ABSTRACT

**MUSE: A parallel Agent-based Simulation Environment**

By Meseret R. Gebre

The use of agent-based modeling and simulation-based analysis is rapidly gaining importance in many areas. Realizing the advantages of simulation-based methodologies requires the use of a software environment that is conducive for modeling, simulation, and analysis. Furthermore, parallel simulation methods must be employed to reduce the time for simulation, particularly for  
large problems, to enable analysis in reasonable timeframes. Unfortunately, effective and efficient parallel, agent-based simulation software is not available as of this proposal. Accordingly, this thesis covers the development of a general purpose agent-based, parallel simulation environment called MUSE (Miami University Simulation Environment). MUSE, provides an Application Program Interface (API) for agent-based modeling and a framework for parallel simulation. The API was developed in C++ using its object oriented features. The core parallel simulation capabilities of MUSE were realized using the Time Warp synchronization methodology and the Message Passing Interface (MPI). We envision MUSE to be a scalable and efficient simulation environment for a broad spectrum of models. Accordingly, the research demonstrates the qualitative advantages of MUSE by using several well-defined criteria. In addition, the investigations include empirical analysis to quantitatively assess the efficiency and scalability of MUSE using suitable benchmark applications.

**MUSE: A parallel Agent-based Simulation Environment**

A Thesis

Submitted to the Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Masters of Computer Science

Department of Computer Science and System Analysis

by

Meseret R. Gebre

Miami University

Oxford, Ohio

2009

Advisor\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. DHANAJAI M. RAO

Reader\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. MUFIT OZDEN

Reader\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. LUKASZ OPYRCHAL

**TABLE OF CONTENTS**

[1. Introduction 6](#_Toc223819200)

[2. Background and Related Work 7](#_Toc223819201)

[2.1 Message Passing Interface (MPI) 7](#_Toc223819202)

[2.2 Choosing the Programming Language (C++) 7](#_Toc223819203)

[2.3 Synchronization Methods 7](#_Toc223819204)

[2.4 Parallel Non-Agent based simulation frameworks 7](#_Toc223819205)

[2.4.1 WRAPED 7](#_Toc223819206)

[2.4.2 GTW 7](#_Toc223819207)

[2.4.3 Parsec 7](#_Toc223819208)

[2.5 Non-parallel Agent based Simulation frameworks 7](#_Toc223819209)

[2.5.1 NetLogo 7](#_Toc223819210)

[2.5.2 SWARM Objective-C 7](#_Toc223819211)

[2.5.3 SWARM Java 7](#_Toc223819212)

[2.5.4 Repast 7](#_Toc223819213)

[2.5.5 MASON 7](#_Toc223819214)

[3. Details of MUSE design 8](#_Toc223819215)

[4. Benchmarking 8](#_Toc223819216)

[5. Conclusion and Future Work 8](#_Toc223819217)

[References 9](#_Toc223819218)

**LIST OF FIGURES**

**Acknowledgements**

# Introduction

Agent-based models have been used since the mid-1990s to solve a variety of business, technology, and medical problems (www.wikipedia.org). The following are example of such applications:

* Supply chain optimization
* The spread of epidemics
* Threat of bio-warfare
* Modeling of consumer behavior
* Social network effects
* Workforce management

The examples above are important topics and the amount of time required to reach valuable solutions can make the difference between success and failure or even life and death. There are five main Agent-based simulation frameworks that are in use. NetLogo, MASON, Repast, Swarm (Objective-C), and Swarm (Java). None of these frameworks utilize parallelism and most with the exception of Swarm (Objective-C) and NetLogo are written in Java. NetLogo uses its own language that is at a very high-level and is menu driven, which several researchers believe to be very restrictive.

From literature surveys (Railsback and Lytinen), the following key issues with agent-based modeling and simulation software were identified:

1. Platform complexity is a major concern.
   1. Very large API can be intimidating to users.
2. Is Java the right language?
   1. Syntax and object typing.
3. Error checking and garbage collection
   1. Error must be easy to identify and troubleshoot.
   2. Memory management is going to be a tough concept to beginners (coders). Need a way to minimize memory leaks from within the MUSE kernel.
4. Availability of development tools
   1. List the types of development tools you are referring to (editor, compiler, linker, debugger, etc.)
   2. Many tools are readily available for Java. This may be a battle that cannot be won. However it can be minimized, C++ is also has many development tools.

We propose developing a new framework that will be written in C++. MUSE (Miami University Simulation Environment) will be a parallel agent-based simulation framework and as of the writing of this proposal, is the first of its kind. MUSE’s main advantage will be its speedup, which will be derived from the use of parallelism. We used the recommendations from (Railsback and Lytinen) to help shape the API of this framework. The frameworks mentioned above were all ranked based on well-defined criteria. These include:

* Complete documentation of classes and methods, with examples.
* Follow standard terminology to ease effective use of API for modeling.
  + Most users developing simulations are novices and have little knowledge of programming.
* Provide tools for generating statistical output.
* Provide tools for setting up and executing simulation experiments.

In concordance with these criteria, we propose to develop MUSE and use the aforementioned criteria to measure its qualitative characteristics. In addition, we propose to develop empirical tests to quantitatively assess the parallel simulation characteristics such as: speedup, scalability, and efficiency. Section 2 presents background some of the closely related works. Section 3 describes in detail the design process for MUSE. Section 4 describes the benchmarks we used to insure proper quality and quantitative. We all present the results obtained when evaluating MUSE. Section 5 concludes the thesis and discusses future works.

# Background and Related Work

## Message Passing Interface (MPI)

## 2.2 Choosing the Programming Language (C++)

## 2.3 Synchronization Methods

## 2.4 Parallel Non-Agent based simulation frameworks

### 2.4.1 WRAPED

### 2.4.2 GTW

### 2.4.3 Parsec

## 2.5 Non-parallel Agent based Simulation frameworks

### 2.5.1 NetLogo

### 2.5.2 SWARM Objective-C

### 2.5.3 SWARM Java

### 2.5.4 Repast

### 2.5.5 MASON

# Details of MUSE design

# Benchmarking

# Conclusion and Future Work

# References

Bailey, M. and L. Snyder. A Model for Comparing Synchronization Strategies for Parallel . 1989.

Bryant, R. E. "Simulation on a distributed system." In Proc. Of the 16th Design automation Conference. 1979. 544-552.

Chandy, K.M. and J. Misra. "Distributed simulation: a case study in design and verification of distributed programs." IEEE Transactions on Software Engineering. September 1979. 440-452.

Chen, Gilbert and Boleslaw K. Szymanski. "DSIM: SCALING TIME WARP TO 1,033 PROCESSORS ." Proceedings of the 37th conference on Winter simulation. Orlando, Florida, December 2005. 346-355.

Cleary, John G. and Jya-Jang Tsai. "Conservative Parallel Simulation of ATM Networks." SIGSIM. July 1996. 30-38.

Collier, Nick. RePast: An Extensible Framework for Agent Simulation. Chicago: University of Chicago, n.d.

Das, D., et al. "GTW: A Time Warp system for shared memory multiprocessors." In Proceedings of the 26th Winter Simulation Conference. 1994. 1332-1339.

Das, Samir R. and Richard M. Fujimoto. "Adaptive memory management and optimism control in time warp." ACM Transactions on Modeling and Computer Simulation (TOMACS). April 1997. 239-271.

Fredman, M. L. and R. E. Tarjan. "Fibonacci heaps and their uses in improved network optimization algorithms." ACM (July 1987): 596-615.

Grama, Ananth, et al. Introduction to Parallel Computing (Second Edition). Addison-Wesley, n.d.

Jefferson, David R. "Virtual Time." ACM Transactions on Programming Languages and Systems. UCLA, July 1985. 404-425.

Kühl, Dietmar. Dietmar Kühl's Homepage. 10 January 2009 <http://www.dietmar-kuehl.de/cxxrt/heaps.tar.gz>.

LIN, YI-BING and EDWARD D. LAZOWSKA. "A study of time warp rollback mechanisms." ACM Transactions on Modeling and Computer Simulation (TOMACS). January 1991. 51-72.

Luke, Sean, et al. "MASON: A New Multi-Agent Simulation Toolkit." In Proceedings of the Eighth Annual Swarm Users/Researchers Conference. 2004.

Matsumoto, Makoto and Takuji Nishimura. "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator." ACM Transactions on Modeling and Computer Simulations: Special Issue on Uniform Random Number Generation. 1998. 3-30.

Mattern, Friedemann. "Efficient Algorithms for Distributed Snapshots and Global Virtual Time Approximation." Jornal of Parallel and Distributed Computing. 1993.

Minson, R. and G. K. Theodoropoulos. "Distributing RePast Agent-Based Simulations with HLA." European Simulation Interoperability Workshop. 2004.

MPI-2: Extensions to the Message-Passing Interface. 18 October 2008 <http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html>.

N., Minar, et al. "The Swarm simulation system: A toolkit for building multi-agent simulations." Santa Fe (NM): Santa Fe Institute, 1996.

R. Bagrodia, R. Meyerr, et al. "PARSEC: A Parallel Simulation Environment for Complex System." Technical report. 1997.

Radhakrishnan, R., Martin, D. E., Chetlur, M., Rao, D. M., and Wilsey, P. A. "An Object-Oriented Time Warp Simulation Kernel." In Proceedings of the Second international Symposium on Computing in Object-Oriented Parallel Environments. 1998. 13-23.

Railsback, Steven F. and Steven L. Lytinen. "Agent-based Simulation Platforms:Review and Development Recommendations." Simulation (A journal from The Society for Modeling and Simnulation International). 2006. 609-623.

Steinrnan, Jeff S. "BREATHING TIME WARP." Proceedings of the seventh workshop on Parallel and distributed simulation. San Diego, 1993. 109-118.

Tisue, Seth and Uri Wilensky. "NetLogo: A Simple Environment for Modeling Complexity." International Conference on Complex Systems . 2004.

www.wikipedia.org. 3 November 2008 <http://en.wikipedia.org/wiki/Agent\_based\_modeling>.