SRC: Computational Steering for Geosimulations

Ashwin Shashidharan*
North Carolina State University
Raleigh, NC 27695
ashdharan@ncsu.edu

ABSTRACT

Geosimulations using computer simulation models provide researchers an effective way to study complex geographic phenomena and their outcomes. These simulations allow for scenario based exploration by capturing spatial and temporal relationships between various geographic processes in a region. However, current approaches to geosimulation limit manipulating model input and exploring alternative scenarios by controlling the simulation model at runtime. This paper proposes a computational steering system for geosimulation models and presents a prototype, *tFUTURES*, developed for the FUTURES Urban Growth Model (UGM). By allowing tangible inputs and implementing mechanisms to control model execution, this system solves the problem of lack of user-interactivity experienced at runtime. We develop a web interface and leverage the WMS, WFS-t and WPS OGC services to help visualize, modify and execute geosimulations. We define new *steering controls* within this interface and implement *application checkpointing*, allowing a user to provide new *steering input* and execute *steering actions* that can *pause*, *advance* or *rollback* a geosimulation and display the model outcomes in near real-time.

CCS Concepts

•**General and reference** → *Design;* •**Computing methodologies** → *Real-time simulation; Interactive simulation; Scientific visualization;* •**Computer systems organization** → *Special purpose systems;*

Keywords

Geosimulation, Computational steering, Visualization

1. INTRODUCTION

Modeling and simulation has revolutionized many scientific and engineering fields in the past two decades. In recent years, geosimulation has emerged at the intersection of Geographic Information Science, Complex Systems Theory and Computer Science. Geosimulations [1], where real-world processes are modeled and studied over time, have been successfully applied in urban studies, epidemiology, land use and land cover changes, and climate change studies. Using geosimulations, "what if" scenarios can be studied to understand potential impacts of geographic events. However, such scenario analyses rely on static inputs prepared beforehand by GI scientists.

Computational steering [3, 4] is a mechanism that supports interactivity in simulations while they are in progress. Specifically, it allows for manipulation of the internal state of a simulation and its inputs during execution. For instance, in a UGM geosimulation, computational steering mechanisms could be used to specify new zoning regulations and transportation networks to an in-progress simulation. Further, the ability to visualize the impact on development patterns in real-time, could be used to tweak the inputs for subsequent time-steps or in retrospect. Such interactivity helps improve the quality of simulations, allows on-the-fly "what if" scenarios, and improves computational efficiency. However, little work has been carried out to integrate computational steering and geosimulations with visualization support [5]. In our system, tFUTURES, we attempt to bridge this gap by supporting computational steering for geosimulations from a javascript enabled web browser. Finally, we enable *application checkpointing* in geosimulations and support *steering actions* that can *pause*, *advance* or *rollback* a geosimulation from any such browser.

2. tFUTURES SYSTEM

The tFUTURES system is designed to support practitioners and users who wish to simulate and understand urbanization under varying human decision scenarios. It supports tactile input to be provided to the FUTURES UGM [2] and the analysis of their outcomes in real-time. The tFUTURES computational steering system comprises of three components as shown in Fig. 1, namely (i) Monitoring server; (ii) Steering client; and (iii) Visualization service.

2.1 Monitoring Server

The monitoring server acts as an interface between the *visualization service* and the *steering client* in the system. It receives WPS requests generated by the *visualization service* and forwards them to the *steering client* in the UGM.

*The author acknowledges the support and guidance received from his advisors Dr. Ranga Raju Vatsavai and Dr. Ross K. Meentemeyer.

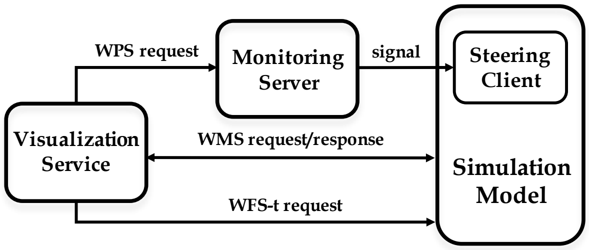

Figure 1: tFUTURES System Architecture.

2.1.1 Capabilities

The monitoring server is setup and initialized as a signal handler hub capable of routing *steering actions* to the UGM. It is aware of the *steering controls* available to a user at run-time and maps *steering actions* to specific signal handlers in the steering client.

2.1.2 Mechanism

Each *steering action* is an event that generates a particular type of signal. When a *steering control* is selected by a user, the monitoring server interrupts the current flow of UGM execution, and delivers a *steering action* to be executed by the UGM.

2.2 Steering Client

The steering client augments the UGM code to handle *steering actions* and user-defined *steering input*. It defines signal handling routines to service the *steering actions* forwarded by the *monitoring server*. Specifically, the steering client embedded in a UGM allows (i) modifying UGM simulation state; (ii) altering the control flow of the UGM simulation at runtime; and (iii) periodically checkpointing simulation state to enable *rollback* of the UGM simulation.

2.2.1 Capabilities

The steering client implements handling routines that asynchronously process signals delivered to the UGM during execution. The *steering controls* shown in Fig. 2a are defined as follows: (i) **skipPrev:** rollback the simulation by a single time-step; (ii) **restart:** reset the *steering input* and restart the existing simulation run; (iii) **play:** run the simulation from current state till completion; (iv) **skipNext:** advance the simulation by a single time-step; and (v) **pause:** pause the simulation at the end of the current time-step.

2.2.2 Mechanism

When a signal is received by the steering client, the specified *steering action* for that signal is taken. A handler function in the steering client implements this action for the UGM. In the FUTURES UGM, these handlers are predefined in the steering client as part of program annotation.

2.3 Visualization Service

The visualization component is a web service accessible from a web browser on a user's local machine. It provides the end-user with (i) web controls for interacting with the simulation; and (ii) on-line visualization of the simulation results. We use the OpenLayers JavaScript library for online map visualization, and develop the *steering controls* as web widgets using HTML, CSS and JavaScript.

2.3.1 Capabilities

The web interface provides dynamic rendering of output raster maps from the simulation. It also supports drawing vector data as input to the simulation and rendering them from within the web browser. A user can select controls from the "Steering Controls Menu" or the "Map Controls Menu" from within this web interface (Fig. 2).

(a) Steering Controls (b) Map Controls

Figure 2: Map and Steering Controls Menu.

2.3.2 Mechanism

To experiment with various development scenarios, a user defines patterns using the *map controls* (Fig. 2b). These map controls trigger WFS-t requests directly modifying the UGM input. The *steering controls* provide the ability to run scenarios based on these inputs in time-steps as defined by the UGM. At the end of every *steering action*, the resulting urbanization map is refreshed in the browser. The visualization component thus, acts as an endpoint that accepts user input and displays simulation outputs in tFUTURES.

3. CONCLUSIONS

In this paper, we show that computational steering capabilities can be easily extended to geosimulations with a small set of interacting components and minimal changes to legacy model code. At a bare minimum, the set of interacting components must include 1) a visualization interface with *steering controls*; 2) a monitoring server to intercept and relay *steering actions* to the simulation; and 3) a steering client embedded in the legacy simulation code. Finally, by intertwining user interactions with geosimulations, we empower practitioners and novice users to dynamically vary model inputs at runtime and produce desired simulation results.

4. REFERENCES

[1] I. Benenson and P. M. Torrens. *Geosimulation: Automata-based modeling of urban phenomena*. John Wiley and Sons, 2004.

[2] R. K. Meentemeyer, W. Tang, M. A. Dorning, J. B. Vogler, N. J. Cunniffe, and D. A. Shoemaker. FUTURES: multilevel simulations of emerging urban-rural landscape structure using a stochastic patch-growing algorithm. *Annals of the Association of American Geographers*, 103(4):785–807, 2013.

[3] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of computational steering environments. *Future generation computer systems*, 15(1):119–129, 1999.

[4] R. Van Liere, J. D. Mulder, and J. J. Van Wijk. Computational steering. In *International Conference on High-Performance Computing and Networking*, pages 696–702. Springer, 1996.

[5] R. R. Vatsavai and R. K. Meentemeyer. Lightning talk: Intertwining computational steering with geosimulations. http://ucgis.org/sites/default/files/document-sharing-form-files/209/Vatsavai_.pdf. University Consortium for Geographic Information Science (UCGIS), 2016.