVM), separamos a estrutura em back-end e front-end;
- Essa abordagem garante o desacoplamento entre estrutura e produtos finais;

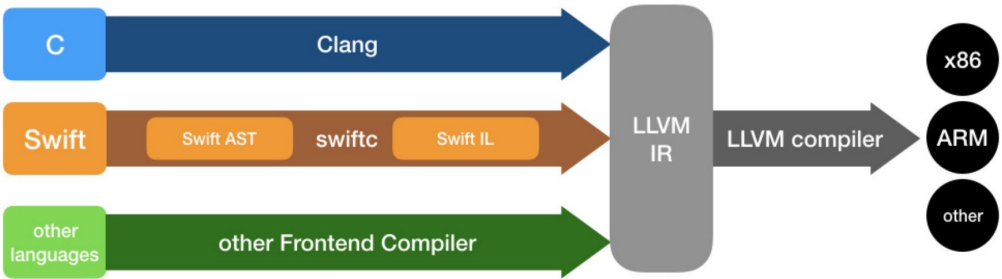


Figura 7: Exemplo de RI: LLVM-IR.

RI — Serde: Conversões automatizadas para JSON

Rust

```rust
#[derive(Serialize, Deserialize)]
struct Person {
    name: String,
    age: u8,
    phones: Vec<String>,
    address: Address,
}
#[derive(Serialize, Deserialize)]
struct Address {
    street: String,
    city: String,
}
```

JSON

```json
{
  "name": "John Doe",
  "age": 43,
  "address": {
    "street": "1st St.",
    "city": "London"
  },
  "phones": [
    "+44 1234567",
    "+44 2345678"
  ]
}
```

- `Structs` *serializáveis* podem ser usadas diretamente em *templates*;
- Suponha a variável `people: Vec<Person>`:

minijinja

```
{% for person in people %}
    {{ person.name }}, {{ person.age }} anos.
    Contato: {% for phone_nb in person.phones -%}
      {{ phone_nb }}
      {%- if not loop.last %}, {% endif -%}
    {% endfor %}
{% endfor %}
```

Fluxo de conversões

3 Metodologia

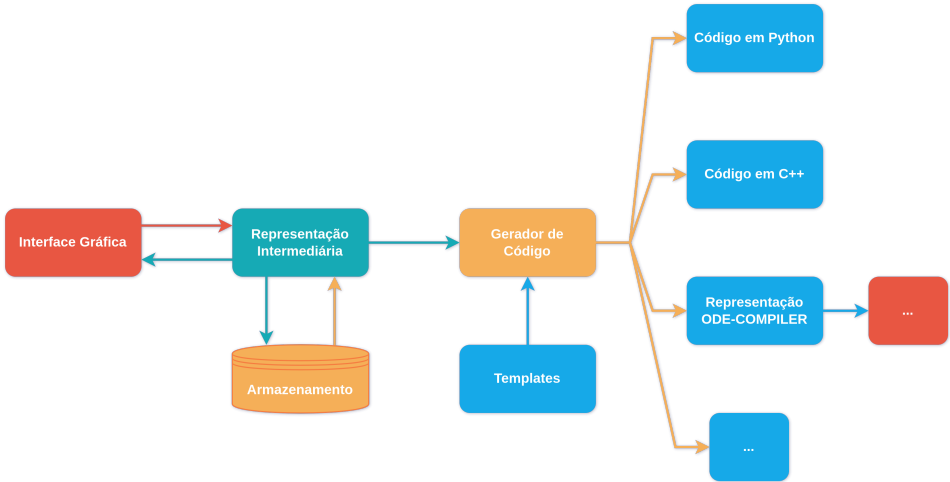


Figura 8: Fluxograma das transformações realizadas pelo software.

UFSJ

Sumário

4 Tecnologias Utilizadas

Bibliotecas e Frameworks — Links

- Programa Principal / **GUI**;
  - C++;
  - OpenGL;
  - GLFW;
  - ImGui;
  - ImNodes;
  - ImPlot;
  - SciPlot;
- Programa Transformador / **RI**;
  - Rust;
  - Serde;
  - minijinja;
  - Cbindgen;
  - Cbindgen Action;
  - GitHub Actions;
  - Catch2;

UFSJ

Sumário

5 Proximos Passos

- Iterações de design para layout da interface gráfica;
- Prototipagem da interface gráfica;
- Implementação das fundações extensíveis da interface;
- Implementação da Representação Intermediária e salvamento/carregamento de arquivos do computador;
- Testes de integração entre GUI e RI;

Futuro

5 Proximos Passos

1. Geração de código em Python;
2. Exportação dos resultados das simulações em PDF;
3. Testes do software através da construção de diversos modelos da literatura;
4. Integração com odecompiler e implementação da simulação interativa com exibição de gráficos em tempo real;
5. Definição do template e implementação da geração de código para o algoritmo de Gillespie;
6. Escrita de artigo para publicação em revista;
7. Escrita do texto final da dissertação de Mestrado.

Cronograma

5 Proximos Passos

	2023					2024
	Mar/Abr	Mai/Jun	Jul/Ago	Set/Out	Nov/Dez	Jan/Fev
1						
2						
3						
4						
5						
6						
7						

UFSJ

Desenvolvimento de um software *open source* para a modelagem e simulação do Sistema Imune Humano