MODIME: An Open-Source Oil Spill Simulation Platform for Coastal Governance and Environmental Resilience

Amar Seeam
Centre for African Smart Public Value Governance (C4SP)
Mauritius
amar.seeam@c4sp.org

***Abstract*—This paper presents MODIME (Modular Oil Disaster Interactive Modeling Environment), an open-source platform for simulating marine oil spills using GNOME (NOAA's General NOAA Operational Modeling Environment), Flask, Leaflet.js, and real-time meteorological data via Open-Meteo. MODIME was developed to support digital environmental governance, rapid-response simulation, and accessible decision support for vulnerable coastal regions, particularly in the African context. We describe the system architecture, workflows, case study implementation, and its significance in smart public value governance.**

***Index Terms*—GNOME, Flask, oil spill simulation, open-meteo, environmental governance, public value, open-source, coastal resilience.**

I. Introduction

Marine oil spills pose significant threats to ecological, economic, and social systems in coastal states. However, high-cost modeling tools and inaccessible datasets often hinder local governments and institutions from conducting real-time forecasting or scenario testing. MODIME addresses this gap by offering a modular and extensible web-based simulation platform designed for rapid deployment, public engagement, and open research.

II. System Architecture

A. Backend

MODIME is built with:

- **GNOME**: NOAA's oil spill modeling engine used to simulate oceanic drift and spreading.
- **Python/Flask**: Provides a lightweight API interface to launch GNOME simulations using user-defined coordinates and real-time environmental data.
- **Conda Environment**: Ensures reproducibility and easy deployment of dependencies, including 'gnome', 'numpy', and 'matplotlib'.

B. Frontend

- **Leaflet.js Map Interface**: Allows interactive spill location selection.
- **HTML/CSS/JavaScript**: Renders outputs (e.g., GIF, KMZ, NetCDF) and displays simulation logs in-browser.

This work supports exploratory research on digital tools for public value creation and environmental decision-making in the Majority World.

C. Data Inputs

- **Wind Forecasting**: Open-Meteo API is queried based on selected coordinates to generate a daily wind timeseries.
- **Oil Type**: Based on real-world ADIOS oil property files.
- **Map and Shoreline**: High-resolution .bna files define coastlines and barriers.

III. Simulation Workflow

1) User selects a spill location on the Leaflet map.
2) Coordinates are passed to the Flask backend.
3) Flask calls GNOME scripts with wind data pulled from Open-Meteo.
4) Simulation runs for a defined duration (e.g., 7 days) and outputs:
   - NetCDF (scientific model data)
   - KMZ (Google Earth visualization)
   - Animated GIF
5) Outputs are rendered on the map page and logged in real-time.

IV. Case Study: Wakashio 2020

The MV Wakashio ran aground near Mauritius on July 25, 2020, releasing over 1,000 tons of oil. MODIME reproduces key conditions from the event, including:

- Historical start date: August 6, 2020
- Wind profiles matching Open-Meteo archive
- Oil type: VLSFO and marine diesel
- Geographic constraints via 'mauritius.bna'

A. Custom 7-Day Wind Forecast Integration

To simulate realistic environmental forcing beyond Wakashio conditions, the platform incorporates a custom wind mover using 7-day forecast data from the Open-Meteo API.[1] Upon user input of a spill location (latitude and longitude), the system dynamically queries the hourly wind speed and direction at 10 meters above sea level for that coordinate.

The retrieved hourly data is programmatically averaged into daily values. The averaging process converts wind speeds from km/h to m/s, and calculates the mean wind direction per day.

[1] https://open-meteo.com/

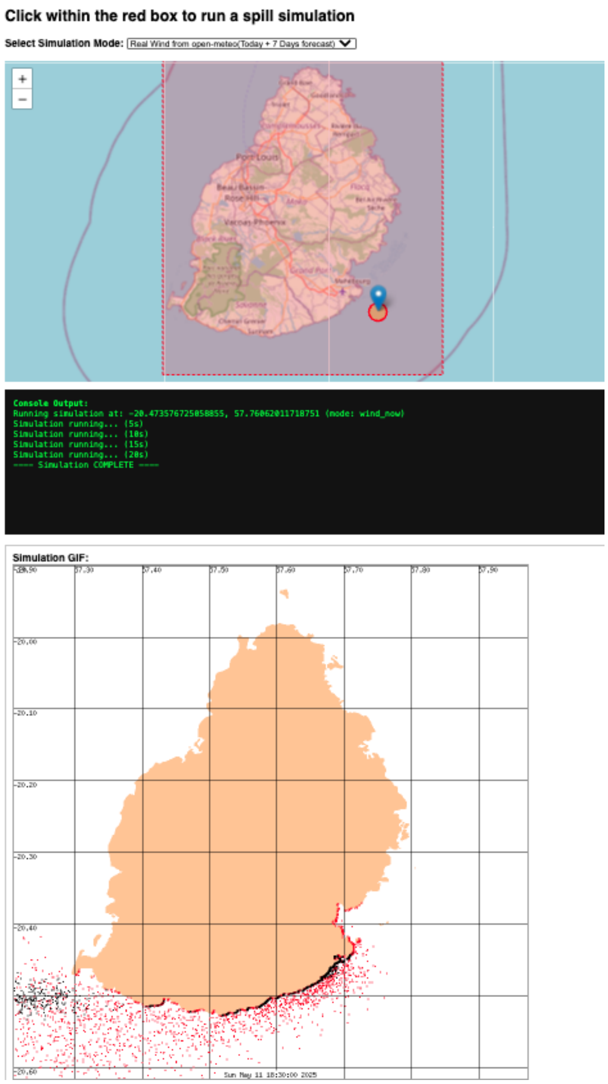

Fig. 1. Web-based user interface of the oil spill simulation platform. Users click within the red bounding box to initiate a GNOME-based forecast using real-time wind data.

The result is a daily timeseries of tuples (v, θ) representing wind velocity magnitude and direction. This timeseries is formatted according to the GNOME scripting module's expectations and passed into a `WindMover` instance as shown below:

```
wind = gs.Wind(
  timeseries=[
    (start_time + gs.days(0), (5.46, 137.7)),
    (start_time + gs.days(1), (7.96, 140.8)),
    ...
  ],
  units='m/s',
  extrapolation_is_allowed=True
)
model.movers += gs.WindMover(wind)
```

This integration allows the simulator to reflect realistic and location-specific atmospheric forcing without hardcoding environmental data, enhancing the platform's relevance for forecasting and educational use. The method ensures consistent reusability, as the wind input automatically updates on each simulation run based on the latest forecast.

V. Governance Implications

MODIME exemplifies how open-source simulation tools can support:

- Localized scenario testing for disaster preparedness
- Public awareness and participatory planning
- Policy experimentation for climate adaptation and resilience

This aligns with broader research into public value governance, digital sovereignty, and smart infrastructure.

VI. Safe Passage Simulation

An emergent application of MODIME lies in the simulation of *Safe Passage*—a computational toolset to determine and visualize optimal maritime transit corridors in relation to dynamic environmental hazards such as oil spills.

A. Rationale

In heavily trafficked Exclusive Economic Zones (EEZs) such as the Indian Ocean waters around Mauritius, the risk of navigational incidents is amplified by high vessel density, variable meteorological conditions, and limited real-time routing support. MODIME's integration of real-time wind data and GNOME-based spill forecasts enables stakeholders to test alternative maritime passage routes during unfolding marine incidents.

B. Methodology

Using user-defined spill coordinates and a 7-day forward wind forecast from the Open-Meteo API, MODIME generates spatial outputs such as beached oil overlays, surface drift patterns, and NetCDF current fields. These are compared against typical maritime routes to identify:

- Potential intersections between planned vessel paths and contamination zones
- Temporal windows for safe transit before the spill reaches specific coastal regions
- Proactive re-routing options to mitigate secondary environmental or economic impact

C. Application Potential

This feature extends the simulation framework beyond post-incident analysis toward real-time decision-support systems. In collaboration with port authorities or marine traffic surveillance systems, MODIME could interface with live AIS (Automatic Identification System) feeds to flag high-risk trajectories and suggest alternate shipping lanes dynamically.

D. Contribution to Governance

This extension reinforces the platform's alignment with Smart Public Value Governance by:

- Supporting resilience-focused maritime governance
- Bridging data models with operational maritime safety decisions
- Enhancing the digital sovereignty of small island states in managing EEZs

VII. Availability

The source code, simulation scripts, and data files are available on GitHub:
https://github.com/amarseeam/modime

VIII. Conclusion and Future Work

MODIME serves as a replicable model for digital environmental decision tools. Planned future extensions include:

- Tidal and current data integration (HYCOM/CMEMS)
- UI/UX for community forecasting
- Automated report generation