Institutional Partners

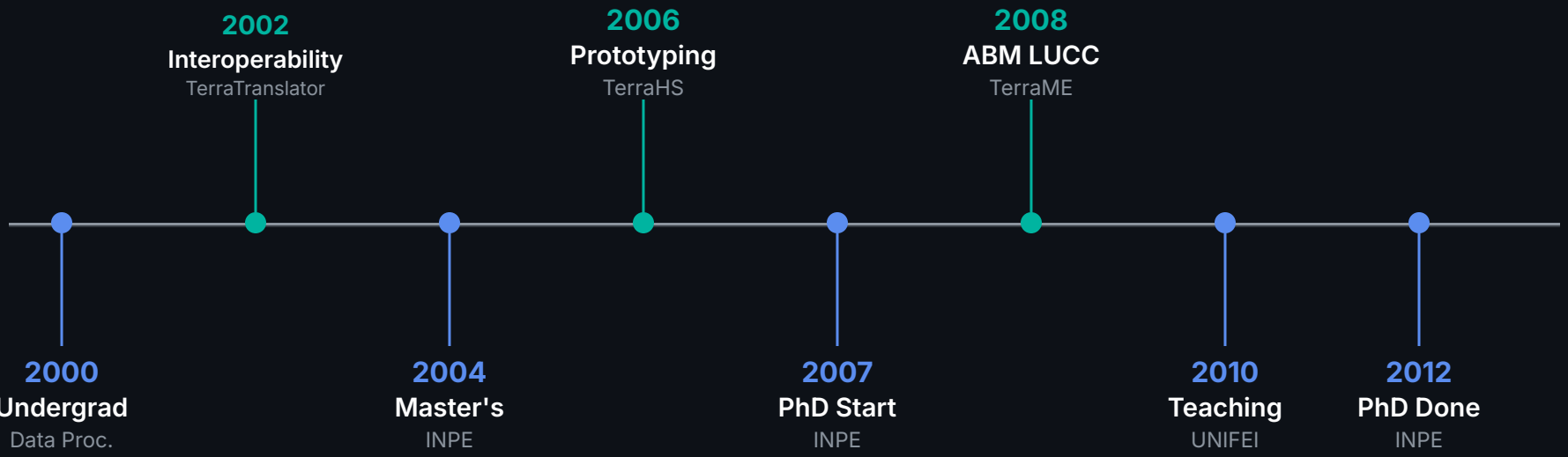
DisSModel

Building Open & Reproducible Geospatial Simulations with Python

Sérgio Costa
UFMA / LambdaGeo

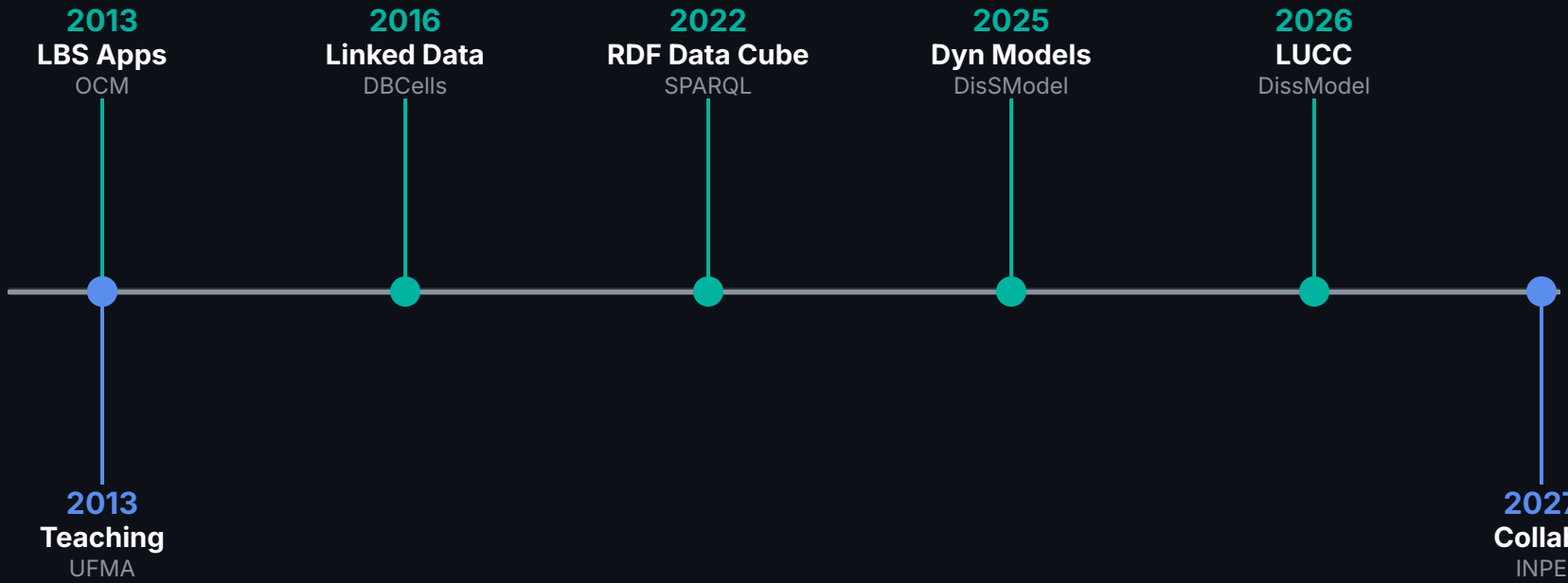
Career & Research

2000–2012 · From undergraduate to PhD — foundations in spatial modeling



Career & Research

2013–2027 · UFMA · Open Science · DisSModel · LuccME++



Research Projects Academic & Professional

Science is not reproducible.

This is a documented crisis — not an opinion.

70%

of researchers cannot reproduce experiments from other groups.

Baker (2016) · Nature

Three Root Failures

Code doesn't run

Implicit environments — works only on author's machine.

Data is missing

Input data not versioned. No data = no audit.

Parameters unknown

Configs live in ad hoc scripts or researcher's memory.

We model how land changes.

Deforestation

Urban Expansion

Coastal Dynamics

Agricultural Frontier

*Focus: **Process-based geospatial modeling** — not general-purpose ML.*

Existing tools are powerful...

…but they weren't built for reproducibility at scale.

Dinamica EGO	No support for process-based geospatial models (INPE)
TerraME	Lua Language · Last release 2020 · No cloud/STAC support
LuccME	Inherits TerraME/TerraLib constraints

DisSModel

Discrete Spatial Modeling Framework for Python

Open source · Reproducible · Cloud-native

```
pip install dissmodel
```

"Science should not need to be rewritten to go into production."

— DisSModel Principle

End-to-End Workflow

1

Create

Design model using base library

2

Package

Encapsulate in executors

3

Publish

Push to GitHub + catalog

4

Run

Same code: local or cloud

Core Pillars

Reproducibility

Executor + TOML
Docker isolation
SHA-256 audit trail

FAIR Software

Findable via Zenodo DOI
Accessible via REST API
Interoperable + Reusable

Layer Separation

Science: pure logic
Platform: APIs & infra
SimOps: operations

Three Components

Library
`dissmodel`

- Core simulation engine.
- Local dev & execution.
- `pip install dissmodel`

Platform
`dissmodel-platform`

- Cloud-native infra.
- FastAPI + Docker + Redis.
- `docker compose up`

Catalog
`dissmodel-configs`

- Single source of truth.
- TOML manifests.
- Version-controlled PRs

Same code. Runs locally and on the remote platform.

Validation through laboratory models.

Cellular Automata · System Dynamics

Game of Life

DisSModel

Cellular Automaton · Built on GeoDataFrames

game_of_life.py

```python
class GameOfLife(CellularAutomaton):

    def rule(self, idx) -> int:
        state = self.gdf.loc[idx, 'state']
        n = self.neighbor_values(idx).sum()

        if state == 1:
            return 1 if 2 <= n <= 3 else 0

        return 1 if n == 3 else 0

# Environment Initialization
gdf = vector_grid((20, 20), attrs={"state": 0})
env = Environment()
GameOfLife(gdf).initialize()
env.run()
```

Cellular Automata Simulation

Visualization of grid state at step 50. The model uses a **Queen** neighborhood (8 neighbors) and classic Conway's rules.

Explore the full code at **github.com/DisSModel**

SIR Model

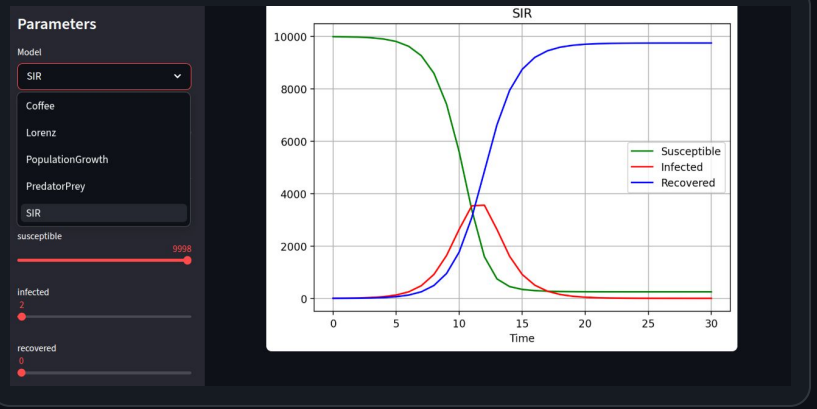
System Dynamics · Flows, Stocks, and Feedback Loops

sir_model.py

```
class SIR:
  def execute(self):
    total = S + I + R
    alpha = contacts * prob
    new_inf = I * alpha * (S/total)
    new_rec = I / duration

    S -= new_inf
    I += new_inf - new_rec
    R += new_rec

env = Environment()
SIR(9998, 2, 0, 2, 6, 0.25)
Chart(title="SIR Model")
env.run(30)
```



Dynamic Simulation Output

Visualization of Susceptible (green), Infected (red), and Recovered (blue) populations over 30 time steps. Parameters control the interaction rates and duration.

Explore the full code at **github.com/DisSModel**

Now: real-world empirical models.

From theory to application.

brmangue

Coastal dynamics model · Maranhão Coast, Brazil

🌊 FloodModel

Simulates flooding due to sea level rise scenarios based on terrain elevation and elevation rate (mm/yr).

📍 Study Area

Ilha do Maranhão · Vector spatial grid · 97,309 cells
Scientific basis: Bezerra et al. (2014)

🌿 MangroveModel

Models mangrove migration driven by tidal height and soil type dynamics over time.

⚙️ Key Parameters

Elevation rate · Tide height · Active accretion
Simulation: 2008 → 2030 (88 steps)
CRS: EPSG:31984 · Resolution: 100m

Coastal Case Study: brmangue

Mangrove migration + flooding · Maranhão Coast, Brazil

39.4×

Raster speedup
vs Vector

99.8%

Accuracy match
vs TerraME

97K

cells at scale

QGIS Plugin available — simulation inside the GIS environment, no API calls required.

DisSLUCC

Land Use & Cover Change — Continuous CLUE-type allocation

1

Demand

Magnitude of change per timestep. Pre-computed via historical CSV data (MapBiomas/LuccME).

```
DemandPreComputedValues(
annual_demand=load_csv(...)
)
```

2

Potential

Change suitability via linear regression with spatial driver variables.

```
PotentialLinearRegression(
gdf=gdf, drivers=[...]
)
```

3

Allocation

CLUE-like distribution: cells receive fractional proportions, not discrete classes.

```
AllocationClueLike(
demand, potential
)
```

Veldkamp & Fresco (1996) CLUE · Evaluation: MAE (Willmott 2005) · Dataset: LuccME AC 2008–2014 · 6,574 cells

DisSLUCC — Land Use & Cover Change

CLUE-type allocation · LuccME AC dataset (2008–2014)

🚀 Performance Speedup

3.9×

Raster speedup vs Vector

31.6 ms vs 122.5 ms per step

🎯 Accuracy Metrics

MAE = 0.002276

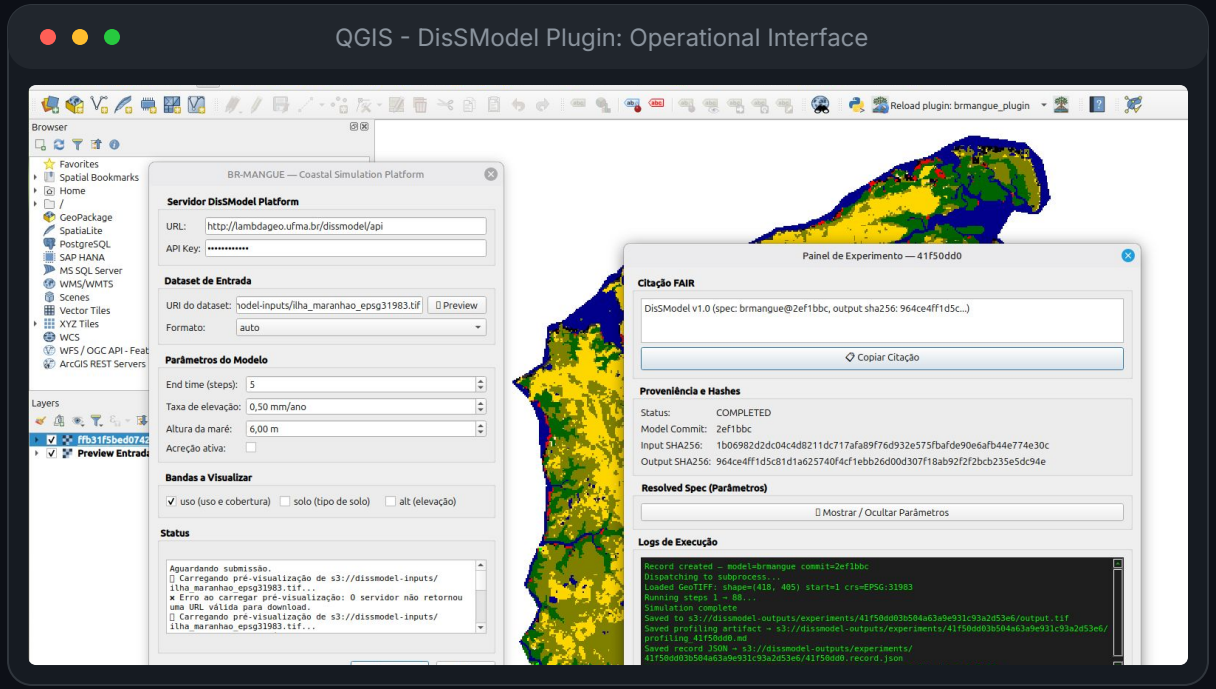
Accuracy vs TerraME

- Vector = Raster on both substrates
- Preserves full algorithmic correctness

Evaluation based on Willmott (2005) methodology

Closing the Loop

From research prototype to operational decision support.



Research

↓

Reproducibility

↓

Operational Use

DisSModel simulations executed directly inside QGIS.

What Do We Gain?

🔧 Hidden Engineering

Data orchestration runs seamlessly — infrastructure is invisible to the researcher.

🔍 Full Traceability

SHA-256 hashes + FAIR citations guarantee provenance at every simulation stage.

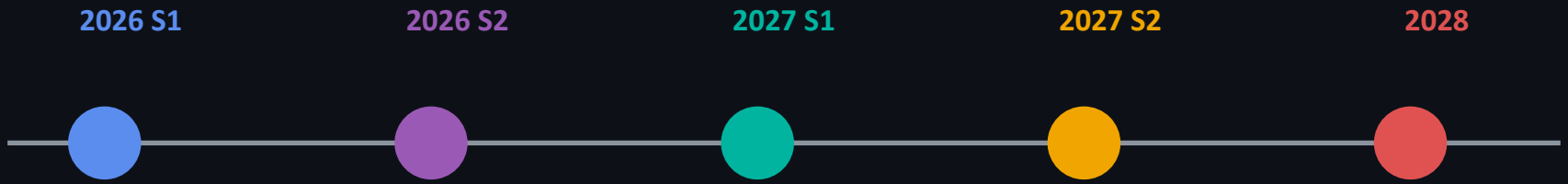
🗺️ Direct GIS Delivery

Results rendered natively in QGIS — no middleware, no extra steps.

🎯 End-to-End Focus

Researchers stay focused on science. Users stay focused on decisions.

Roadmap 2026–2028



Technology solves the how.

Governance solves the who and the where.

Infrastructure

Who maintains the platform? Server, storage, uptime.

Data Policy

Where do results live? Integration with Brazil Data Cube?

Model Catalog

Who reviews PRs? Acceptance criteria?

Community

Without shared infra, reproducibility problems remain.

DisSModel → LuccME++

2027/28 — DisSModel as the innovation laboratory for LuccME at CCST/INPE.

TerraME was the starting point — DisSModel advances as the foundation.

Reproducibility is not optional.

It is the foundation of science.

Thank you!

DisSModel — open source · open science

GitHub

`dissmodel.github.io`

Docs

`dissmodel.github.io/dissmodel/`

2005 - DPI - SenZ(s)ala

Install

```
pip install dissmodel
```

Contact

`sergio.costa@ufma.br`

> *"Science should not need to be rewritten to go into production."*
>
> — DisSModel Principle