# THE DESIGN AND IMPLEMENTATION OF A MULTIPLE RESOLUTION MODELING FRAMEWORK WITH APPLICATIONS TO POPULATION MODELING

by

KEVIN ANTHONY GLASS

A DISSERTATION

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2003

"The Design and Implementation of a Multiple Resolution Modeling Framework with Applications to Population Modeling," a dissertation prepared by Kevin Anthony Glass in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science. This dissertation has been approved and accepted by:

Dr. Kent A. Stevens, Chair of the Examining Committee

MARCH 4, 2003

Date

Commitee in charge:

Dr. Kent A. Stevens, Chair
Dr. Sarah Douglas
Dr. Allen D. Malony
Dr. Daniel Udovic

Accepted by:

Dean of the Graduate School

An Abstract of the Dissertation of

Kevin Anthony Glass for the degree of Doctor of Philosophy
in the Department of Computer and Information Science
to be taken March 2003

Title: THE DESIGN AND IMPLEMENTATION OF A MULTIPLE RESOLUTION MODELING FRAMEWORK WITH APPLICATIONS TO POPULATION MODELING

Approved: ______________________________
Dr. Kent A. Stevens, Chair

Ecologists have relied increasingly on computer simulation to model population dynamics. New models that focus on the behavior and interactions of individuals, rather than aggregate measures of population dynamics, have increased in use. With the increase in computationally intensive models, however, ecologists have been forced to find compromises between the accuracy of the new techniques and computational efficiency of the old techniques. Multiple-resolution modeling is one proposed method to solve this problem. A multiple-resolution model uses the accurate, but computationally intensive, models only when necessary; otherwise, it uses the less accurate, but computationally efficient, models.

This dissertation examines issues related to the development of a multiple-resolution modeling framework. Developing this framework required the formulation of a general, population modeling schema on which to build the multiple-resolution modeling framework and the formulation of a general structure for multiple-resolution modeling. Software engineering techniques were applied in the development of the framework.

A set of multiple resolution models were developed and tested. The results of these tests do not verify the generality of multiple-resolution modeling; rather, they suggest that the framework can produce correct multiple-resolution models.

# CURRICULUM VITA

NAME OF AUTHOR: Kevin Anthony Glass

PLACE OF BIRTH: Oakland, CA

DATE OF BIRTH: July 15, 1958

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon
California State University, Hayward
University of California, Berkeley

DEGREES AWARDED:

Doctor of Philosophy in Computer and Information Science, 2003, University of Oregon
Master of Science in Computer and Information Science, 1995, California State University, Hayward
Bachelor of Arts in Physics, 1980, University of California, Berkeley

AREAS OF SPECIAL INTEREST:

Computational Ecology
Modeling and Simulation

PROFESSIONAL EXPERIENCE:

Graduate Teaching Fellow, Department of Computer and Information Science, University of Oregon, Eugene, OR, 2000-2002

Graduate Research Fellow, Department of Computer and Information Science, University of Oregon, Eugene, OR, 1995-1996

Satellite Operations Engineer, Lockheed Missles and Space, Sunnyvale, CA, 1981-1992

AWARDS AND HONORS:

U. S. Department of Energy, Compuational Science Graduate Fellowship, 1996-2000

PUBLICATIONS:

K. A. Glass, K. A. Stevens, "A Discrete Event Framework for Spatial and Hierarchical Ecological Models," IASTED International Conference Applied Modelling and Simulation, September, 1999.

K. A. Glass, M. Livingston, J. Conery, "Distributed Simulation of Spatially Explicit Ecological Models," PADS '97, June, 1997

# ACKNOWLEDGEMENTS

At the end of any adventure it is always appropriate to reflect on the path just taken and to thank those whose help guided the way. This particular adventure would not have been possible without the advice and guidance of Kent Stevens and I am indebted to his patience and forbearance throughout this process. Also, without Sarah Douglas' insight into the technical details of simulation analysis and procedural details of the doctoral degree, this dissertation would not have come to fruition.

This work was supported by the Computational Science Graduate Fellowship Program of the Office of Scientific Computing and Office of Defense Programs in the Department of Energy under contract DE-FG02-97ER25308. I would especially like to thank Barbara Helland and Shellie Hosch of the CSGF for their assistance.

In addition to the software I developed for my research, I also relied heavily on the use of STELLA, from High Performance Systems Inc., SPSS, from SPSS Incorporated, Framemaker, from Adobe Systems Incorporated and TAU, from the University of Oregon.

During the course of earning this degree, I have had the opportunity to work with some fine people. In particular, the staff working in the front office of the CIS department was always a great help, especially during the most trying times. I would like to thank Betty Lockwood, Jan Saunders and Star Holmberg, all of whom served as Graduate Coordinators for the department and Cheri Smith, who served as the Undergraduate Coordinator while I performed teaching duties. In addition to the office staff, I would like

to thank the system administrators, Paul Bloch and Lauradel Collins who kept the machines running. It is easy to underestimate or forget the contribution these people have made to my dissertation and my life, but it would be a terrible omission.

My experience at the University of Oregon was made richer by the intellectual and social interactions I have had with my fellow students. A detailed list is too large to cite in such a short space, but some mention of specific people is necessary. I would specifically like to thank my friend, Chris Harrop, who acted as a sounding board for many of my musings and many of my ventings. No one could ask for a better friend. I would also like to thank Sue Ratner and Kevin Higgins for the ecological and mathematical insights they have shared with me. Others, including John Fiskio-Lasseter, Sameer Shende, Chris GauthierDickey, and Odile Wolf have helped me through this process.

The experience I have had at the University was not limited to campus life. Two groups of people have contributed to my sanity and focus throughout this process. Martial Arts America has been a tremendous asset. I would specifically like to thank Masters Gary Martin and Kathy Butler, who have been inspirations and models to follow. The congregation at Springfield Lutheran Church, especially Pastor Zane Wilson, Mark and Jeanie Walker and Brent and Tracy Hudson, have helped to remind me that this stay is temporary and that it is in service, not self-interest, that contributions are measured. I thank God for these people and their faithfulness.

Mostly, I would like to thank my family, my sons, David and Michael, and my wife, Donna. The boys have suffered through this experience because of a partially absent father. Well, guys, now its time to play a little basketball. I will not belabor this statement with trivialities about my wife being my friend, lover, inspiration, and so on. Simply put, Donna is the reason my heart makes its next beat, my lungs take their next breath and my life is blessed. Her incredible faith in me has made this work possible.

# TABLE OF CONTENTS

Chapter Page

Chapter Page

Chapter Page

# LIST OF TABLES

Table Page

# LIST OF FIGURES

Figure Page

# CHAPTER I

# MULTIPLE RESOLUTION MODELING IN POPULATION ECOLOGY

## 1.1 Introduction

Traditionally, ecologists have modeled population dynamics with finite difference equations or differential equations using aggregated measurements of population characteristics. Huston, DeAngelis and Post (1988), however, argued that the traditional models violate fundamental precepts of organism behavior. Aggregating the properties of individuals into a single variable causes a loss of information regarding individuals and their impact on population dynamics. This led Huston et al. (1988) to argue for an individual-oriented modeling approach in which populations are represented as a set of discrete individuals. These individual models implement a set of predefined rules to represent the behavior and interactions of real individuals.

An advantage of the systems approach is that IOMs use the observations made by ecologists to represent population dynamics (Judson, 1994; McGlade, 1999). The resulting models are assumed to be easier to understand, as they are expressed in ecological terms. Moreover, the model parameters and simulation output are based on the data collected by ecologists (Huston, et al., 1988), so the modeler can compare simulation results with experimental observations of the population. These benefits led Huston et al.

and others to the belief that IOMs would replace traditional, analytic models as the dominant population modeling paradigm (Huston, et al., 1988).

By 1992, the optimism regarding the future of IOMs gave way to the understanding that IOMs have advantages and disadvantages relative to their traditional counterparts (DeAngelis and Rose, 1992). This understanding led DeAngelis and Rose to develop a set of criteria for selecting a modeling approach based on the characteristics of a problem. Though these criteria are an important aspect of this dissertation, the conclusion reached by DeAngelis and Rose is of more immediate interest, namely, the authors "envision many situations in which both approaches [system theoretic and analytic] will be used together" (1992). This vision is the basis of this dissertation.

The vision expressed by DeAngelis and Rose can be restated as a desire for a modeling mechanism that combines two models of a single system, at different levels of resolution, into a single representation. Coupling single-resolution models to form a multiple-resolution model (MRM) requires a specialized approach to model development at both the design and implementation levels. Computer scientists have studied techniques for multiple-resolution modeling since the late 1980's (Davis, 1993), however, research in this field has failed to produce a general system for supporting the development of multiple-resolution models (Reynolds, Natrajan and Srinavasan, 1997). The problem with creating such a system for ecologists is exacerbated by the lack of a general population modeling framework. That is, ecologists have not adopted a general system for describing or implementing IOMs and such a system is necessary as a basis for multiple-resolution modeling. This dissertation describes an investigation into the problems of developing of a general population modeling framework and a multiple-resolution modeling extension to the framework.

## 1.2 Population Modeling and Computer Simulation

The resolution at which a population should be modeled is, according to Levin (1992) and Pascual and Levin (1999), a central problem in ecology. An individual's behavior is influenced by its spatial, temporal and organizational experience, and these experiences vary relative to the resolution, or scale, of the experience. For example, an individual's response to a threat depends both on its recent and long-term experiences. Therefore, the effect of the threat is based on both fine and coarse-scaled temporal experiences. In modeling the individual's response to the threat, the modeler will include a representation of one or the other scales of experience.

A system should be represented at a scale consistent with the properties under investigation. For example, a small population should be represented as a set of interacting individuals rather than a single entity according to DeAngelis and Rose (1992). This choice is reasonable because the influence of random processes is greater on smaller populations than it is on larger populations and IOMs represent randomness better than aggregate models. On the other hand, a large population should be represented by an aggregate model because, according to the law of large numbers, a large population will exhibit average case behavior. If, however, the population fluctuates between small and large population sizes, the representation should accommodate both scales. As the aggregate model cannot adequately express the behavior of a small population, the modeler must use the more accurate of the two, that is, the population should be represented by an IOM.

The requirement, that a modeler choose one scale over another, is a limitation of simulation technology and not the desire of the modelers. Besides the comment by DeAngelis and Rose, several ecologists have argued for the use of multiple resolution models (Levin, 1992; Hyman, McAninch & DeAngelis, 1991; De Vasconcelos, Goncalves, Catry, Paul & Barros, 2002; Maxwell, 2000). Multiple-resolution models are seen as a necessity because the complex temporal, spatial and social interactions encountered by organisms can take place across several scales. Nevertheless, the current state of simulation technology does not support a general framework for multiple-resolution modeling (Reynolds, Natrajan & Srinavasan, 1997).

The problems associated with multiple resolution modeling are well known in computer science (Davis 1993, Davis & Hillestad 1993, Reynolds, et al., 1997). Researchers have developed MRMs for military simulations and, in the process, they have identified fundamental issues for creating valid MRMs. An MRM, as described by Davis and Hillestad (1993), consists of two models that represent a system at two resolutions and a set of transformations that map the state of one model to the state of the second. Figure 1, which is adopted from Davis (1993), shows the temporal and state behavior between a low-resolution model (LRM) and a high-resolution model (HRM). At time $t$, the state of HRM, $Q_H$, must be consistent with the state of the LRM, $Q_L$ and vice-versa. As the two

$$\begin{tikzcd}
\boxed{\text{LRM: } t_n} \arrow[r, "\delta_{LRM}(Q_{LRM}, X_{LRM})"] \arrow[d, shift right=2, "f^{-1}(Q_{LRM})"'] & \boxed{\text{LRM: } t_{n+1}} \arrow[d, shift right=2, "f^{-1}(Q_{LRM})"'] \\
\boxed{\text{HRM: } t_n} \arrow[r, "\delta_{HRM}(Q_{HRM}, X_{HRM})"] \arrow[u, shift right=2, "f(Q_{HRM})"'] & \boxed{\text{HRM: } t_{n+1}} \arrow[u, shift right=2, "f(Q_{HRM})"']
\end{tikzcd}$$

Figure 1. Davis-Hillestad Diagram adapted from Davis (1993, p. 5).

models have different representations of state, imposing consistency between the models requires a translation from the state of one model into the state of the other. That is, there must be an invertible function, $f$, such that $f(Q_{HRM}) = Q_{LRM}$ and $f^{-1}(Q_{LRM}) = Q_{HRM}$. Moreover, if the states of both models are consistent at time $t$ and one model transitions its state at time $t+1$, the state of both models must be consistent at time $t+1$. For example, if the HRM changes its state, using $\delta_{HRM}(Q_{HRM,t}) = Q_{HRM,t+1}$, then the LRM must have the state $f(Q_{HRM,\ t+1})$. Though this description does cover the essential problem with multiple-resolution modeling, it is incomplete. The Davis and Hillestad diagram only considers synchronized models and it does not consider interacting. Populations, however, are asynchronous systems that include interacting between populations and the environment.

Recent work by Reynolds et al. (1997) and Natrajan, Reynolds and Srinavasan (1997) offers a set of empirically derived rules for maintaining consistency between interacting MRMs. The rules proposed by Reynolds et al. were designed to coordinate interactions between MRM, by describing a model coupling mechanism that maps state data between models. The rules do not, however, provide a general framework for multiple-resolution modeling. Consider an MRM consisting of an IOM and an aggregate model. The rules proposed by Reynolds et al. impose consistency between the two models, but the system described by Reynolds et al. (1997) does not impose consistency between the IOM and the individual models belonging to the IOM.

This dissertation will describe the development and implementation of a multiple-resolution population modeling framework designed to support general multiple-resolution modeling. Conceptually, the framework unites two models of a system into a single representation that coordinates the behavior of all aspects of its constituent single-resolution models. The framework is based on a generally accepted theory of modeling and simulation (Zeigler, 1976) and the multiple-resolution modeling component is developed from the theory. The result of the research performed for this dissertation is the development of the Spatio-Temporal Ecological Modeling and Simulation System (STEMSS) (Glass, 2002, Glass and Stevens, 1999) and its multiple-resolution modeling extension, STEMSS-MRM.

## 1.3 The Central Hypothesis

The central hypothesis of this dissertation is that multiple resolution modeling will allow the representation of populations at more than one level of resolution while keeping the results bounded by, and arbitrarily close to, the behavior of the constituent single-resolution models. An MRM is also expected to reduce the execution time of the high-resolution models. Please note that spatial modeling will not be discussed in this dissertation, instead, the dissertation will focus on what Levin calls the scale of "ecological organization" (Levin, 1992, p. 1994).

## 1.4 Research Questions

The hypothesis assumes three main points. First, ecologists want to represent a system at more than one level of detail. Second, there exists a system for multiple-resolution modeling. Third, there exists some methodology for comparing models. These assumptions are examined and the results of the examination are used in the development and testing of STEMSS.

The assumptions described above motivate several important sets of research questions:

- How do ecologists use computer simulation?
- How does a model's resolution affect its accuracy and execution speed? What factors affect the selection of a particular resolution? What problems may arise in representing a population at two levels of resolution?
- What are the computer science issues in developing MRMs and using them in a simulation? In particular, how can consistency between constituent models, in an MRM, be ensured?

- How well do multiple-resolution population models perform?

The dissertation will address these questions.

## 1.5 Research Methodology

### 1.5.1 Research Areas

#### 1.5.1.1 Research Area 1: Population Modeling and Computer Simulation

A motivating argument for the development of a population modeling framework is, that the current computer simulation approaches of population dynamics lack credence among some theoretical ecologists (Judson, 1994). The lack of credence, Judson argues, is due in part to the lack of a verifiable and replicable method for describing the programs used to model these systems. As these programs can consist of thousands of lines of code, written in cryptic styles and with idiosyncratic naming conventions. The underlying theoretical structure of the simulation is rarely presented, so the readers of simulation studies do not have sufficient information to verify the claims made in these studies.

One solution to these problems is development and implementation of a validated population modeling and simulation framework. The framework must be based on sound modeling and simulation principles, with well-defined simulation structures. It must also support the development of population models from the population modeling perspective.

To create replicable simulations, a method for developing models, within the context of the framework should be developed.

To create a general purpose population modeling framework, the framework designer must consider three issues. First, he or she must consider the questions ecologists ask. If the framework is to be of general use to ecologists, it must be able to support these questions without forcing the modeler to adopt idiosyncratic programming rules. Therefore, the framework must support ecological concepts by supplying an ecological modeling interface to the modeling and simulation structures. Such an interface will include, for example, a basic IOM that represents a population as a collection of individuals and an individual model that supplies predefined behaviors such as birth and death. The IOM's structure is, therefore, an association between individual models and a population model that implements these rules.

The second issue the modeler must consider is how computers are used to answer ecological questions. For example, there are two major types of computer simulation: discrete time (DTS) and discrete-event (DES) simulation. The designer must consider why an ecologist would select a particular simulation technique and how to support alternative techniques.

The third issue the designer must consider is the role of software engineering in the development of the framework. One problem with the current state of ecological modeling, particularly in light of Judson's critique, is the lack of a well-designed population modeling process. By considering these three issues, the framework designer should be able to construct a versatile and extensible system for population modeling.

### 1.5.1.2 Research Area 2: Resolution and Population Modeling

A model's resolution or scale refers to the granularity at which the system is represented. Before the work by Huston et al. (1988), modelers represented a population either as a single entity, such as the logistic equation, or as a set of equivalence classes, such as the Leslie matrix (Leslie 1945; Sinko & Streifer, 1967). These models use statistical aggregation of individual characteristics to represent the system's state and an equation, or set of equations, that update the state. When a population is large and the environmental conditions are static, these models are expected to generate valid simulations (Judson, 1994). However, as several researchers have noted, small populations or environmental stochasticity can diminish the reliability of these models. Therefore, the analytic equations are of limited utility.

To increase the validity of population models, ecologists increased the granularity at which populations are represented. For example, Sinko and Streifer (1967) developed a model of population dynamics based on the physiological characteristics of individuals. The model describes a population in terms of distributions of individual age and mass and represents population dynamics as a set of integro-differential equations. The increased resolution of the distribution models improves the model's accuracy when the population size is small relative to the more aggregated values. The advent of personal computers in the 1960's opened a new direction for increasing granularity based on a system theoretical approach (Huston et al., 1988). The increase in computing capacity allowed ecologists to develop computationally expensive models based on the explicit representation of individuals and their interactions by a set of behavioral rules deduced from direct observation.

The different approaches to population modeling pose an interesting question. Which approach is appropriate for a given problem (DeAngelis & Rose 1992)? This question is periodically asked and answered in reviews of ecological modeling techniques and the conclusions rarely vary. In general, IOMs are preferred when the modeled populations are small or subjected to large stochastic effects and the aggregate models are preferred when the modeled populations are large (DeAngelis & Rose 1992, McGlade 1999). Despite their advantages when the population size is small, IOMs have some important disadvantages. In particular, IOMs are computationally intensive, which has led to some interesting techniques to improve computational efficiency. For example, Rose, Christensen and DeAngelis (1993) and later Scheffer, Baveco, DeAngelis, Rose, and van Nes (1995) proposed partial aggregations of IOMs to improve efficiency. The latter work grouped physiologically similar individuals into objects called "super-individuals." Super-individuals exhibit the same behavior as individuals, except that mortality and birth are represented as changes in the size rather than the creation or removal of an individual.

One response to the problem of computational efficiency, and the basis of this dissertation, is that the selection of a single resolution may not be the optimal solution for the problem. For some problems, the time required to execute an IOM is too long and the inaccuracy of aggregate models is too great to rely on either approach. Consider an environmental emergency, such as an oil spill. The main priority for the groups responsible for minimizing the damage caused by the spill is the placement of clean up resources (U.S. Environmental Protection Agency [EPA], 2002). To optimize the placement of these resources, the groups need to compare clean up alternatives in a timely manner, but a thorough study using IOMs can take several days, weeks or even months to complete. The assumption in this dissertation is that an MRM could provide sufficiently accurate results, quickly enough to be useful.

Though researchers have suggested the development of population MRMs, no such models were found in the ecological literature. There are two reasons for the lack of progress. First, the lack of an ecological modeling framework inhibits the ability to extend modeling techniques and, second, the lack of a general structure for multiple-resolution modeling forces the modeler to create a complex system by hand. To facilitate MRM development, the requirements for a population modeling framework and a multiple-resolution modeling extension are identified in chapter II; the design and implementation for the framework and extension are discussed in chapter III.

#### 1.5.1.3 Research Area 3: Multiple Resolution Modeling

The fundamental problem in multiple-resolution modeling is the maintenance of consistency between resolutions (Davis, 1993; Natrajan, et al., 1997; Reynolds, et al., 1997). The MRM controls state changes by switching control between the constituent LRM and HRM. When an MRM switches control from one constituent model to the other, the states of the models must match. If they do not match, the model's state is not defined. Increasing consistency is particularly difficult when a simulation includes interacting

MRMs. Each MRM in the system must provide a consistent temporal view of its state to other models in the system. Otherwise, the interactions are inconsistent (Reynolds et al., 1997). To address this problem, an understanding of the computational issues related to multiple-resolution modeling is required.

#### 1.5.1.4 Research Area 4: MRMs Performance

The idea of multiple-resolution modeling makes sense in some contexts. There are reasonable ideas from probability theory to suggest the validity of MRMs. Also, MRMs are used in military simulations and the suggestion that MRMs may be useful in ecological modeling has lead to some credence in the technique. However, until the viability of MRMs is demonstrated, the usefulness of multiple-resolution modeling will not be accepted. To this end, three sets of tests, based on existing ecological models, were implemented to validate the STEMSS and STEMSS-MRM. Each test consists of a low-resolution (LRM) and a high-resolution (HRM) model developed using STEMSS and a STEMSS-MRM that couples the LRM and the HRM. The tests were designed to characterize the behavior of MRMs and are described in sections 1.6, 3.2.3, and 3.4.

The tests are designed to answer three questions: does the MRM maintain state consistency for these models? Do simulations based on the MRM remain bounded between the high and low resolution models? Do the MRMs improve execution time? In addition to answering these questions, the performance tests will be used to estimate the trade off between accuracy and execution time. This estimate is, in some sense, the most important result of the dissertation. The goal of multiple-resolution modeling, with respect to the dissertation, is to exploit the execution speed of aggregate models, while maintaining the accuracy of IOMs. By providing a method for estimating the trade off between accuracy and execution, the user can estimate the accuracy attainable by a simulation running for a given time.

### 1.5.2 Summary of Research Questions

The research performed for this dissertation follows a path from the motivating questions in ecological modeling and computer simulation to the development of a framework for population modeling and simulation and the development an MRM component for this framework. The first motivating question addresses the computational needs of modelers and how computer science can address these needs. The answer to this question is the development of a framework for ecological modeling and a method for developing models within the context of the framework. With the framework in place, the next step in developing and testing multiple-resolution population models is the creation of a definition for multiple-resolution modeling and the incorporation of that definition into the population modeling framework.

## 1.6 Ecological Model Primer

### 1.6.1 Introduction

“Ecology deals with organisms and their environments . . .” (Begon, Harper & Townsend, 1996, p. 5) and, as such, ecologists are interested in the behavior of individuals and the impact of the environment on individuals. In particular, ecologists are interested in how individual behavior affects the abundance of the organism over time. Hence, ecological models are concerned with the relationship between an individual’s environment and the ability of the individual to survive and reproduce.

Early efforts to describe population growth focused on the average-case behavior of individuals and projected the behavior to the population level. Theorists hypothesized that populations have an intrinsic growth rate and that resource limitation would constrain the population to a maximum allowable size. Verhulst and Pearl (Olinick, 1976), for example, developed population models that incorporate these inhibitory effects. For example, the discrete logistic model represents a population as a finite difference equation:

$$\Delta N = rN_t\left(1 - \frac{N_t}{K}\right) \tag{1}$$

where $N$ is the population size, $r$ is the intrinsic birth rate and $K$ is the population's carrying capacity. When $N$ is less than the $K$, $\Delta N$ is positive, that is, the population's size increases. When $N$ is greater than $K$, the population size decreases. The model relies on some aggregate measures, such as the average individual birth and death rates, as model parameters.

The representation of a single population by a finite difference equation was extended to represent interacting populations such as predator-prey relationships. For example, the Lotka-Volterra model (Olinick, 1976) of predator-prey interactions uses a pair of finite difference equations:

$$\Delta P = P_t(cN_t - d) \tag{2}$$

$$\Delta N = N_t(a - bP_t) \tag{3}$$

where $P$ is the size of the predator population and $N$ is the size of the prey population. The terms $a$ and $c$ are the growth rates of the prey and predator populations respectively and the terms $b$ and $d$ are the mortality rates for the prey and predators respectively.

The model assumes density independent growth for the prey population, which is constrained by predation. According to Murray (Murray, J. D., 1993), this assumption is unrealistic. For example, in the absence of predators, the prey population will encounter resource limitations. A more realistic version of the predator-prey model includes density-dependent terms for both the predator and prey populations:

$$\Delta P = kP_t\left(1 - \frac{hP_t}{N_t}\right) \tag{4}$$

$$\Delta N = rN_t\left(1 - \frac{N_t}{K}\right) - \frac{AN_tP_t}{N_t + B} \tag{5}$$

where $P$ is the size of the predator population and $N$ is the size of the prey population. The terms $k$ and $r$ are the growth rates of the prey and predator populations respectively, $h$ is the prey-dependent carrying capacity, $K$ is the prey population's carrying capacity, and the terms A and B are predation related constants.

Finite difference models ignore several important features related to the structure of a population. For example, individuals birth rates vary over the age of the individual. Therefore, a model of an age-structured population must account for the distribution of ages within a population, as well as the age dependent birth rate. An early model of age-structured populations, developed by Leslie (Leslie, 1945), represents a population as a set of equivalent age-classes $v$ where, for each $v_i \in v$, $v_i$ is the number of individuals in age-class $i$. Population growth is represented by a matrix **L** of birth and survival rates called the Leslie matrix. Population growth is computed as $v_{t+1} = \mathbf{L}v_t$ and the matrix is used to compute important population characteristics, such as stable population size and generation time. As with the analytic population models, the Leslie matrix represents a population by the average-case birth and death rates.

The models described so far subsumed the linkage between population dynamics and the environment. To make this linkage explicit, Sinko and Streifer (1967) extended Leslie's approach by decomposing populations into physiological distributions. These distributions include properties such as weight or length distributions. The model represents population dynamics as an integro-differential equation that depends on distribution-dependent growth and mortality rates. The impact of the environment is expressed as a function of the distribution of physiological characteristic and environmental characteristics.

More recently, ecologists have developed IOMs that use a physiological description of individuals to simulate population dynamics (Huston, et al., 1988; Caswell & John, 1992). IOMs represent a population as a set of explicitly modeled individuals. These models represent the behavior of individuals and their environmental and social interactions. That is, individual behavior is determined by the individual model, not the population model. This representation uses a fine-grained resolution of organizational behavior which gives ecologists a technique for developing more realistic models of population dynamics.

One such model, developed by Rose and others (Rose, Cowan, Houde, Coutant, 1993; Rose, Rutherford, McDermot, Forney, and Mills, 1996; Rose, Tyler, SinghDermot, and Rutherford, 1996), represents a yellow perch population as a set of individuals, partitioned into age classes. Each individual can grow from a "first-feeder" to an eight-year-old, assuming the individual survives that long. As the individual matures, the model determines the individual's gender (50% female) and each female breeds according to a length dependent probability function. The length of an individual is determined by the amount of food it eats, which, in turn, depends on prey density. Adults can prey on the young.

The model expresses growth due to consumption as:

$$\Delta w = pC_{max} - E - R_{tot} \tag{6}$$

where $C_{max}$ is the maximum possible consumption for an individual and $p$ is the proportion of $C_{max}$ realized by an individual. The variable $E$ represents consumption losses due to waste and $R_{tot}$ is the loss due to metabolic processes. The value of $C_{max}$ is a function of weight and environmental temperature:

$$C_{max} = a_c w^{b_c} - F(T) \tag{7}$$

where $a_c$ and $b_c$ are age-specific constants and $F(T)$ reflects the environmental impact on consumption. The expression for $R_{tot}$ is similar to $C_{max}$. The expressions for $E$ and $p$ both rely on realized consumption, $C_r$:

$$C_r = \sum_{j=1}^{n} \frac{\left[wC_{max}\left(\frac{PD_{ij}^{\ V}}{K_{ij}}\right)\right]}{1 + \sum_{k=1}^{n}\left(\frac{PD_{jk}^{\ V}}{K_{jk}}\right)} \tag{8}$$

where, $PD_{ij}^{V}$ is the density of prey item $j$ adjusted by its vulnerability $V$ to individual $i$, $K_{ij}$ is a saturation constant. The importance of Equation 8 is its impact on computational efficiency. To account for adult predation, the model must impose a method for finding appropriately sized individuals and supply a method for an interaction between each adult and the young cohort.

This model is computationally expensive. A 400-year population simulation based on this model took approximately four days to execute. Given that a thorough analysis of the model may require simulations of thousands of years, the model is impractical for general use. In response to this problem, Rose developed the super-individual concept of population modeling (Scheffer, et al., 1995). A super-individual is a modified individual model that aggregates similar individuals into a single representation. The super-individual's consumption and reproduction behavior are multiplied by the number of individuals it aggregates.

### 1.6.2 Test Models

The logistic model, density-dependent predator-prey model and the yellow perch models were used to test the STEMSS and STEMSS-MRM. The low-resolution models were based on the logistic equation (Equation 1), the predator-prey equations (Equation 4 and Equation 5) and a "super-individual" version of the yellow perch model. The high-resolution models are IOM equivalents of the low-resolution models. Details for each model are provided in Chapter III.

## 1.7 Plan of the Dissertation

Chapter 2 will discuss the current state of ecological modeling frameworks and the current state of multiple-resolution modeling. Two conclusions that dictate the direction of this dissertation are drawn from this discussion. First, the lack of a well-structured, discrete-event simulation framework has inhibited both the incorporation of discrete-event modeling and multiple-resolution modeling to the ecological modeling repertoire. Second, the lack of a structured method for multiple-resolution modeling has hindered the use of this approach generally.

Chapter 3 will discuss the development of the STEMSS and STEMSS-MRM designs. This discussion will show how STEMSS relates to modeling and simulation in general and how STEMSS can be used in ecological modeling specifically. The STEMSS-MRM design will introduce a new concept for multiple-resolution modeling and an implementation of the concept.

Chapter 4 will discuss the validation tests and results for STEMSS-MRM. The tests were done in three phases: validation of the method, use of STEMSS-MRM in a simple model and the use of STEMSS-MRM in a complex model.

Chapter 5 will discuss the performance tests of the STEMSS-MRM framework. The tests will show that STEMSS-MRM can improve the performance of the high-resolution model and it will identify specific costs in using STEMSS-MRM.

Chapter 6 will summarize the contributions of this dissertation and suggest possible future research directions.

# CHAPTER II

# LITERATURE REVIEW

## 2.1 Introduction

Chapter I outlined the research questions addressed in this dissertation. Chapter II will address the research questions related to the current state of ecological modeling and multiple-resolution modeling. The chapter begins with a discussion of the use of computer simulation in population ecology, focusing on the approaches used by ecologists to solve computational problems. The crux of this discussion is that ecologists have created modeling frameworks with varying degrees of success, but these frameworks have not produced a generally accepted technique for population modeling. The discussion also identifies a set of requirements for a population modeling framework:

1 the framework must be based on a well-defined and generally-accepted set of modeling and simulation system,

2 the modeling and simulation system must support a set of general modeling structures that can be specialized to arbitrary model types (e.g., discrete-time, continuous-equation, and discrete-event),

3 the framework must provide interface classes that refine the modeling structures to population modeling structures,

4 the framework must specify a method for code development based the framework's implementation.

The last requirement is particularly important. The framework consists of a set of classes that implements a general modeling and simulation system and an anoter set of classes that implements general concepts of population ecology. However, without a systematic way of describing code development, the modeler cannot express his or her models in a reproducible way.

The second part of the chapter will review the current state of multiple-resolution modeling. Specifically, it will focus on the definition and requirements of multiple-resolution modeling. This discussion will develop two significant points. First, the literature is ambiguous with respect to the meaning of multiple-resolution modeling and second, the current research has failed to produce a general framework for multiple-resolution modeling. The conclusion of this section is that following requirements are necessary for developing general MRMs:

1 the MRM must maintain consistency between the states of the constituent models at all times,

2 the MRM must support interactions between MRMs,

3 the MRM must maintain consistency between aggregated and disaggregated states.

## 2.2 Population Modeling and Computer Simulation

Chapter I included a discussion of general population modeling approaches. This discussion emphasized two population modeling paradigms: aggregate modeling that includes differential equations, and individual-oriented modeling, which describes populations as a set of interacting models of individuals. IOMs are designed to represent populations using the theoretical concepts and experimental observations of population ecology. This design implies that IOMs rely on ecological intuition, rather than the mathematical abstraction associated with aggregate models (DeAngelis and Rose, 1993; McGlade, 1999). Moreover, by aggregating individual parameters, the resulting models lose information regarding population dynamics. Aggregate models are therefore presumably less accurate than models that incorporate explicit representations of individual behavior. IOMs, on the other hand, are computationally expensive, as the models must track thousands of individuals. According to Huston et al. (1988), the rise of individual-oriented modeling is the result of dissatisfaction with aggregate models and an increase in the computational capacity of computers.

The central role of systems theory in the development of individual-oriented modeling has lead to two specific complaints about IOMs. First, IOMs do not lend themselves to analysis. The models are complex, potentially consisting of thousands of simultaneous equations and the temporal behavior of these models can be discontinuous. The second complaint is that the code used to represent individuals and populations is not published or subject to review (Judson, 1994; McGlade, 1999). The difficulty in analysis limits the theoretical use of IOMs, but the lack of review makes the models "suspicious" (Judson, 1994, p. 12), which can reduce their acceptance.

To reduce the suspicion caused by the lack of a formal review, ecologists have pursued various design techniques to control the model development process. These techniques range from the imposition of coding requirements to the development of meta-modeling languages. Though many of these techniques are no longer areas of active research, they have identified requirements for a general population modeling system. This section will describe representative examples of these techniques from which a set of design principles will be derived.

## 2.2.1 Software Design in Ecology

### 2.2.1.1 Style and naming conventions

A basic technique for controlling the software development process is the imposition of coding style and naming conventions. By imposing conventions on programmers, the interface between different pieces of code will have a consistent meaning, regardless of whom actually writes or develops the code. This approach enhances the code's readability and supports the development of models as components rather than complete programs.

These benefits are particularly important in collaborative systems where groups of developers contribute to modeling efforts and the models must share code. An example of this technique is found in the Across Trophic Level System Simulation (ATLSS) project (ATLSS, 2002a; ATLSS, 2002b). ATLSS is a collaborative effort between academic and government researchers to develop models of the ecosystems of the Everglades and Southern Florida.

The ATLSS base model consists of hydrological and topological models of Southern Florida that gives contributing modelers a predefined structure for spatial interactions. The modelers create specific population sub-models, which are coupled to the base model to produce spatially-explicit ecosystem simulations (DeAngelis, et al,

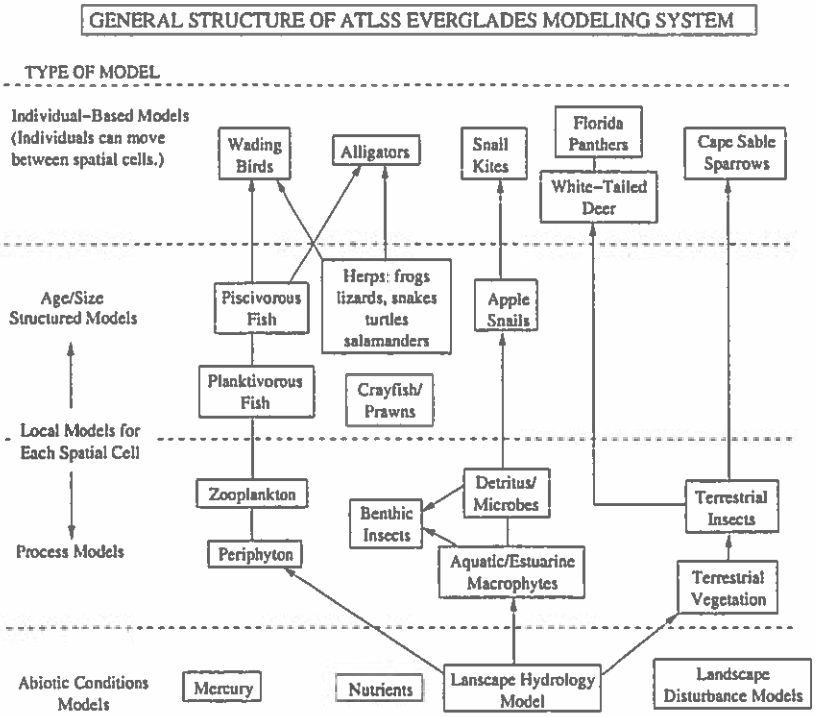

Figure 2. ATLSS Ecosystem Structure, adapted from the ATLSS project home page (ATLSS, 2002a)

1998; Abbot, Berry, Comiskey, Gross & Luh, 1997). The structure produced by coupling the models forms a graphical representation of the ecosystem that shows the biomass and nutrient flow through the system. Figure 2 is one such representation used by the ATLSS group (ATLSS, 2002a).

The graphical representations and the naming and style conventions facilitate the description of the model by explaining how information is shared among the components of the model. The description, however, is limited to data flow; it does not describe behavior of the population models developed by the contributors and it provides no support for defining these models. Instead, it imposes a programming discipline on the modelers, which can only be successful if the modelers are willing to abide by that discipline. Therefore, the population models are developed in an *ad hoc* fashion and the concerns raised by Judson (1994) and McGlade (1999) are not satisfied by this approach.

#### 2.2.1.2 Frameworks

Two problems with using coding requirements as a software development technique is their reliance on disciplined programming and the lack of support for specifying system behavior and interactions. To aid modelers in specifying models, theoretical ecologists have created population modeling frameworks. A software framework, according to Booch, is "a set of classes that provide a set of services for a particular domain; a framework thus exports a number of individual classes and mechanisms that clients can use or adapt" (Booch, 1994, p. 514). A population modeling framework therefore supplies the modeler with structures for developing population models that subsume the programming discipline required by coding requirements. If the framework is developed with domain specific support in mind, then it can also support model development.

The hierarchical and concurrent individual-based model (HCIBM) system, developed by Palmer (1993), is an early attempt to produce an object-oriented population modeling framework. Though work on that project apparently has been discontinued, some of his ideas are important for the development of a population modeling framework. In particular, Palmer discusses three specific issues that would facilitate the development of a general population modeling framework: the use of object-oriented design, the hierarchical description of a population model and the explicit representation of individual development within the model.

The interest in using object-oriented design (OOD) in developing a simulation framework is not surprising, as many object-oriented concepts originated in the discrete-event simulation language, Simula 67. The relationship between OOD and modeling is especially significant because software engineers have applied this relationship to software development and to the graphical representation of programs (Booch, 1994; Booch, Rumbaugh, Jacobson, 1999). In the development of HCIBM models, Palmer uses a form of OOD and a graphical representation, to describe the relationship between spatial location and individual behavior.

Figure 3 is an example of Palmer's representation of a spatially distributed, age-structured population. The diagram describes a population in the context of its

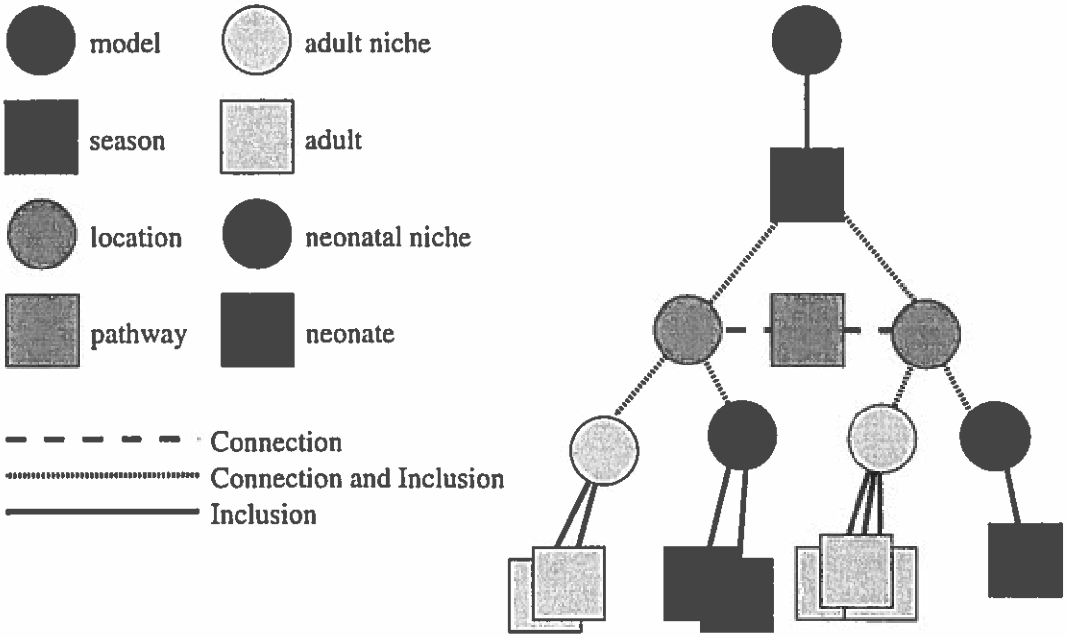

Figure 3. Palmer's Representation of Spatially Explicit Population from Palmer (1992 p.194)

environment (i.e., season and spatial location) and its organization (i.e., sub-populations). The relationship between the environment, social organization and individuals is described as a hierarchy of models, where each level of the hierarchy influences the models below it through a set of predefined paths.

One of the more important individual-oriented modeling problems addressed in Palmer's example is the explicit representation of individual development. An individual model represents the individual's life-cycle holistically, as opposed to a set of "life-stage" classes. This allows the modeler to specify an individual by its developmental and social behavior, so the individual model is consistent with an intuitive perception of individual behavior.

The HCIBM hierarchy is constructed using two class base classes: Part and Whole. The Part class represents entity models in a system and the Whole represents collections of Part objects. For example, an individual is represented as a Part object and the niche is represented as a Whole object consisting of zero or more individuals. The Part and Whole classes implement a set of accessor methods for setting and retrieving state values and a set of behavior methods, for representing the model's dynamics. These classes are general modeling classes, in that they provide a structure for implementing models, but they do not support the modeler in developing models. For example, Palmer describes how to define a specific individual as a Part class. However, to use this definition, the modeler is required to formulate the model, then translate that formulation into the HCIBM structure. This translation requires the use of idiosyncratic methods, such as "setPotential," and statements, such as "parts := self inclusions," which limits system to those users who are willing to learn the HCIBM terms and structures. Anecdotal evidence suggests, however, that modelers are unlikely to adopt a framework that imposes a set of terms, structures and concepts that are not consistent with the modelers existing set of terms, structures and concepts. As Maxwell and Costanza put it, "the current generation of models tend to be idiosyncratic monoliths that are comprehensible only to the builders" (Maxwell & Costanza, 1995, p. 251).

Despite the problems associated with HCIBM, the system does provide a starting point for the construction of a population modeling framework. Palmer's graphical representation is an improvement on the representation supported by the ATLSS project as it identifies the interactions between individuals and their environment. Therefore, the representation aids communication about the model's structure, which partially addresses Judson's complaints regarding the ability to review simulations. HCIBM also provides structures that support model construction even if the support is idiosyncratic.

#### 2.2.1.3 Ecological Modeling Languages

Modeling frameworks provide developers with a set of structures that support the construction of models, but they require adoption of the framework's underling assumptions regarding modeling and simulation. Alternatively, one may use modeling languages. Maxwell and Costanza (1995), Costanza and Maxwell (1997), for example, developed the Spatial Modeling Environment (SME) as a meta-modeling language for spatially-explicit population models. SME uses off-the-shelf modeling languages, such as STELLA, to specify model behavior, then translates these specifications to the underlying spatial representation through the Modular Modeling Language (MML) which produces Java or C++ code (SME, 2000).

The important feature of SME, with respect to the current state of population modeling, is the use of a graphical modeling system, STELLA. STELLA uses a set of icons to represent a system. These icons include stocks (i.e., variables), flows (finite difference equations), converters (constants) and connectors (flows), in addition to more specialized flow types. A STELLA implementation of the logistic equation (Equation 1),

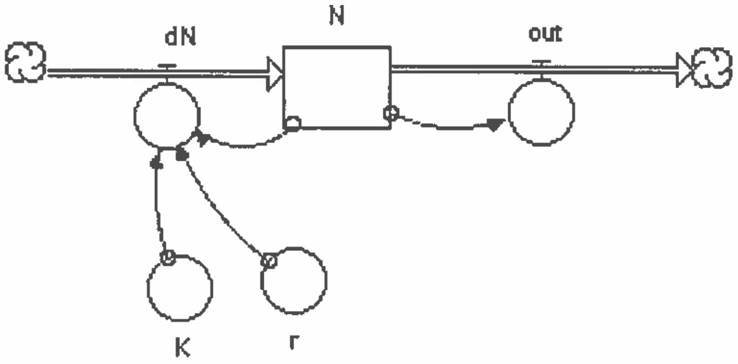

Figure 4. STELLA Representation of the Logistic Model

shown in Figure 4, demonstrates how a user would produce a population model. The number of individuals in a population, *N*, is represented by the stock icon labeled *N*, the birth rate, *r*, and carrying capacity are represented by converters labeled *r* and *K*, respectively, and the differential equation is represented by the flow, *dN*. On each time step, *N* individuals "flow out" of *dN* and *rN(1- N/K)* individuals "flow in." The modeler defines the start and stop times, and the time step of the simulation, which are used to control the simulation. STELLA converts the graphical representation to a finite difference equation specification, which MML uses to describe the non-spatial part of and SME model.

Since STELLA is a differential equation solver (STELLA, 1997), SME supports the development of aggregate models. Ecologists, however, have expressed an interest in the use of discrete-event simulation for population modeling, especially with respect to individual-based modeling (Palmer, 1992, Haefner, 1992; Gathmann and Williams, 1997).

The SME project was extended to include for discrete-event simulation frameworks such as Swarm (Hiebler, 1994; Santa Fe, 2002), which is an agent-based simulation framework. As with the other modeling and simulation frameworks discussed earlier in this chapter, however, Swarm does not provide a domain specific interface for ecological modeling, so the models are apparently developed in an *ad hoc* fashion.

### 2.2.1.4 A Synopsis of Ecological Software Development

The desire for a generally accepted technique for describing and creating IOMs has not been fulfilled. The existing structures fail to provide the necessary ecological specificity and modeling support for the development IOMs. This claim is made apparent when considering the current state of individual-oriented modeling. In a survey of papers from the journal, Ecological Modeling, between May 2001 and May 2002, ten papers with the keywords "individual-oriented" or "individual-based" were published. The papers were sorted according to software design style: "ad hoc," "design concept" or "software package." "Ad hoc," implies the modelers did not explain their software design principles. "Software design concepts," implies that the paper describes the software design. "Software package," implies the use of off-the-shelf software. Of the ten articles that qualified, six were "ad hoc," one was a "structured design" and three were "software packages." The survey suggests that population modelers have not yet addressed the problems identified by Judson (1994) and McGlade (1999).

The purpose of a framework is to supply the modeler with a set of services that support the development of arbitrary models. From the literature review, these services include a simulation engine for simulating system dynamics and a set of classes that support basic modeling structures. In providing these services, the framework removes important programming and design issues from the modeler and, therefore, the framework must implement a well-designed and generally-accepted modeling and simulation theory. The literature also suggests that modelers require several simulation types. These types include differential equation simulators, such as SME/STELLA, discrete-time simulators, such as ATLSS and discrete-event simulation, such as SME/Swarm and inter-cite (Gathmann and Williams, 1997; Gathmann, 1996). A successful framework must, therefore, support these modeling systems.

As described in the literature review, the population modeling framework must support the modeling process from the ecologist's point-of-view. This support is not a matter of conforming to the nomenclature of the population ecology, rather, the framework must supply a set of classes that conform to population modeling concepts. For example, an IOM population model is structure that contains zero or more individuals of a given species. To represent an IOM, the framework must include a container class to represent the population and an individual model class to represent the individual. The population class must include specific methods that represent an individual joining the population such as birth or immigration and methods that represent an individual leaving the population such as death or emigration.

The final requirement for the population modeling framework identified in this review, is a method for concisely specifying models in a way that supports the derivation of code from the specification. STELLA is an example of such a system. The graphical specification of the model is used, in conjunction with user input, to derive code for the model. In doing so, a STELLA specification also supplies a format for describing, discussing, and reproducing the model.

To summarize, the framework must meet the following requirements.

1 The framework must be based on well-defined and generally accepted modeling and simulation system.

2 The modeling and simulation system must support a generalized notion of modeling that can be specialized to arbitrary model types (e.g., discrete-time, continuous-equation, discrete-event).

3 The framework must provide interface classes that refines the modeling structures to population modeling structures,.

4 The framework must also specify a technique for defining how the code will be structured.

### 2.2.2 Software Design in Computer Science

Software engineers have developed methods for designing and implementing frameworks that are extensible and maintainable (Booch, 1994; Shaw & Garlan, 1996;

Stroustrup, 1997; Oestereich, 1997; Booch, Rumbaugh & Johnson, 1999). The Unified Modeling Language, or UML (Booch, Rumbaugh & Jacobsen, 1999; Conallen, 1999; Kobryn, 1999), is used as the basis for specifying models in this dissertation. Specifically, the model classes (e.g., individuals and populations) are identified from a narrative description of the model. From this narrative, the inheritance and containment relationships between classes are identified and used to develop a structural description of the system. These structures are used, in conjunction with class and interaction diagrams, to specify the system.

The interaction diagram, Figure 5, describes the interactions between objects. Each column of in the diagram shows the sequence of interactions for a particular object. The

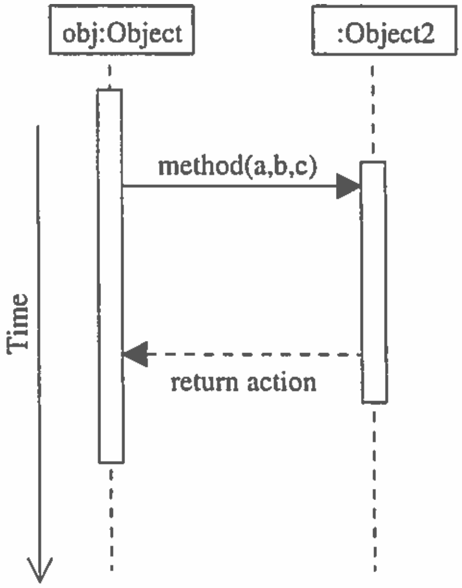

Figure 5. UML Interaction Diagram

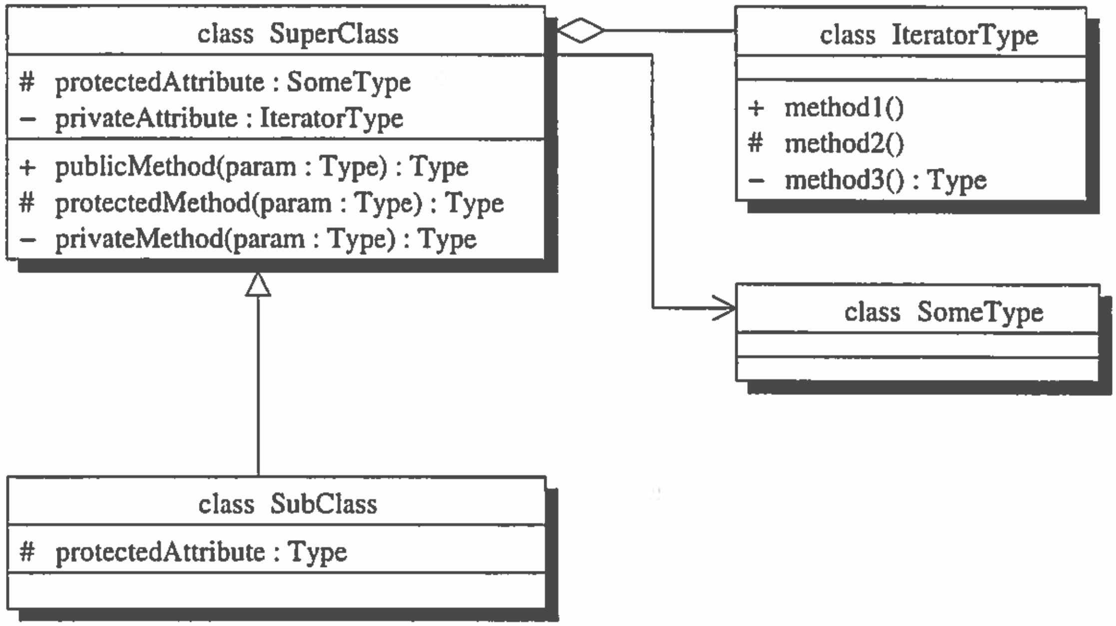

Figure 6. UML Class Diagram

interactions are represented as arrows pointing from one object to another and are annotated with the name of the interaction method. In this example, an object *obj* interacts with an arbitrary object of class Object2 using the interface called *method(a,b,c)*. This method returns the value *action.*

The class diagram, shown in Figure 6, shows the relationships between classes, including inheritance and association. The class, SuperClass includes an aggregate of IteratorType objects, called "privateAttribute." The SuperClass object interacts with objects of type SomeType and is inherited by class SubClass. The tags "#," "-," and "+" mean that the associated method or attribute is protected, private or public respectively.

## 2.3 Creating an Ecological MRM

Ecologists have argued that multiple-resolution models are desirable for efficient simulations, assuming the models can maintain reasonable accuracy (DeAngelis & Rose, 1992; Levin, 1992; de Vasconcelos, et al., 2002). By representing a system at multiple levels of resolution, the simulation can direct computational effort where it is needed while ensuring consistency between the constituent models. To produce a multiple-resolution modeling extension for a population modeling framework, two issues must be resolved. First, a general definition of MRM must be developed and, second, structures to implement this definition must be developed.

### 2.3.1 What is Multiple Resolution Modeling?

To implement an MRM, there must be some description of the requirements for an MRM and this description must satisfy the requirements of the source problem. Recall from chapter I, the multiple-resolution modeling idea expressed by DeAngelis and Rose. The idea, was to model a system using a valid high-resolution model (HRM) unless state of the MRM allowed the use of a valid, low-resolution model (LRM). In population models, the switching condition could be based on population size. The MRM switches between models under specific conditions either by aggregating the state of the high-resolution model or by disaggregating the state of the low-resolution model. To produce a valid MRM, the transition functions of the constituent models must produce consistent results. This is a particularly challenging constraint as the MRM's constituent models will not necessarily have the same functional form. For example, growth in an IOM (high-resolution model) is due to individual behavior and growth in an aggregate model, such as the logistic model, is computed by a finite difference equation. The relationship between the consistency requirements and examples of multiple-resolution modeling found in the literature are examined in the following section.

### 2.3.2 Types of MRMs

The first category of MRMs consists of models with multiple state resolutions, such as grid refinement models. These models represent the state of a system at more than one level of granularity, for example, *Geofogo* is a fire ecology modeling system (de Vasconcelos, et al., 2002) that represents spatial data at several levels of resolution. In the example provided by the authors, fire spread is modeled as a function of the combustible material in a given area. The spatial distribution of the combustible material is represented as a grid of nodes, where each node includes the amount of combustible material within the region represented by the node. *Geofogo* allows the modeler to define spatial data at more than one resolution, though each simulation will use only one representation (i.e., the models do not switch between representations). There are three interesting points from this study. First, *Geofogo* does not use the multiple resolutions directly, rather it combines the resolutions as required. Second, the combination of resolutions is one-way, that is, spatial-data is aggregated only. Third, the model supports multiple state resolutions, though it maintains a single behavioral representation, that is, all resolutions use the same equation.

The second MRM category consists multiple temporal resolution models. The essential problem with multiple temporal resolutions is that differing time scales can create representational inconsistencies. This is most easily seen in discrete-time systems when the constituent models update their state values at regular, but disparate, time steps. One model, usually the high-resolution model, operates on a finer time scale than the other. When the MRM switches the active model, the passive model's transition function may not have a valid update at the current time. As the state of a discrete time model is not defined between transitions, the disparity in time steps could lead to an undefined state. To correct this problem, the framework must support a mechanism that approximates the model's behavior between state updates.

The ability to impose state consistency is necessary for arbitrary multiple resolution models, however, if the constituent models transition simultaneously, then imposing state consistency is not sufficient: the states induced by one model will not necessarily match the state of the other model. If both the transitions are valid, but the states are different, then imposing consistency based on the results of one model is not sufficient. This leads to the third category of MRM, those that impose behavioral consistency. The MRM must supply a mechanism for resolving state differences induced by the models. This problem is suggested by the Davis-Hillestad diagram (see Figure 1).

The fourth category of multiple-resolution models consists of the models that impose interaction consistency. When a system consists of interacting MRMs, the MRMs may interact using different resolutions. For example, in a predator-prey model, the high-resolution predator model may interact with the low-resolution prey model. To account for this condition, the output of the predator model must be translated to the resolution of the prey model, then mapped to the input of the prey model.

The Multiple Resolution Entity (MRE) concept developed by Reynolds, et al. (1997) address the problem of cross resolution interactions. The modeler must identify a set of core state values that can generate the required state values for each level of resolution. When a state value is required for an interaction, the MRE's consistency manager generates the value from the core set. To coordinate the interactions between MREs, the MRE uses a "public world view" to map interaction requests and data to the MRE. The MRE uses a consistency enforcer to map the data to the constituent models and to map core values to the "public world view."

Though the MRE is a general concept for developing MRMs, it only addresses the interaction aspect of the MRM and it does not operate within the context of a theoretical framework for modeling and simulation. As such, it offers neither programming nor conceptual support for the development of MRM interactions. Nevertheless, the observations made by Reynolds et al. (1997) and Natrajan et al. (1997) suggest important constraints for multiple-resolution modeling.

The fifth category of multiple-resolution models consists of models that impose the organizational consistency. LRMs are represented by model structures, whereas HRMs are represented by network structures. When the MRM imposes state consistency, it forces the state of the LRM to match the state of the HRM but it does not impose consistency on the HRM's component models. Therefore, as a consequence of translating the state of an LRM to the state of an HRM, the aggregation of the HRM's component model might not match the HRM's state. To ensure organizational consistency, the MRM must resolve this inconsistency. The process is known as disaggregation. The MRM suggested by DeAngelis and Rose (1992) requires that, when the population size falls below some value, the MRM must impose state consistency on the HRM and the HRM must disaggregate its state by adjusting the number of individual models.

This literature review suggests five properties for characterizing model resolution: time, state, behavior, organization and direction (i.e., aggregating only or aggregate-disaggregate). They also suggest three requirements for the development of valid MRMs, namely:

1. the MRM must continuously maintain consistency between the states of the constituent models,
2. the MRM must support interactions between MRMs,
3. the MRM must maintain consistency between aggregated and disaggregated states.

## 2.4 Summary

This chapter reviewed the current literature regarding the use of ecological modeling and simulation frameworks, the interest in multiple-resolution population modeling among ecologists, and the current state of multiple-resolution modeling approaches. The findings are summarized below:

- Ecologists have been working on various techniques to create modeling frameworks to solve both problems with consistency in model representation and communicating about models. These approaches range from imposing a programming style to providing a graphical user interface.
- The ecological modeling frameworks are based on well-defined structures for modeling and simulation.
- Imbedded in the discussions regarding individual representation is the problem of dynamic behavioral changes. An arbitrary IOM must represent parallel behaviors and developmental changes to these behaviors.
- MRMs are desired by ecologists, but there are no frameworks for multiple-resolution modeling.
- The key to producing MRMs is to impose state consistency.
- MRMs need access to both data and behavior of a single entity.

# CHAPTER III

# SOFTWARE DESIGN

## 3.1 General Design Problem

Chapter II described the current state of multiple-resolution population modeling frameworks and derived a set of rules for producing a population model framework and an MRM extension based on the description. Chapter III will describe the implementation of a multiple-resolution population modeling framework based on the requirements identified in chapter II. This discussion begins with the development of a modeling framework and a set of population modeling structures. It then describes the implementation of an MRM using the modeling structures. The chapter finishes with a description of a method for defining population models,

## 3.2 The Design and Implementation of the STEMSS

The Spatio-Temporal Ecological Modeling and Simulation System (STEMSS) is a population modeling framework consisting of four packages: the model abstraction, the simulation engine, the population modeling interface and a support package (see Figure

7). The simulation engine controls time progress by storing model events in time order and iteratively selecting and processing them. The model abstraction is a set of classes that implement the model and network classes described in chapter II and form the base classes for the population modeling interface. The interface classes implement concepts from population ecology while conforming to the modeling structures. The support package provides the framework with basic abstract data types, such as lists, and simulation utilities, random number generators and file I/O handlers. This section will describe the development of the model abstraction package, followed by the development of the simulation engine package.

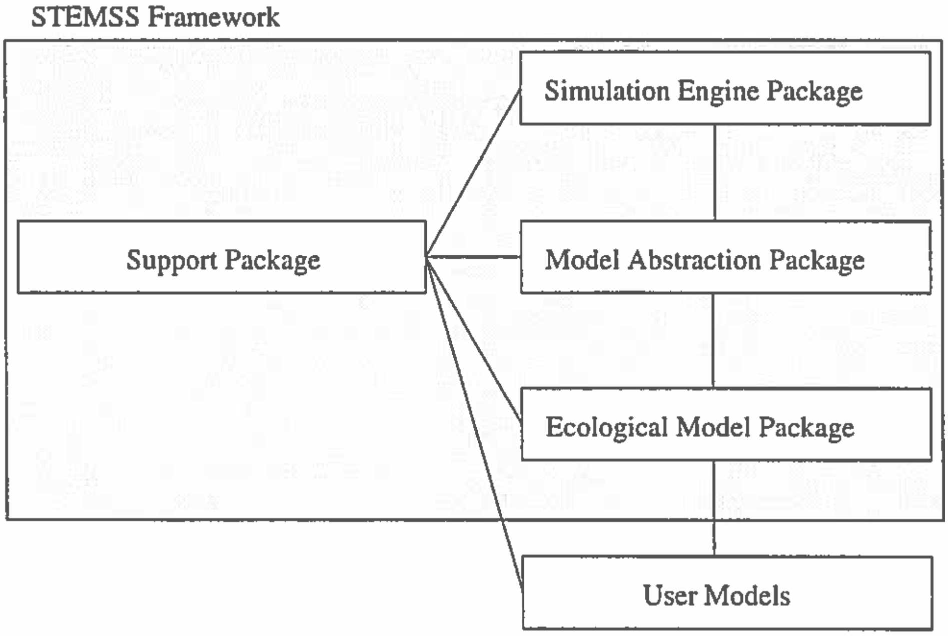

Figure 7. Simulation Engine Class Diagram

### 3.2.1 Model Abstraction Package

The model abstraction is based on a formal theory of modeling and simulation (Zeigler, 1976). The theory identifies two general modeling structures, the model and the network. The model is defined as the structure,

$$M = \langle X, Y, Q, \delta, \lambda, \tau \rangle \tag{9}$$

where, $X$ is the set of input values, $Y$ is the set of output values, $Q$ is the set of state values, $\delta$ is the transition function, $\lambda$ is the output function and $\tau$ is the time advance function. The transition function, $\delta$, has two components, the internal transition, $\delta_{int}$, which represents the model's behavior and the external transition, $\delta_{ext}$, which represents the interactions between models.

The network is defined by the structure,

$$N = \langle \{M\}, \{I_{ij}\}, \{Z_{ij}\} \rangle \tag{10}$$

where $\{M\}$ is the set of models associated with the network, $\{I_{ij}\}$ is the set of models $\{M_j\}$ that interact with $M_i$ and each element of the set $Z_{ij}$ (also called Zmaps) maps the output values of $M_i$ to the input values of $M_j$. If the transition function of model $M_i$ involves an interaction with another model, $M_j$, (e.g., predation), then the output of $M_i$ is mapped to the input of $M_j$ via $Z_{ij}$.

The model and network structures can be specialized to represent discrete-time, differential equation, or discrete-event models and networks. The flexibility of these structures satisfies the requirement that the framework support these types of simulations. Researchers have also extended these models to include hybrid models, i.e., models using differential equation and discrete-event systems, and dynamic networks that dynamically alter the model set, $\{M_i\}$, the influence set, $\{I_{ij}\}$ and the set of maps $\{Z_{ij}\}$. The STEMSS model abstraction package (see Figure 8) uses the hybrid model as the base class for all models and dynamically structured discrete-event network as the base class for all network types.

### 3.2.1.1 The Model class

The STEMSS Model class is based on the discrete/continuous hybrid model structure described by Praehofer, Auernig and Reisinger (1993) and Reisinger (1995):

$$M_H = \langle X_d, Y_d, Q_d, X_c, Y_c, Q_c, \delta_{int}, \delta_{ext}, \tau \rangle \qquad (11)$$

where $X_d$, $Y_d$, and $Q_d$ represent the system's discrete input, output and state values respectively and $X_c$, $Y_c$, and $Q_c$ represent the system's continuous input, output and state values respectively. The functions $\delta_{int}$ and $\delta_{ext}$ control internal and external state changes and $\tau$ defines the model's time advance function. The purpose and behavior of the time advance function are the same as that of the time advance function of the general model structure, *M*. The internal and external transitions extend the definition to include rate equations.

Transition functions represent the behavior of a modeled system. For example, the reproductive and metabolic behavior of individuals is represented as a set of reproduction

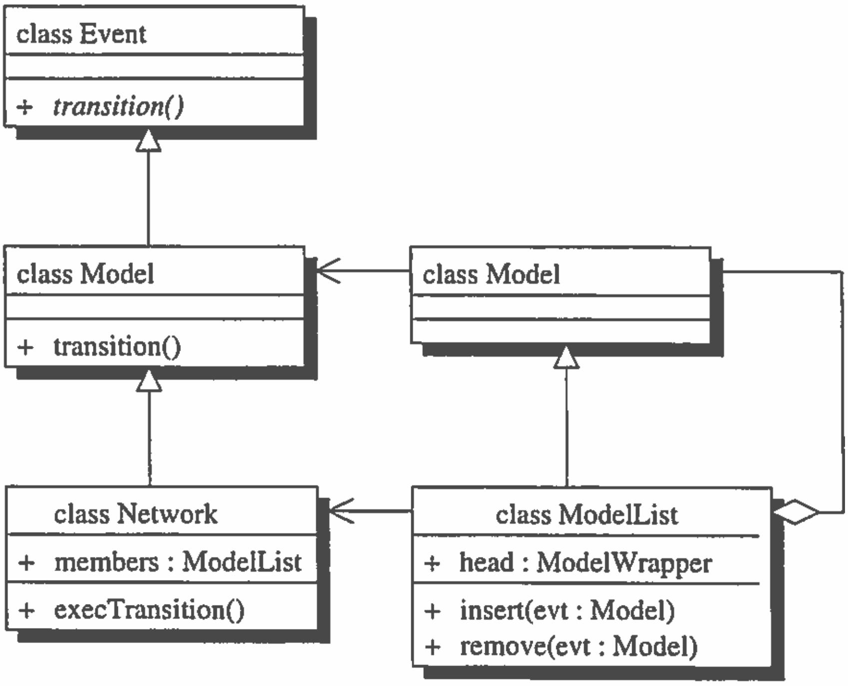

Figure 8. STEMSS Model Abstraction Package

and survival transition functions (see Figure 9). The reproduction transition in this example begins with a female in a quiescent state. The individual moves from the quiescent state to "findMate" state, which involves an interaction between the individual and its environment. Once a mate is found, the individual mates and may become pregnant

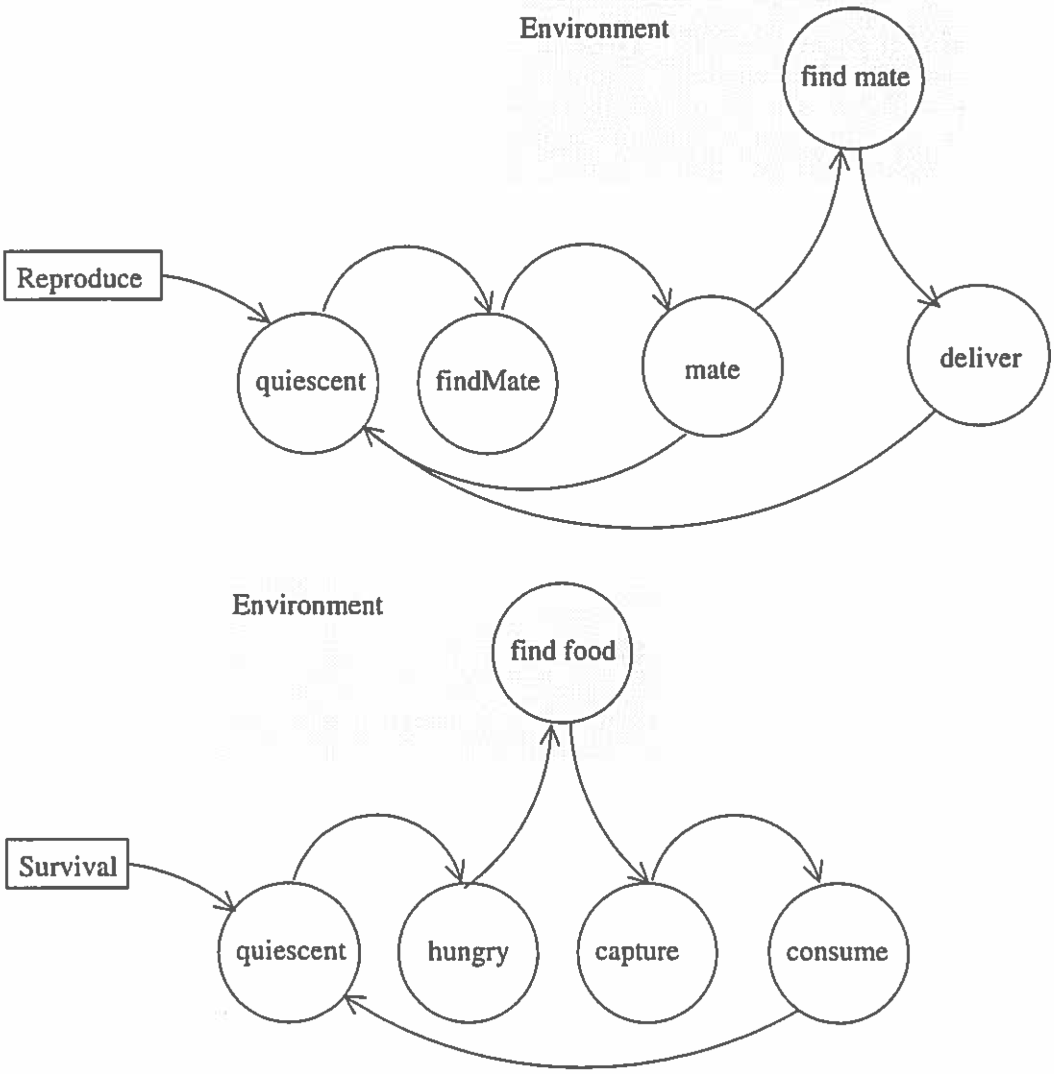

Figure 9. Individual Transitions

which involves another interaction and state change. If she becomes pregnant, she will produce offspring then return to a quiescent state (the model assumes no parental care). If she does not become pregnant, she returns to the quiescent state. The survival transitions begin with the individual in a quiescent state. The individual transitions from the quiescent state to a hungry state and begins a search for food. To find food, the individual must interact with its environment by searching for food. After finding food (i.e., after the environment responds with a prey item), the individual consumes the food, adjusts its physiological make up, then returns to the quiescent state.

To implement model behavior described in Figure 9, the modeler specifies the functions and associates them with an array of method pointers, *deltaInt*, for internal transitions and *deltaExt*, for external transitions, that represent separate sets of transition functions. Each function specifies the conditions for changing the transition state and, if necessary, configuring the transition array to use a different transition function. The model class records the timestamp (i.e., the time at which the transitions must happen) and sets the transition time as the smallest of these timestamps.

### 3.2.1.2 The Network Class

The STEMSS Network class is based on the dynamically structured discrete-event network developed by Barros (Barros, 1996, 1997; Barros, Zeigler, and Fishwick, 1998). As with the Model class, the user must specify the Network's discrete and continuous values and configure the *deltaInt* and *deltaExt* arrays. The Network includes a model list and a set of functions that allow models to join or leave the network. In addition to the join and leave methods, the Network also includes an array of network transitions, *deltaExec*, that coordinate interactions between models and the network. These methods are defined by the user.

The network transitions are used to reconfigure the network by adding or removing models or reconfiguring the relationships between the models. For example, in an IOM, the birth or death of an individual is modeled by the addition or removal of an individual from the population. The user can implement birth as an instantiation of a newborn individual and the insertion of the individual into the population network's model list. When an individual dies, the model notifies the population network, which removes the model from the network.

### 3.2.2 Simulation Engine Package

The simulation engine package is responsible for ensuring that model transitions take place in causal order. To implement this behavior, the package provides three main classes, the Event, EventList and EventLoop classes. Event objects are used to store models in a time ordered EventList object. Specifically, the Event class is a super class of the Model class, which stores the model's time stamp and other ordering information. The EventLoop removes the first event in the EventList, updates the current time to the time stamp of that event and processes it (i.e., it invokes the transition function). After processing, the event, the EventLoop determines whether to reinsert the event into the

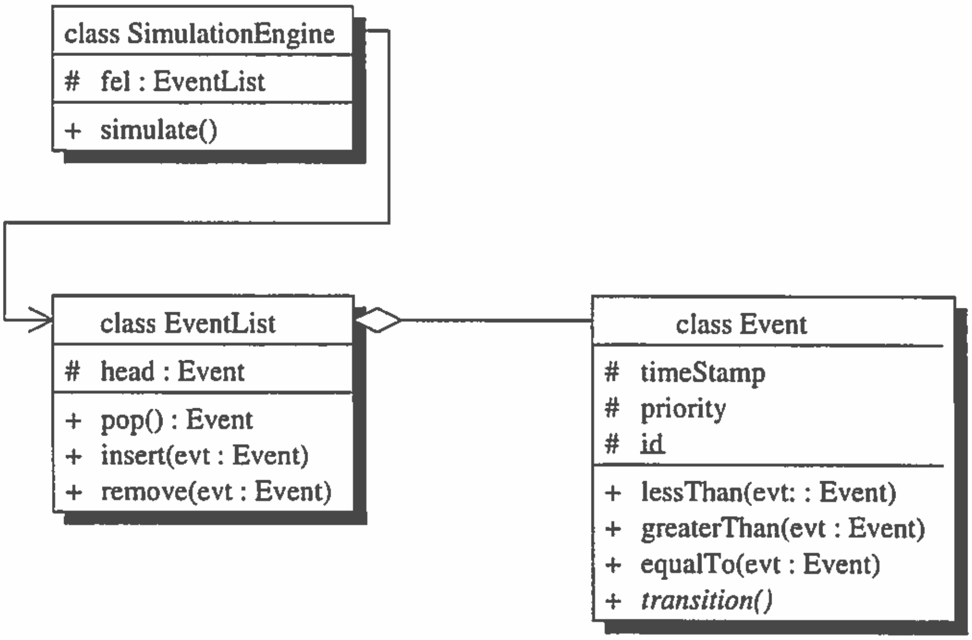

Figure 10. Simulation Engine Class Diagram

loop. This process repeats until the current time reaches a predetermined time stamp or until the event list is empty.

The class structure used to implement the interface diagram is shown in Figure 10. The Event class includes a timestamp, which specifies when to invoke the event transition function, and an abstract internal transition function. The Model class implements the internal transition function. Upon completion of the transition, the user-defined function will return true, if it is going to process another event, or false if it does not. If the transition returns false, then the model must either be deleted or stored elsewhere the list.

## 3.3 The Multiple Resolution Modeling Extension

### 3.3.1 Consistency Maintenance

The implementation of an MRM relies on a definition of multiple-resolution modeling that is consistent with the STEMSS model and network structures. To create this definition, consider the purpose of the MRM. An MRM uses a valid LRM and a valid HRM to model the behavior of a system. In this instance, "valid" implies that the behavior predicted by the model is similar to the real system under a set of specified conditions, for a specified length of time. If both models are valid, then there must be a correspondence between them, that is, the state of one model must match the state of the other model, within some tolerance. The main purpose of the MRM is to leverage the correspondence between the models by imposing state consistency between the LRM and the HRM.

Recall the assertion by Davis and Hillestad, from chapter I that if two models of a system are in the same state before a transition, both models should be in the same state after the transition (see Figure 1). To ensure this equivalence, the MRM must provide a function to translate the state of one model to that of the other model. Unfortunately, the implication of the diagram is that transitions involve state changes only. However, from the discussion of the model and network structures, transitions can include interactions between models (external transitions) and changes in the relationships between models (network transitions). As each transition has an impact on different aspects of the model, each transition must be accounted for separately. That is, an internal transition can impact

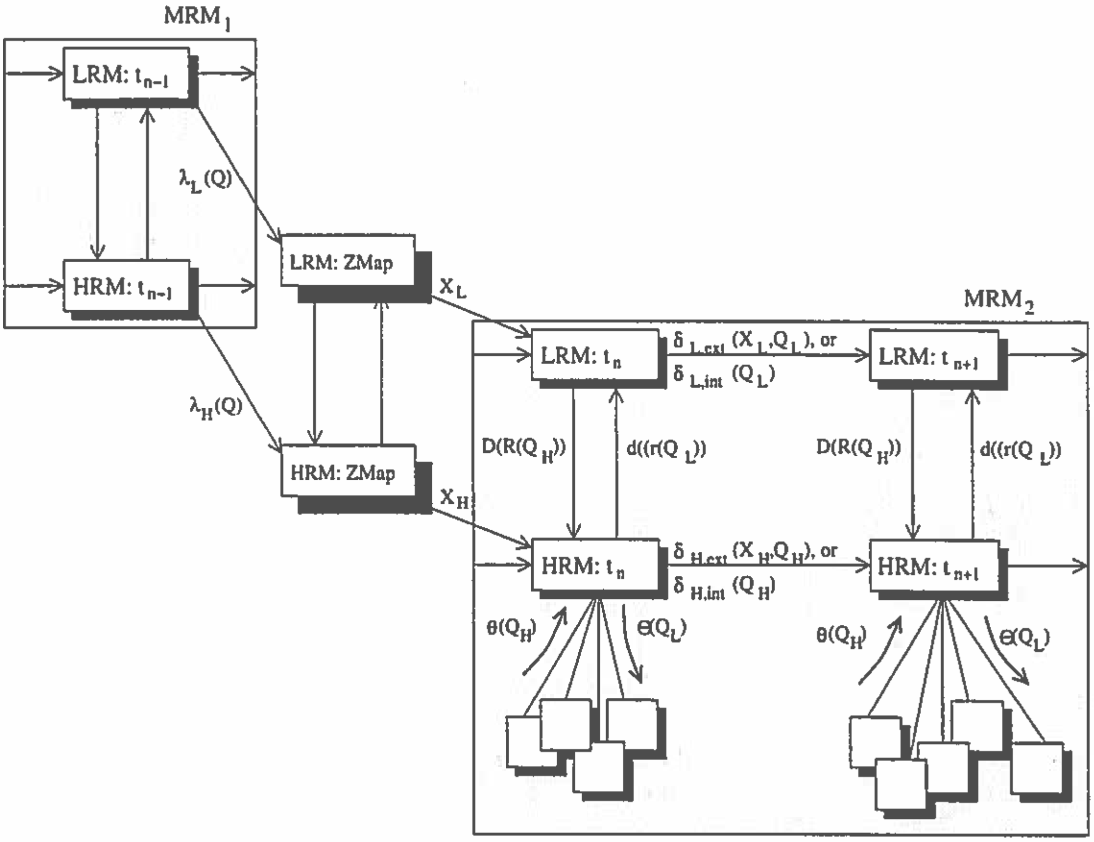

Figure 11. Alternate Version of Davis-Hillestad Diagram

the model's state, external transitions can impact the model's input and network transitions can impact the relationships between the models in a network. These addition behaviors lead to an alternate version of the Davis-Hillestad (see Figure 11).

Another implication of the Davis and Hillestad diagram is that transitions in both constituent models are synchronous. For synchronization, both models must have the same sequence of transitions and their time advance functions must produce the same results over the entire simulation. There is no reason to assume synchronization between

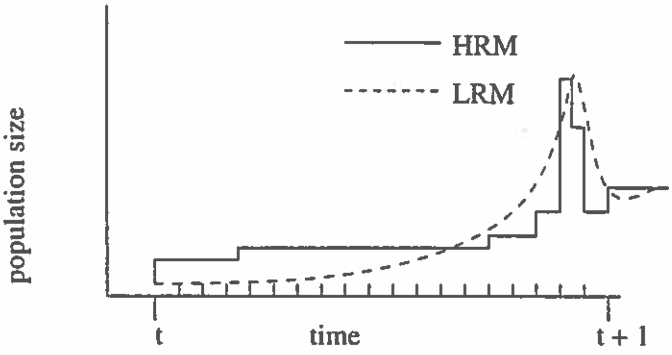

Figure 12. Asynchronous Time Advance

arbitrary models. Consider the example in Figure 12. The state transition for both the LRM and HRM produce identical results at times *t* and *t*+1, however, the state of the low-resolution model is defined only at *t* and *t*+1. If the MRM attempts to resolve state differences between LRM updates, it must supply a method for approximating the LRM state and it must supply a modified transition function that accounts for unanticipated time advances.

Figure 11 shows the consistency relationships between the constituent models of an MRM and Figure 12 shows the impact of asynchronous time advance functions. These diagrams suggest requirements that the MRM must meet if it is to produce valid results. Specifically, the MRM must:

- include a means for translating the state of the constituent models (state consistency),
- include a means for resolving state differences (behavioral consistency),

- include a means for resolving differences induced by temporal scale (segmentation consistency),
- include a means for resolving differences in input values (cross-resolution consistency),
- include a means for resolving differences between network and model states (organization consistency).

### 3.3.1.1 State Translation

To understand state translation, consider an MRM consisting of a logistic equation as the LRM and a logistic IOM as the HRM. The logistic equation is a discrete-time finite difference model of population dynamics and the IOM is a discrete-event network of individual models. The logistic equation maintains a single state value, i.e., population size, that changes at regularly spaced time intervals. The individual-oriented population model determines the population size by "taking a census" of the individuals belonging to it, but the population dynamics are based on the behavior of individuals.

Assume that these behaviors are reproduction and survival and that they are dependent on the individual's weight, then the model must include a state variable for weight. The IOM population is a set of individuals. Therefore, the IOM's state can be described by the pair (*alive*, *w*), where *alive* is an implicit state indicating that the individual is a part of the population, and *w* is the weight of the individual. The aggregated network state can be separated into two sets, the census set, whose size is the population's census, and the weight set. The size of the census set is the population's size and the sum of the values in the weight set is the total biomass of the population. As the logistic equation uses the census to compute its behavior, the MRM must augment the LRM to compute dynamic changes to the weight distribution to ensure consistency between the models. The augmented logistic equation will maintain the same set of values as the IOM population model, so state translation is a matter of forcing both models to have the same state values. It is important to note that forcing the LRM and HRM to have the same values means that the LRM must have the same set of state values as the IOM. It does not imply that the number of individuals in the IOM is consistent with then state of the IOM. This problem must be handled separately.

In some cases, the state of one model is functionally related to the state of the other model. Consider an LRM that represents a population as a distribution of individual lengths and population dynamics as a function of the distribution. If there is a relationship between length and weight, then the MRM can impose state consistency between the LRM and the previously describe HRM, using this relationship. That is, state translation maps one model's state to that of the other by converting weight to length, or vice-versa. Ideally, this mapping is a bijection, so the translation from the HRM is the inverse of the translation from the LRM, however, the states are not required to be exact. The translations must ensure validity, so the translations do not be inverses of each other.

Given the states of the LRM and HRM at time $t$, $S_{LRM,t}$ and $S_{HRM,t}$ respectively, let $RES$ be a Boolean value, where $RES = true$ when the HRM is active and $RES = false$ when the LRM is active. At time $t$, state of the MRM is $(RES \wedge S_{HRM,t}) \vee (\overline{RES} \wedge S_{LRM,t})$. At every simulated time, $S_{HRM,t} = f(S_{LRM,t})$ and $S_{LRM,t} = g(S_{HRM,t})$, where $f$ is a valid translation from $S_{HRM}$ to $S_{LRM}$ and $g$ is a valid translation from $S_{HRM}$ to $S_{LRM}$. If $g = f^{1}$, then we can ensure consistency. If one model is stochastic, and $g$ is the inverse transformation of the distribution function $f$, then the simulation should be statistically valid. Any other condition is up to the user to validate.

### 3.3.1.2 Behavioral Consistency

Behavioral consistency is the type of consistency implied in Figure 1. A state transition in one model must produce the same state as the equivalent transition in the other model. If the LRM and HRM are deterministic and equivalent, then this condition should hold. If, however, either the LRM or HRM is stochastic or the transitions are not equivalent, Figure 1 does not hold. In this case, the MRM must provide a function for resolving differences between the constituent models after a transition. The nature of the function depends on how the MRM uses the constituent models.

The MRM can allow both models to run simultaneously. After each transition, the MRM must resolve state differences between the two models through some process, such as averaging the state values. The MRM can also switch control between the constituent models, which means that state and behavioral consistency are equivalent problems. For example, in the MRM consisting of the logistic model (LRM) and the equivalent IOM (HRM), the LRM can control the state transition and time advance when the population is large and the HRM can control when the population is small. The model responsible for controlling transitions and time advance is the active model and the other model is the passive model. When the MRM switches control, the state of the passive model is forced to match the state of the active model.

The model suggested by DeAngelis and Rose is a switching model. The LRM is active for some portions of a simulation while the HRM is active for others. As the DeAngelis and Rose suggestion is the basis of this dissertation, the remainder of this chapter will consider switching models.

### 3.3.1.3 Segmentation Consistency

A model's state between transitions is defined as a state segment, so the purpose of the time advance function, $t$, is to compute the time interval of the segments. Each model assumes a set of allowable segments based on the model's time scale, where segment size is based on the modeler's assumptions regarding the temporal behavior of the system. Again, consider Figure 13. The LRM state segment on the interval $(t,t+1)$ is constant and the definition of the LRM transition function is based on the state remaining constant on this interval. The HRM makes several state changes on this interval and its transition function is defined so that it makes these changes at the specified times. If the MRM switches from the HRM to the LRM at time $t_x$, where $t_x \in (t,t+1)$, then the MRM must resegment the LRM, that is, the MRM must estimate the state of the LRM at time $t_x$ and approximate the transition function for the resegmentation.

To support a model switch between updates, consider the behavior of the time advance function. Let $\tau_L(q_{t,L}) = t+1$, where $\tau_L$ is the low-resolution model's time advance function and $q_{t,L}$ is the state of the low-resolution model at time $t$. If the high-resolution model switches at time $t + \tau$, where $0 < \tau < 1$, then we can claim that $\delta_{LRM}(x_{[t,t+1]}, q_t) = q_{t+1}$, where $x_{[t,t+1]}$ implies that the system is changed by an external interaction at time $t + \tau$. However, the low-resolution model does not claim validity at that time and, in fact, we know that $\delta_{LRM}$ is not valid at that point. If the user can produce a valid transition function $\delta_{reseg}$ for this situation, where $\delta_{HRM}(X_{t+1}, q_t) = \delta_{HRM}(X_{t+1}, \delta_{reseg}(x_{[t,t+1]}, q_t))$, then the MRM can support arbitrary switching time. The function, $\delta_{reseg}$, is called a resegmentation function.

### 3.3.1.4 Interaction Consistency

A model interaction is represented by an external transition function, $\delta_{xRM,\, ext}$ where $xRM$ is either the LRM or the HRM, and an input mapping from the source of the interaction to the destination. The effect of an external transition is a change in the state of the system and the time advance of the affected model. When an interaction happens across resolutions, the external transition function is no longer valid as it is designed to accept input from a model at the same level of resolution.

The state and behavior of a model are altered by interactions with other models. Specifically, $\delta_{xRM,\, ext}: X \times Q \rightarrow q$ where $X$ is the model's input and $Q$ is the model's state. In a cross-resolution interaction between a model at resolution $x$ and a model at resolution $y$, the external transition becomes, $\delta_{xRM,\, ext}: X_{yRM} \times Q_{xRM} \rightarrow Q_{xRM}$, where $\delta_{yRM,\, ext}$ is the external transition function with resolution $x$, $Q_{xRM}$ is the state with resolution $x$ and $X_{yRM}$ is the input value from resolution $y$.

There are two approaches to resolving this problem: to change the functional form of the external transition, and to translate the input values from the $y$ resolution to the $x$ resolution. The first approach leads to a new definition of the external transition $\delta_{ext} = \delta_{1x,2y}(X_{1x}, Q_{2y})$, where $\delta_{1,2}$ is the external transition between model 1 and model 2. The other approach is to provide a translation function $\Theta$ that translates the request from one resolution to the other. The user should be able to select one or both of these approaches based on the representational sense of the model.

### 3.3.1.5 Organizational Consistency

The fifth source of state inconsistencies, as suggested by diagram 3-7, is inconsistencies involving networks. Again consider the MRM consisting of a logistic model and its equivalent IOM. If the model switches from the logistic model to the IOM, then the MRM must impose consistency between the logistic model's state and the IOM's aggregate state. That is, the MRM will force the logistic model to match the aggregate state of the IOM, it will not, however, force the aggregate state of the IOM to match the aggregation of the IOM's component models. The MRM must impose consistency between the IOM and its component model by disaggregating the state of the IOM.

Consider an aggregate model. The model represents the average-case state of the system ($\overline{q}$) as the average state of the system's entities ($q_0, q_1, \ldots, q_n$). If the model is allowed to switch between a fully aggregated state and a fully disaggregated state, Then the user must define two valid switching operations θ: $Q_H \rightarrow Q_L$ and Θ: $Q_H \rightarrow Q_L$ where θ aggregates all or part of the high-resolution representation and Θ disaggregates all or part of the low-resolution state. In the latter case, disaggregation may lead to several possible configurations. If the model requires a unique configuration, the modeler must impose a set of rules to produce the correct configuration, or this technique will not work. The underlying assumption of this approach is that the user defines valid switch functions.

### 3.3.1.6 MRM Definition

The consistency requirements (sections 3.3.1.1-3.3.1.5) are used to produce an MRM definition. To maintain consistency between a high-resolution model, $M_{HRM}$, and low-resolution model, $M_{LRM}$, of a single system, the multiple-resolution model must supply a coupling system that coordinates model switching by imposing consistency on the constituent models. $M_{HRM}$ must be a dynamic network that coordinates individual models within the network. $M_{LRM}$ may be either a network or a model. The MRM includes a translation function *f* and its reverse *g* defined in section 3.2.1.1, a transition function $\delta_{MRM}$, defined in section 3.3.1.2, a resegmentation function, defined in 3.3.1.3, an interaction translation function, defined in section 3.3.1.4 and a disaggregation function, defined in 3.3.1.5. The MRM also includes a boolean valued variable called resolution. This variable tracks the current resolution of the MRM's active model.

### 3.3.2 MRM Implementation

The multiple-resolution modeling extension to STEMSS, STEMSS-MRM, implements the MRM definition above. The implementation of the switching and consistency functions should be transparent to the user. That is, the functions required to implement the MRM should not force an alteration of the existing constituent models. To accomplish this goal, STEMSS-MRM implements three additional classes: the MRM

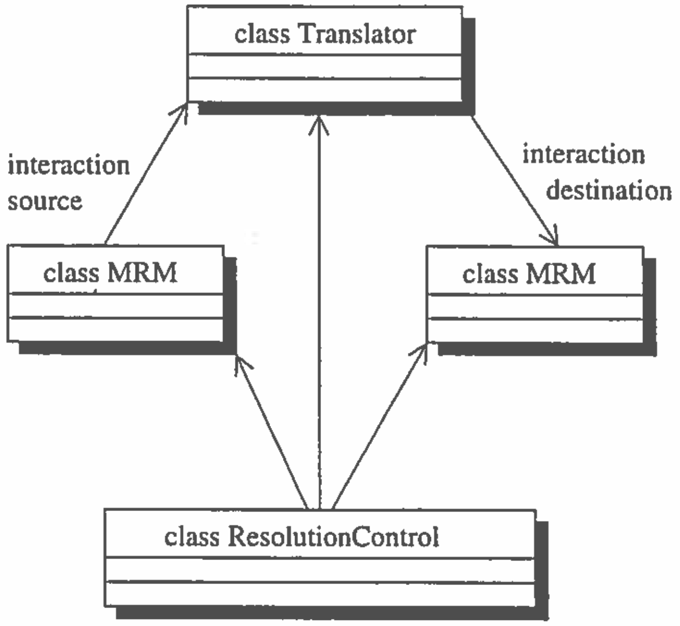

Figure 13. STEMSS MRM Class Structure

class, the Translator class and the ResolutionControl class (Figure 13). The MRM class couples the LRM and the HRM, by implementing a model switching function and by imposing consistency on the state of the constituent models. The Translator class couples map set $\{Z_{ij}\}$, for each resolution level. Coupling resolves inconsistencies induced by cross-resolution interactions. The ResolutionControl is a pseudomodel that maintains configuration control over all MRMs in a given simulation.

### 3.3.2.1 MRM Class

An MRM class represents the LRM and HRM as a single model. So the MRM is, in some sense, an LRM and an HRM. The implication this observation is that the MRM inherits both the LRM and HRM (Booch, 1994; Booch, et al., 1999). By inheriting the LRM and the HRM, the MRM can override the abstract internal transition method of the Event class (recall, from Figure 10, that the Model class inherits the Event class). The internal transition method, implemented in the MRM, invokes the transition function of the active model, then checks if the model has reached a switching condition by invoking the switch detection method. The switch detection method is defined by the user to determine if the model switching criteria are met and, if so, to reconfigure the system by scheduling the ResolutionControl object for execution. The MRM interaction diagram is shown in Figure 14.

An MRM changes state when the EventLoop invokes the internal transition of the MRM's active model. The MRM overrides this method with its version of the internal

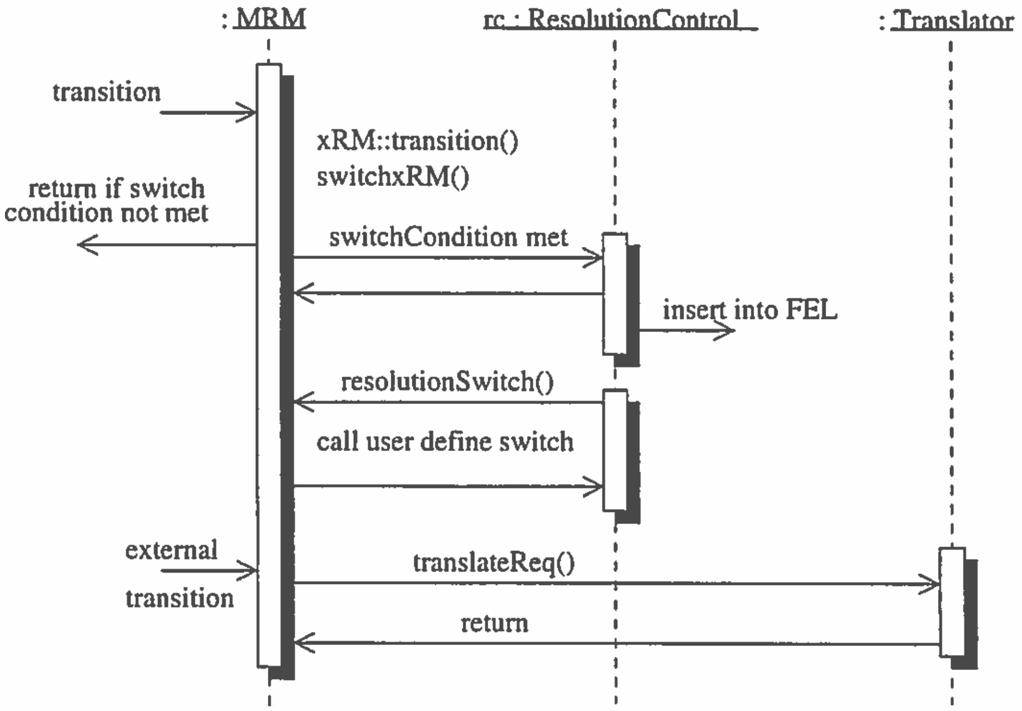

Figure 14. MRM Interaction Diagram

transition. The MRM transition function invokes the active model's transition function (the resolution operator ensures that the appropriate transition method is invoked) then tests for the switch condition. If the condition is met, the MRM schedules the ResolutionControl to execute the appropriate switch and terminates. The ResolutionControl transition method invokes the scheduled resolution change, which removes the active model from the *fel*, and causes the MRM to reconfigure the active model. The class diagram for MRM transitions is shown in Figure 15.

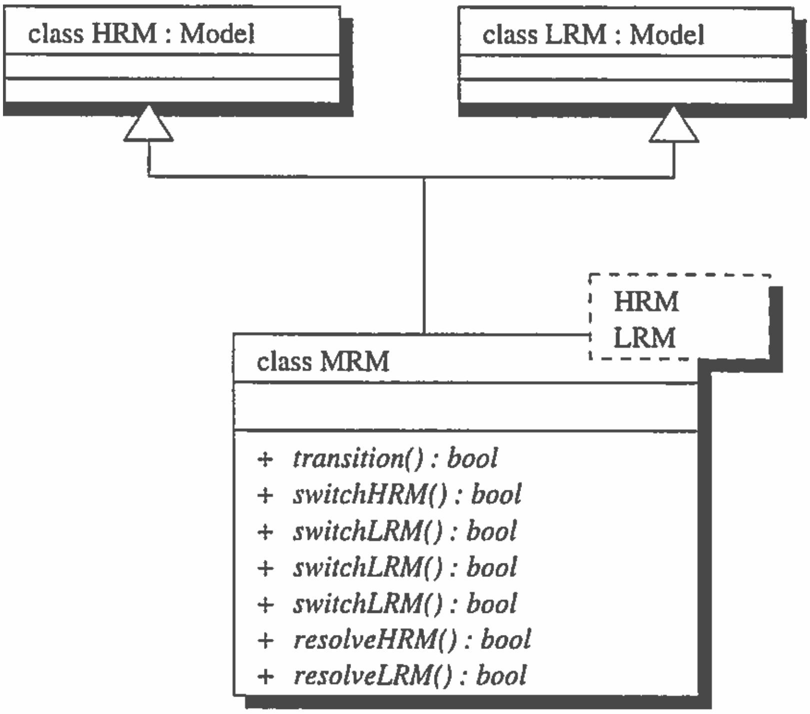

Figure 15. MRM Class Diagram

### 3.3.2.2 Translator Class

The Translator class is used to impose interaction consistency in the same way that the MRM class imposes state and behavioral consistency. As with the MRM class, the Translator inherits the LRM and HRM Zmaps and overrides the interaction methods of the constituent Zmaps. When a single-resolution model maps data to the Zmap, the Translator traps the interface method, translates the input to the resolution of the destination model and maps the data to the Zmap with that resolution. The user may also configure the Translator to map data to both Zmaps.

### 3.3.2.3 ResolutionControl Class

The ResolutionControl object is a pseudomodel that controls all aspects of resolution switching. When an MRM switches resolutions, it schedules the ResolutionControl object for immediate execution. The resolution controller notifies all relevant classes of the switch (e.g., Zmaps and influencing models) and it invokes the MRM's switching mechanism.

### 3.3.2.4 Difference between MRE and STEMSS-MRM

The most fundamental distinction between STEMSS-MRM and the MRE is the role of the MRM in model interactions. The MRE design replaces the LRM and HRM interfaces with the MRE. The user must define mapping functions to provide "public views" for each of the constituent models. The MRE developer also might need to alter the LRM and HRM behavioral representations to accommodate cross-resolution interactions, as the MRE does not provide a method for dynamically changing behaviors. If a static change in the behavior of the model is not desirable, or is invalid, then the MRE is not a good option.

The STEMSS-MRM implementation gives the user access to the complete representation of the model, so the MRM modeler can define specific changes to the model's state and behavior in response to the current configuration of the model. The user can therefore opt to impose consistency by manipulating the state of the system, by manipulating the behavior of the system or by manipulating both state and behavior.

The MRE design concept does not deal with the internal transitions of the active model. That is, the MRE concept does not explain how to resolve differences between constituent models when the changes are internally produced. There are two possible solutions to this problem: make the MRE (i.e., the coupled model) the active model or switch between resolutions. Regardless of the choice made by the modeler, he or she must alter the original LRM and HRM to support interoperation with the MRE (Natrajan, et al., 1997). The STEMSS-MRM framework does not require any changes to the original models, but the MRM may augment the existing model. Any additional behaviors necessary for supporting model switching or state resolution can be defined in the MRM and Translator classes.

### 3.3.3 STEMSS Population Modeling Interface package

The simulation and modeling packages represent arbitrary models, however, the structures provided by these packages require the modeler to adapt their modeling concepts to the conceptual framework of the system. Forcing modelers to adopt the

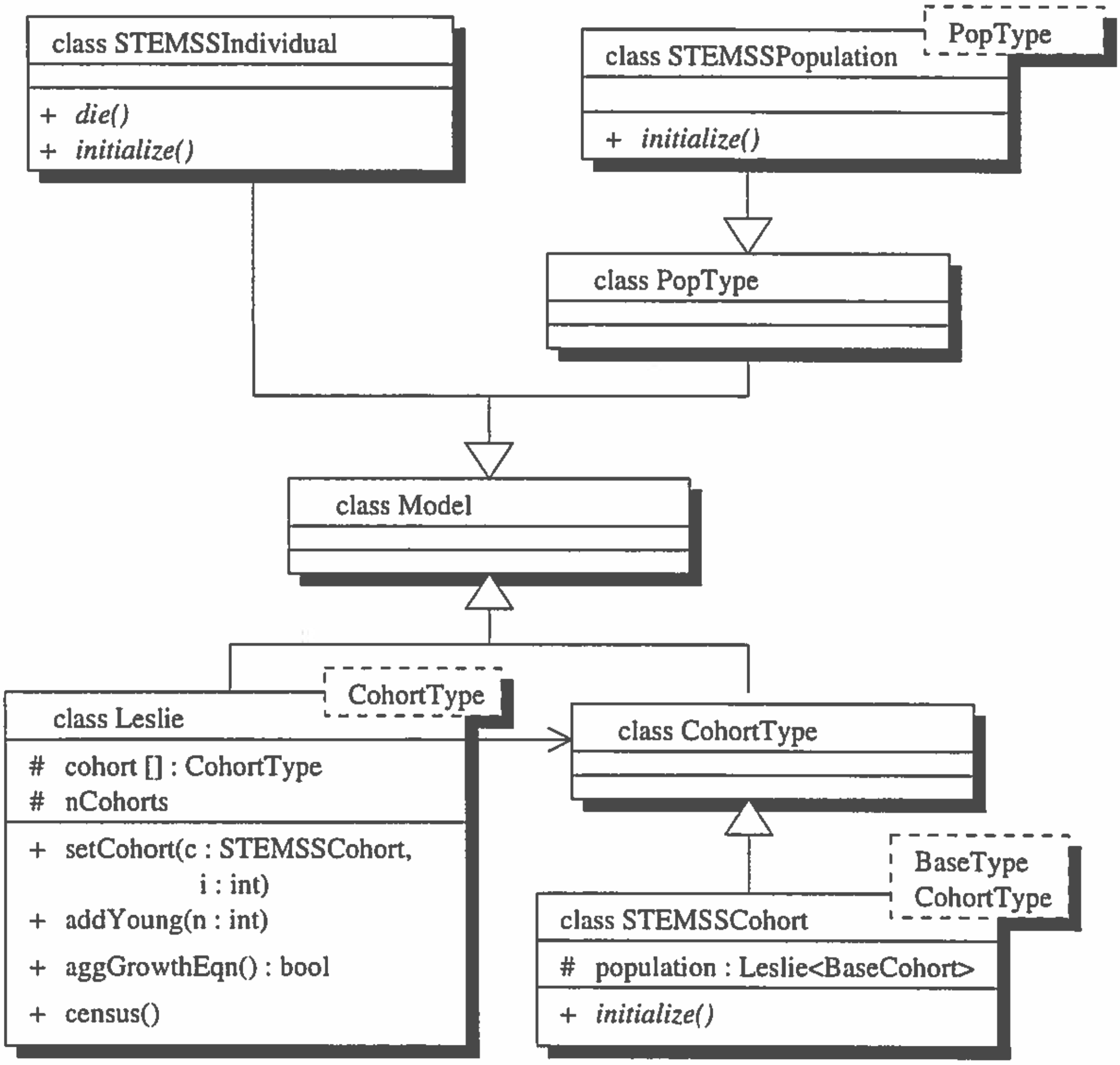

Figure 16. STEMSS Ecological Base Classes

nuances of a particular system has not been a successful design principle in past projects, so it is unlikely to be successful now. Therefore, STEMSS provides a set of modeling classes that implement common population modeling concepts. These classes are extensible allowing the modeler to adapt these classes to specific problems.

Section 1.6 presents some general population representations including aggregate, partitioned and individual-oriented models. Each representation is either based on or uses general ecological concepts such as population, cohort and individual. STEMSS supplies a set of base classes that represent the general behavior of these population modeling concepts (see Figure 16). Using these base classes, STEMSS derives a set of specific classes that represent aggregate models, such as the logistic equation, partitioned models such as the Leslie matrix, and IOMS. Rather than elaborating on the features of these classes, they will be described in the context of actual examples in the next section.

## 3.4 Single-Resolution Models

To develop a model with STEMSS, the modeler must describe a system in terms of the behaviors and influences that change the system significantly. For example, in the logistic IOM, the behaviors of interest are individual reproduction and survival. To create this description, the modeler must first identify the changes of interest and identify the factors responsible for these changes. This description can take the form of a state machine (Papadimitriou & Lewis, 1981). Each change is a transition and the change in state is the result of an internal or external transition. The diagram in Figure 17 represents models and networks by rectangles and their associated transition states as circles. Internal transitions are represented by arrows leading from one state to the next and, if appropriate, the conditions for the transition are included. An external interaction is represented as an

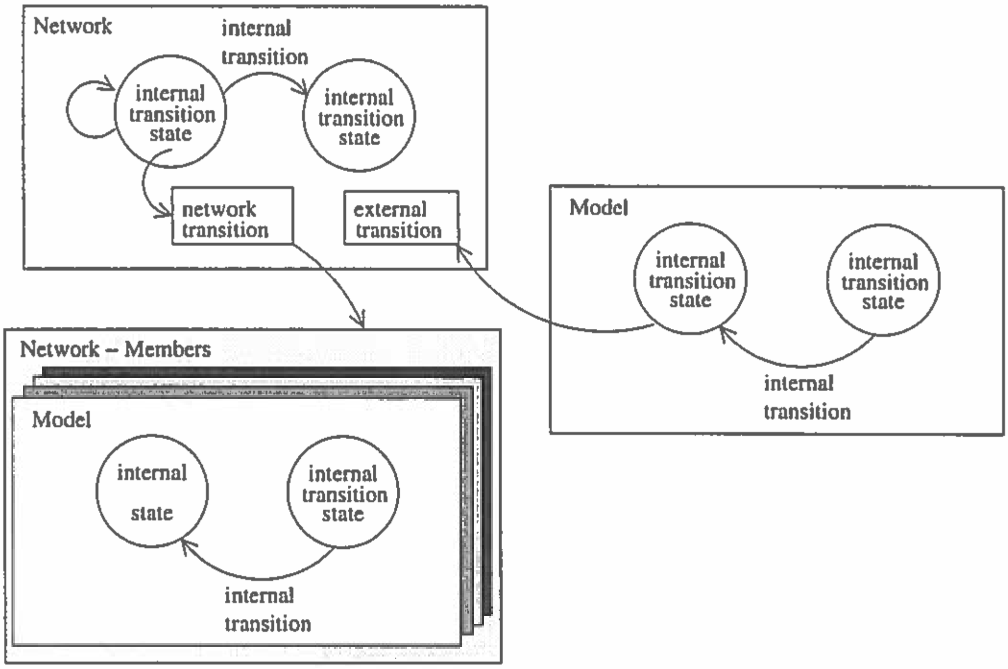

Figure 17. Model Transitions

arrow from the source model to any appropriate states. To represent network behavior, the network model includes a set of ports, represented by small rectangles within a network. For each transition and port, the modeler must identify the state and inputs required to implement it.

To specify the time-advance function, the modeler can use a modified UML interaction diagram (see Figure 5). The transition diagram shows the sequence and timing of actions required to perform an operation or a set of operations. The sequence of actions flows downward from the top, however, it does not imply timing. To specify timing information, the interaction diagram is decorated with specific timing statements.

### 3.4.1 Logistic Model

The logistic model is an aggregate population model. The model's behavior, described in Figure 18, consists of a single transition, *growth*, which is implemented as a modified logistic equation:

$$N_{t+1} = RN_t\left(1 - \frac{N_t}{U}\right) \tag{12}$$

where, $R = r + 1$ and $U = KR/r$. $R$ is the population growth rate and $U$ is the upper bound on population size, that is, if a population exceeds $U$ it will go extinct.

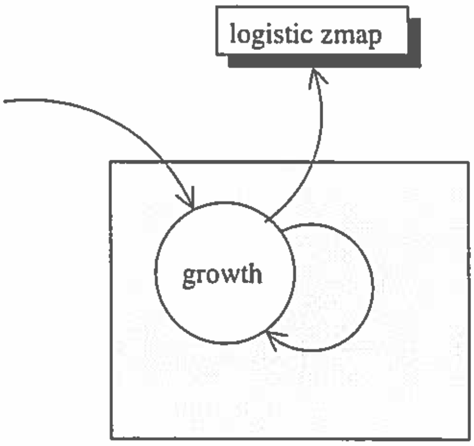

Figure 18. Logistic Model Transition

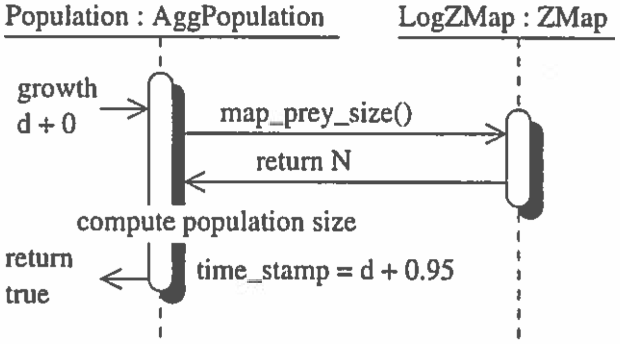

Figure 19. Logistic Model Timing

The model's state is the current population size, $N_t$, from which it computes the population size at the next time step $N_{t+1}$. The population size is mapped to the logistic Zmap, which records the current time and the population size. The timing diagram for the logistic model is shown in Figure 19.

### 3.4.2 Logistic IOM

The individual-oriented version of the logistic model consists of two basic model types, an IOM population model and a basic individual model (see Figure 20). The IOM population, LogIOM, is a network of individuals that allows individuals to become part of the population through birth and removes individuals through culling as part of the individual's life cycle. The LogIOM growth transition records the current population size

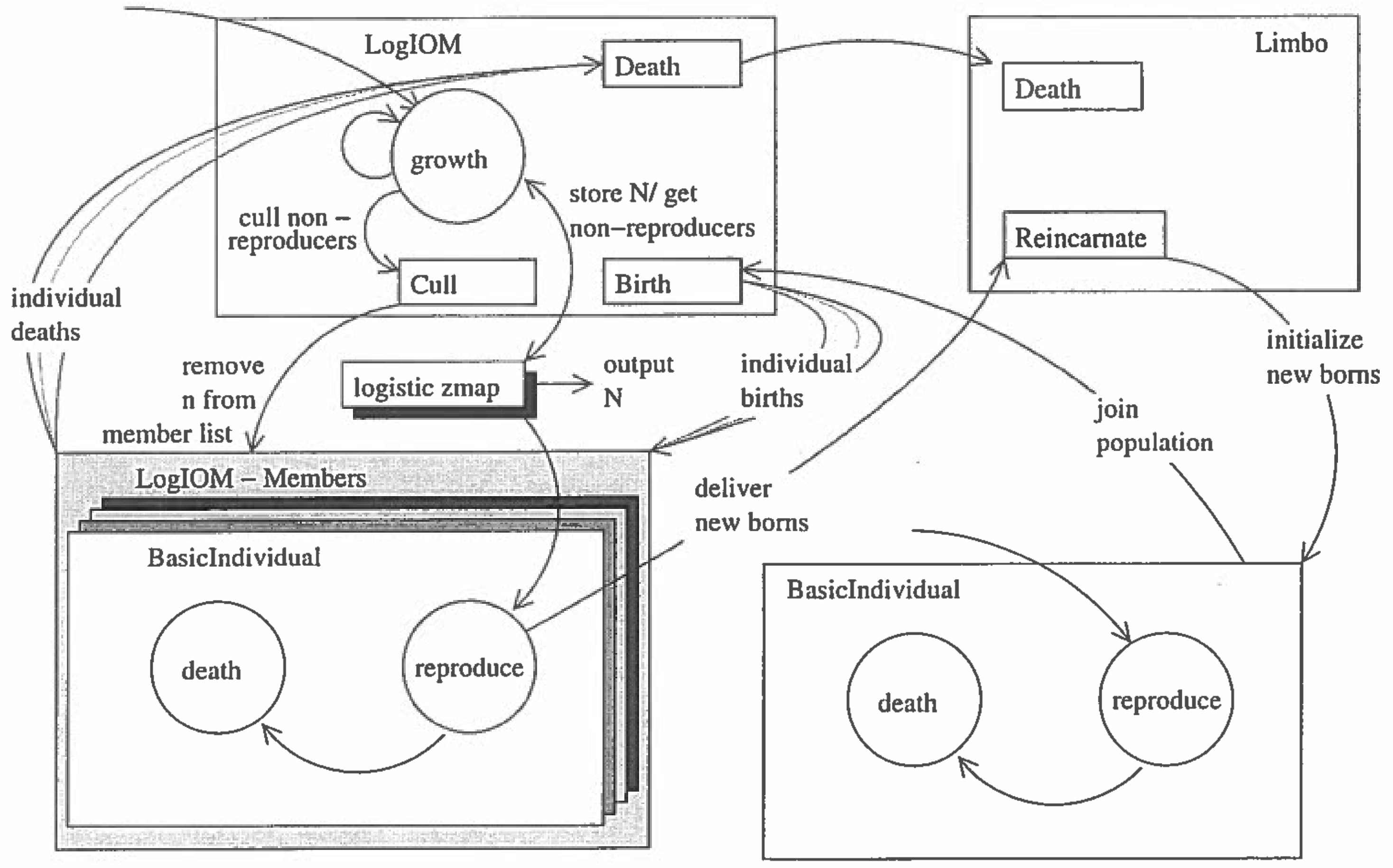

Figure 20. Logistic IOM Transitions

in the logistic IOM's Zmap, LogIOMZmap, which uses the population size to project the future population using the logistic equation (Equation 12). The model assumes the population birth rate is constant and equal to the population growth rate, so the density-dependent effects are accounted for by mortality. To account for density-dependence, the LogIOMZMap determines the number of individuals required to produce the next generation, $N_{repro}$ and removes the excess. The Zmap computes the number of reproducing individuals using:

$$N_{repro} = \left\lfloor \frac{N_{t+1}}{R} \right\rfloor \tag{13}$$

To account for numerical deviations due to truncation error, the adjusted birth rate computed as:

$$R_{adj} = \frac{N_{t+1}}{N_{repro}} \tag{14}$$

Individuals surviving the population's culling event reproduce then die. The individual begins the reproduction event by accessing the adjusted birth rate, $R_{adj}$, from the LogIOMZmap. The truncated value of $R_{adj}$ is the number of offspring produced by the individual, but, in truncating this value, the number of individuals actually produced differs from the number projected by the Zmap. To account for these differences, the individual model, in conjunction with the Zmap, accumulates the fractional part of $R_{adj}$.

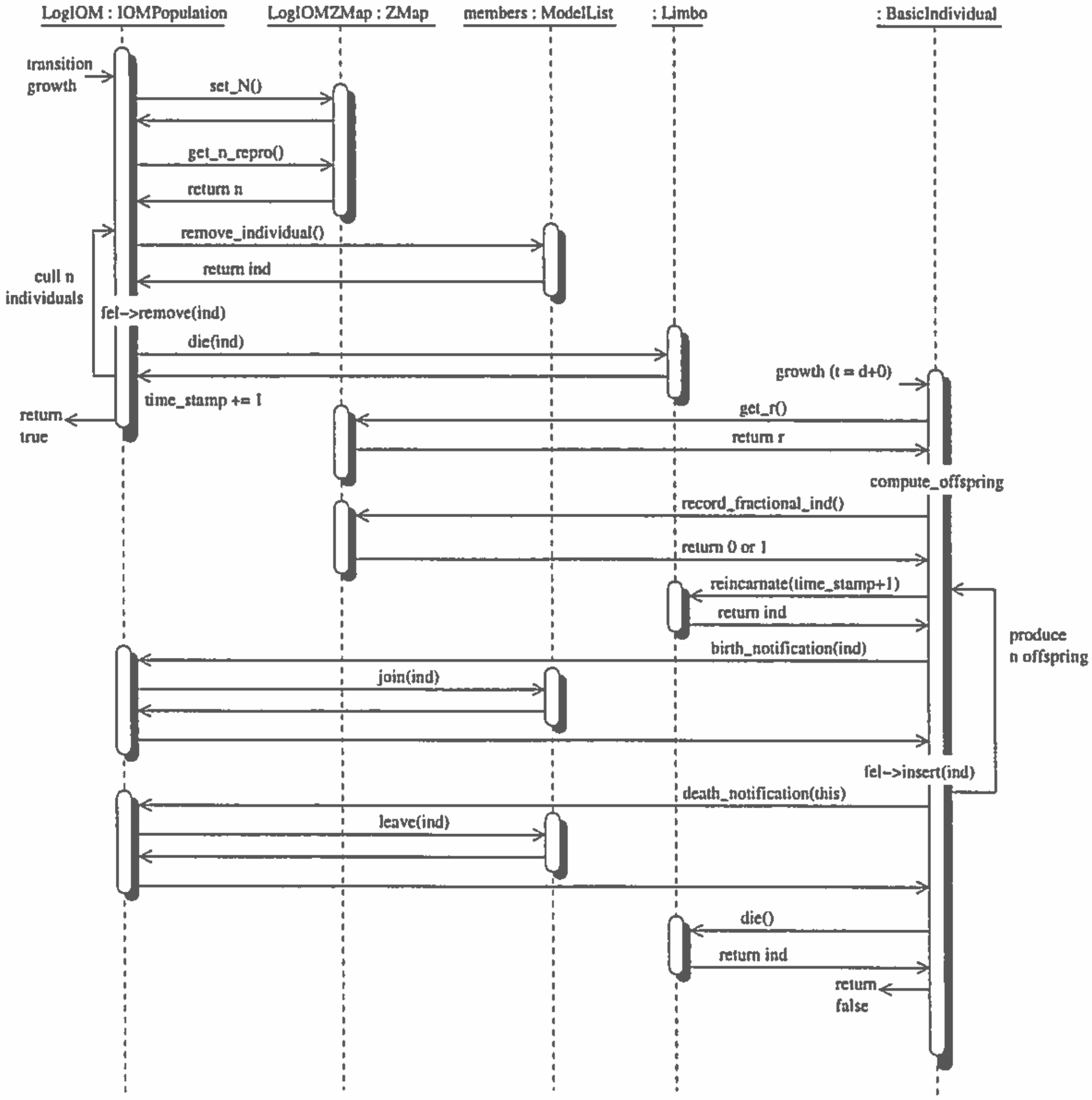

Figure 21. Logistic Model Interaction Diagram

When the accumulated value exceeds one, the individual produces an additional offspring. This procedure allows the individual-oriented logistic model to maintain numerical consistency with the aggregate model. The population timing diagram is shown in Figure 21.

### 3.4.3 Predator-Prey

The predator-prey model is based on density-dependent predator prey equations:

$$P_{t+1} = kP_t\left(1 - \frac{hP_t}{Nt}\right) \tag{15}$$

and,

$$N_{t+1} = RN_t\left(1 - \frac{N_t}{K}\right) - \frac{P_tN_t}{(1+B)} \tag{16}$$

The first equation represents the density dependent growth rate of the predator population and the second represents the density-dependent growth rate prey population minus the number of prey lost to predation. The predator-prey model consists of two population models: the aggregate predator population model and the aggregate prey population (see

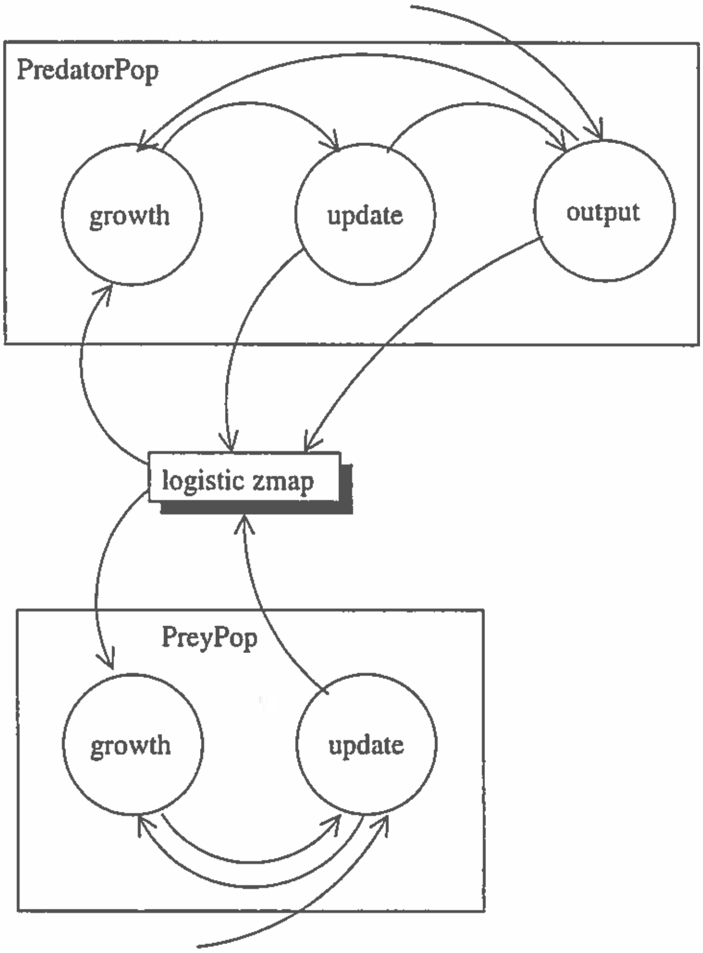

Figure 22. Predator-Prey Model Transition

Figure 22). The growth transition for each model retrieves the current population size of the other model from the PPZmap and computes its new size. The internal transition is changed to *update* and the model is rescheduled for execution. The update transitions record the new population sizes in the Zmap and the predator population model forces an output transition. The interaction diagram is shown in Figure 23.

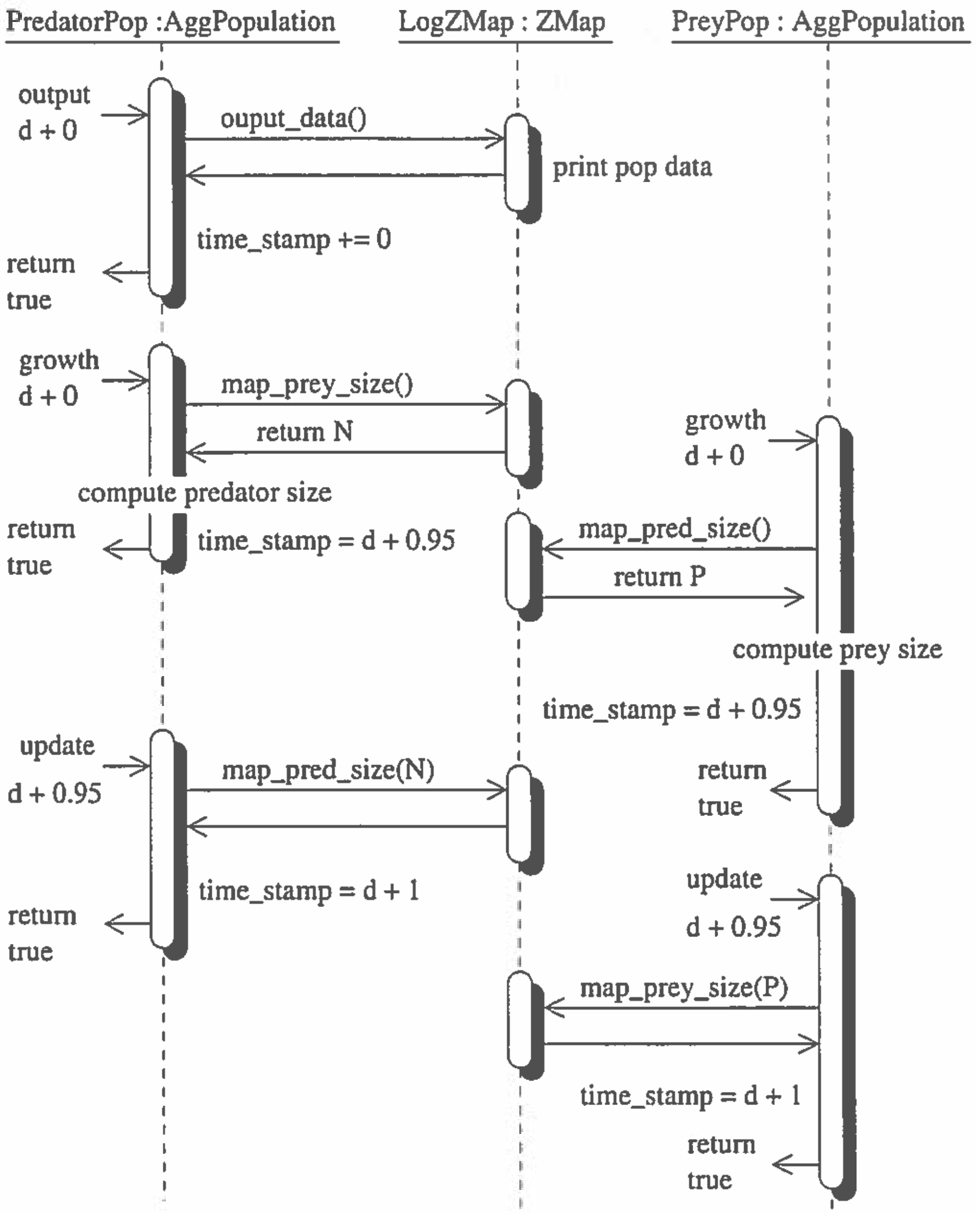

Figure 23. Predator-Prey Model Timing

### 3.4.4 Predator-Prey IOM

The individual-oriented version of the predator-prey model consists of four basic model types: predator and prey IOM population models and predator and prey individual models (see Figure 24). The PredatorIOM and PreyIOM models are identical to the logistic population models. The growth transition, for either model, records the current population size in the Zmap (PPIOMZMap) which computes the number of offspring produced by either population at each time step and the number of prey individuals captured by predator individuals. The number of offspring produced by the predator is predicted by Equation 17,

$$P_{t+1} = kP_t\left(1 - \frac{hP_t}{N_t}\right) - \frac{P_t N_t}{(N_t + B)} \tag{17}$$

where $P_t$ is the population size at time $t$, $k$ is the predator growth rate, $h$ is the upper population bound due to the prey population. The number of predator individuals required to produce the population at time $P_{t+1}$ is:

$$P_{repro} = \left\lfloor \frac{P_{t+1}}{k} \right\rfloor \tag{18}$$

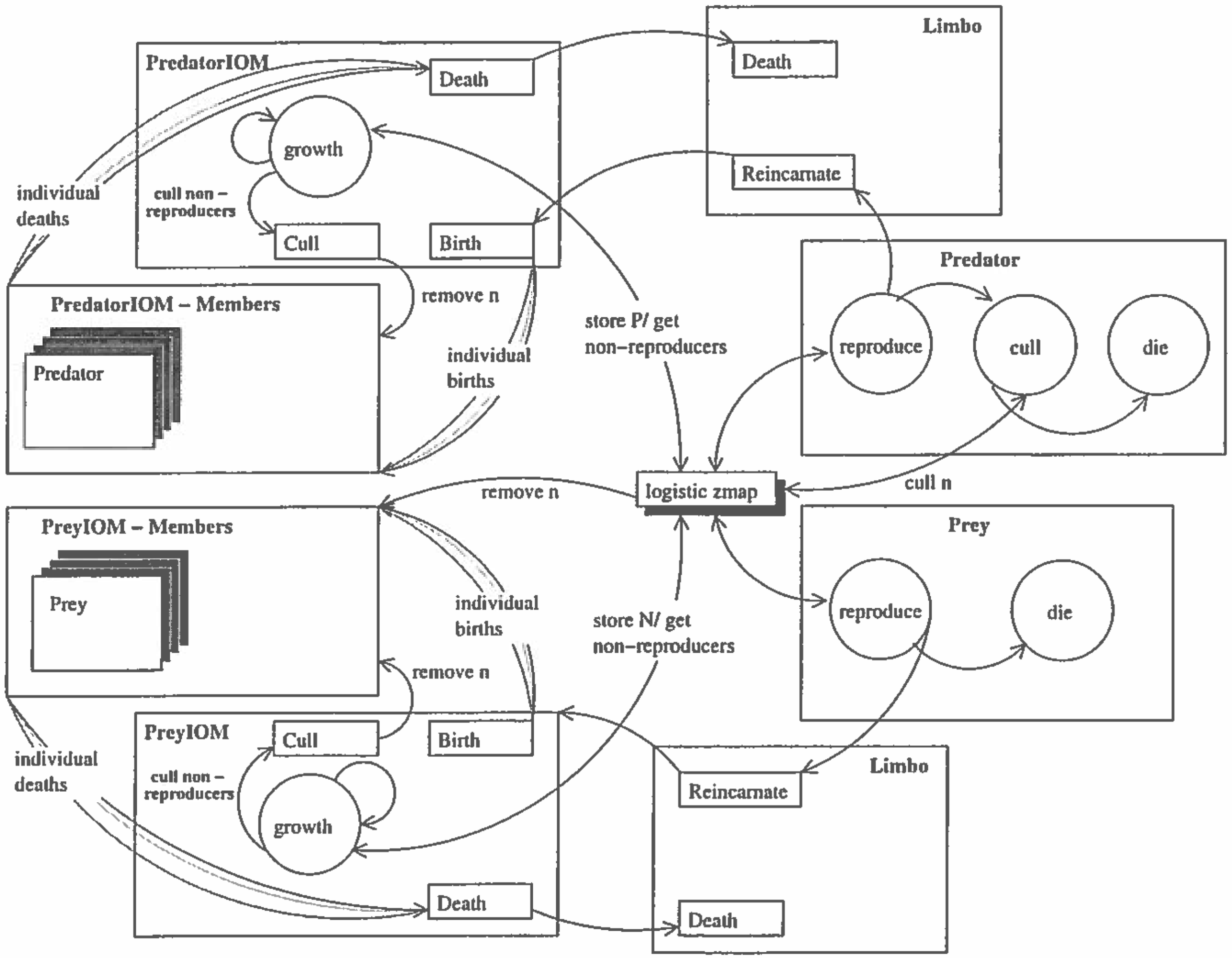

Figure 24. Predator-Prey Model Transitions

The number of offspring produced by the prey population is:

$$N_{t+1} = RN_t\left(1 - \frac{N_t}{U}\right). \tag{19}$$

where $R$ is the prey growth rate and $U$ is the upper population bound of the prey population. The number of prey individuals required to produce $N_{t+1}$ individuals is:

$$N_{repro} = \left\lfloor \frac{N_{t+1}}{R} \right\rfloor. \tag{20}$$

Both $P_{repro}$ and $N_{repro}$ are used to create adjusted birth rates for either population. The PPIOMZMap also computes the number of prey captured by each predator using:

$$n = \left[\frac{A}{(N_t - B)}\right] / P_{repro} \tag{21}$$

That is, $n$ is the number of prey consumed by an individual predator.

Each individual that survives the population's culling event reproduces then dies. The predator individuals cull their allotted number of prey individuals before they reproduce. To ensure numerical consistency, the methods that determine the number of offspring produced and the number of prey individuals captured use a faction accumulator similar to that in the logistic IOM model. The interaction diagram for both populations is shown in Figure 25 and the diagram of predator individuals is shown in Figure 26. The prey interaction diagram is the same as the BasicIndividual sequence shown in Figure 21.

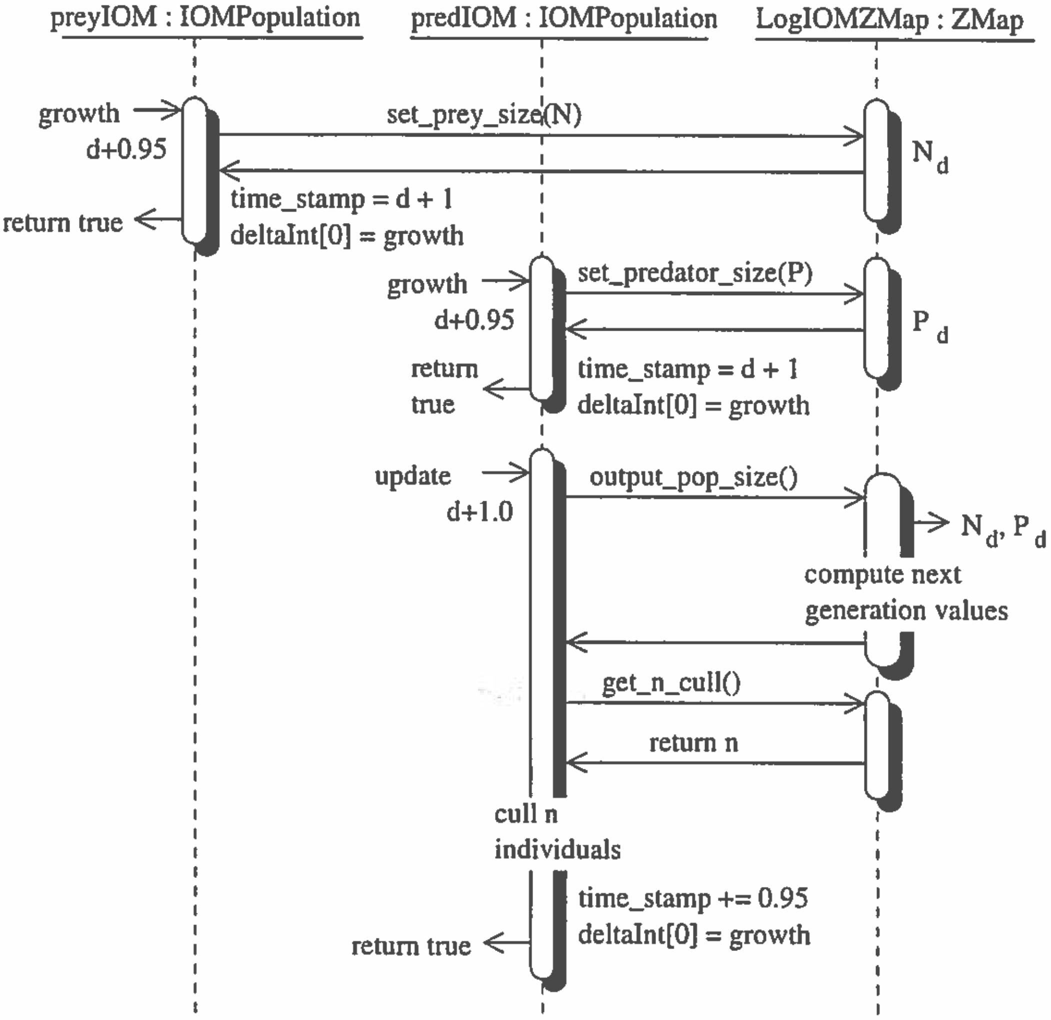

Figure 25. Predator-Prey IOM Model Timing

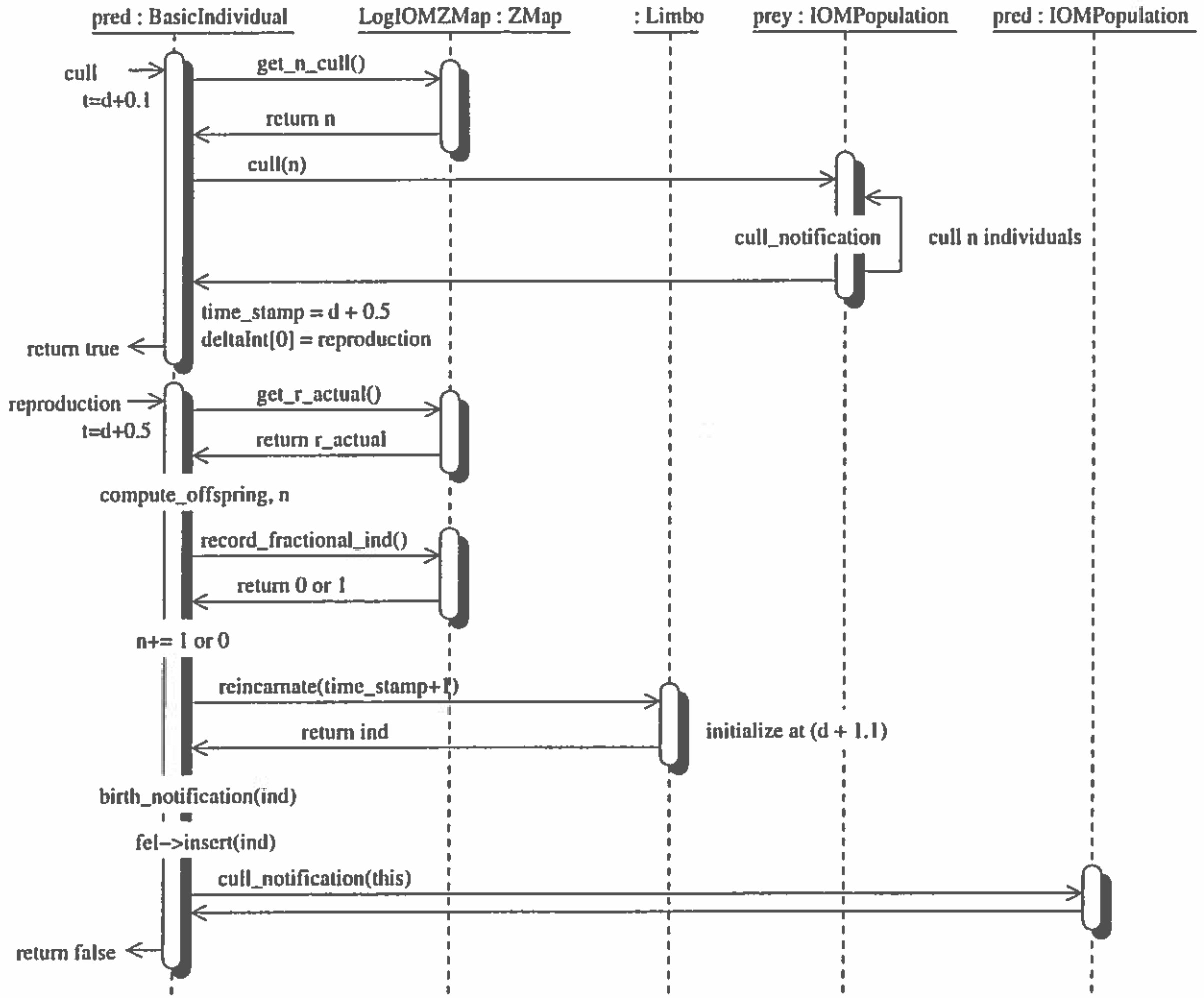

Figure 26. Predator Model Timing

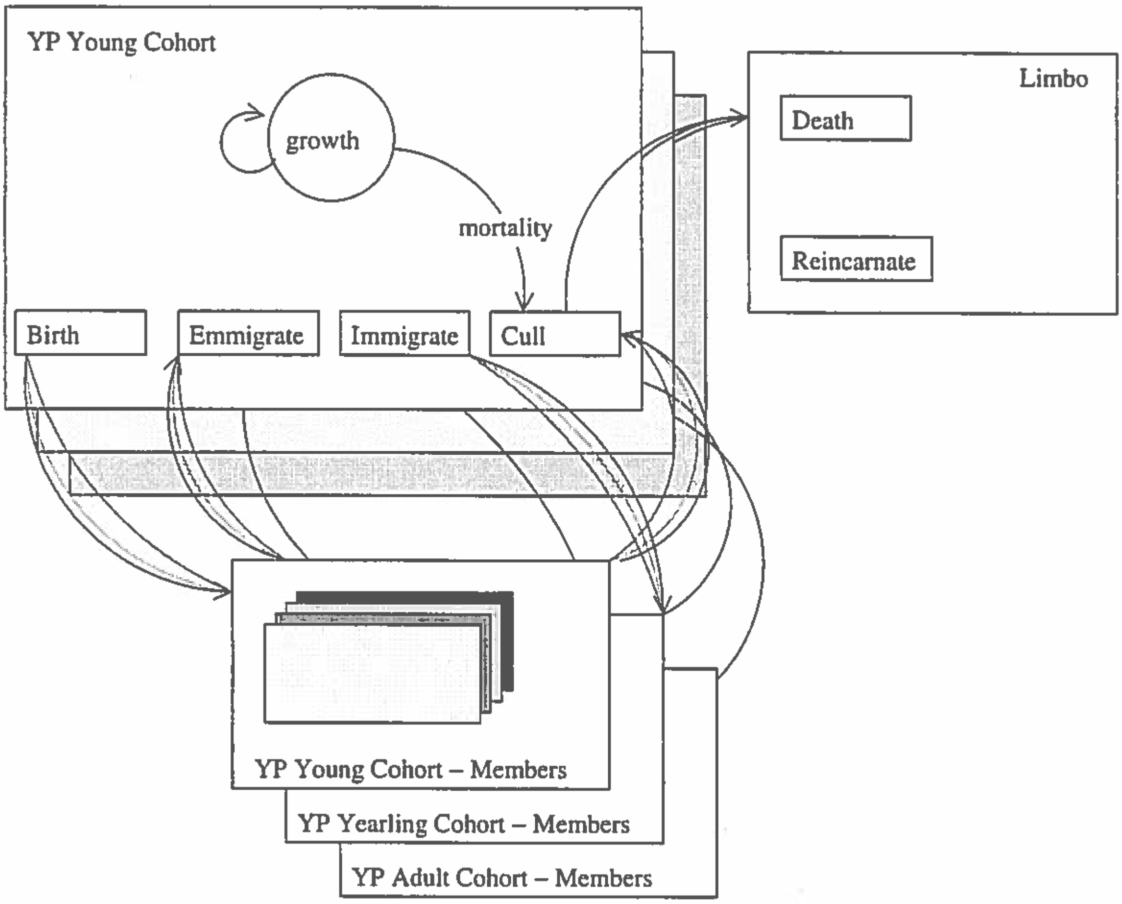

Figure 27. Yellow Perch Cohort Transitions

### 3.4.5 Yellow Perch IOM

The IOM yellow perch model is based on Rose's yellow perch model, described in section 1.5. The population consists of eight cohorts, which are represented as a set of as IOM populations (see Figure 27). The cohorts are partitioned into one young cohort, one yearling cohort and six adult cohorts. The population updates each cohort, on an annual time step, by computing mortality for each cohort and removing that number from the cohort. It then invokes the chronological development transition for each individual, which, for the young and yearling cohorts causes the individuals move from their current cohort to the next older cohort, i.e., the young become yearlings and move to the yearling cohort and the yearlings become adults and move to the first year adult cohort. The adults remain in their current cohort and the cohorts advance in age, that is, adult cohort 0 becomes adult cohort 1, cohort 1 becomes cohort 2, and so on. The individuals in adult cohort 6 are removed from the cohort and the cohort becomes adult cohort 0. In addition to the controlling the development of young individuals, the young cohort also creates the first-feeders for each individual hatching on each day and sorts the young into length classes to facilitate adult cannibalism.

The yellow perch individual model consists of four sets of state machines (see Figure 28). The machines are chronological development, reproduction, consumption and physiological development. On each annual time step, the cohorts cause the individuals to develop, i.e., young become yearlings, yearlings become adults and adults get a year older. Chronological development causes changes in the feeding habits and metabolism of the young and yearling classes and it determines whether an individual, adult female becomes reproductive. As the young and yearlings grow (because of consumption) their predatory behavior change as they cross specific lengths. The first-feeders become young-of-the-year when they reach 20 mm, which changes their success at catching prey, and the young-of-the-year change the type of prey they can collect when they reach 30 mm. Consumption events are processed daily for all individuals and reproduction events are processed annually for reproductive female.

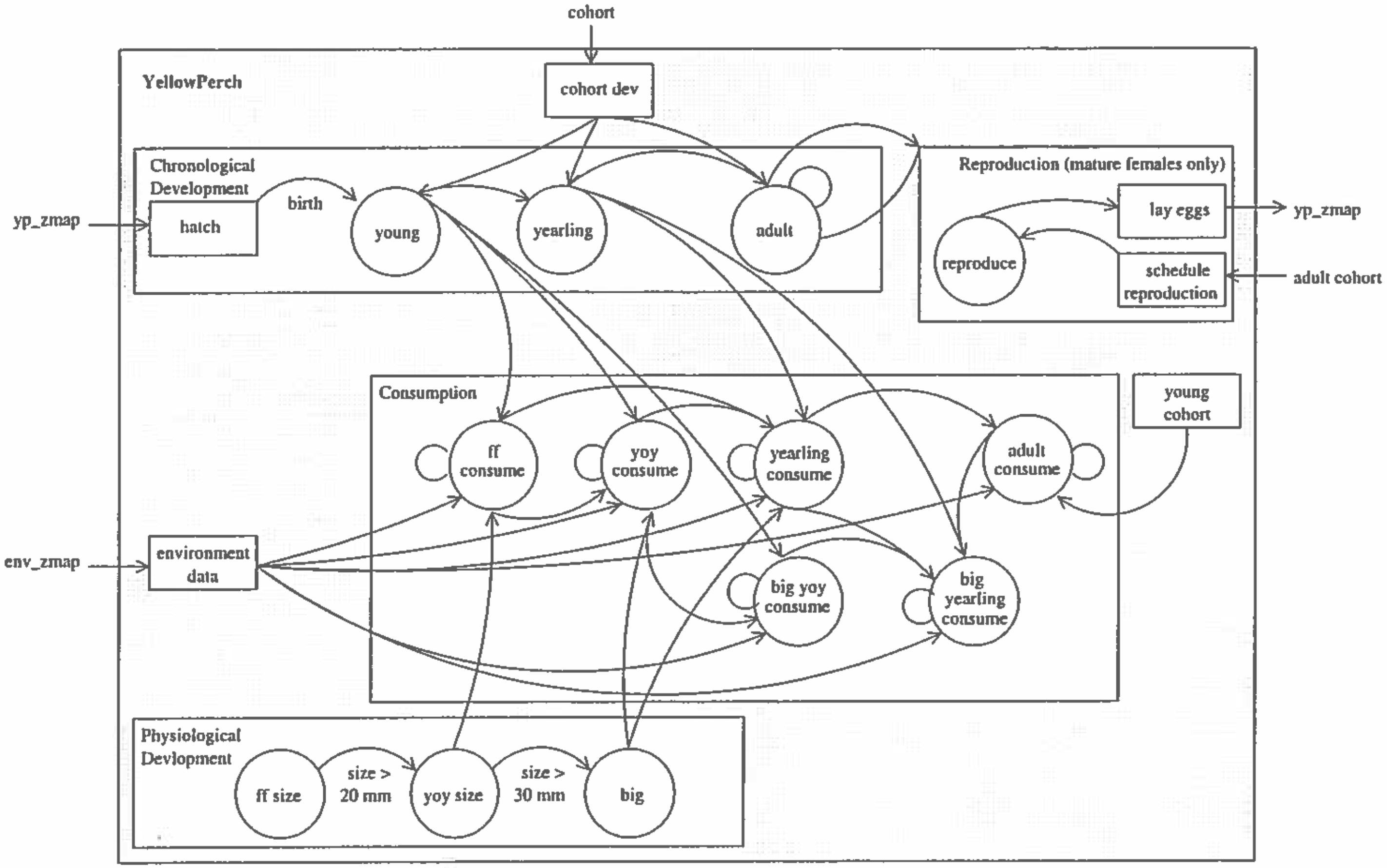

Figure 28. Yellow Perch Individual Transitions

### 3.4.6 Yellow Perch Super-Individuals

The "super-individual" version of the yellow perch model represents the behavior of individuals as a two-dimensional distribution of weight and length. The distribution is represented as a grid and each grid node represents a group of individuals. The growth and reproduction functions are computed for each node using the weight and length of the node. The amount of food consumed and the number of offspring produced is the produce of the individual behavior times the number of individuals at the node.

## 3.5 MRM Examples

Three ecological modeling examples will be used to demonstrate MRM development with STEMSS. The first example uses the logistic population models described in sections 3.2.3.1 and 3.2.3.2. The second example is a density-dependent, predator-prey model based on the models described in sections 3.2.3.3 and 3.2.3.4. The third example is based on the models describe in sections 3.2.3.5 and 3.2.3.6.

### 3.5.1 Logistic MRM

The logistic MRM couples the aggregate logistic model (section 3.3.4.1) and the logistic IOM (section 3.3.4.2), using a threshold population size to determine which model should be active. The aggregate logistic model updates the change in population size, from

$N_t$ to $N_{t+1}$, at time $t$ and the logistic IOM updates the change at time $t+1$. When the switch is made from the aggregate logistic model to the logistic IOM, (LRM to HRM), the population size $N_t$ is mapped to the aggregate model's Zmap and the necessary culling and reproduction parameters are generated. That number of reproductive individuals are "reincarnated" and inserted into the fel and the logistic IOM growth method is scheduled for $t+1$. These actions ensure that the state of the aggregate model is consistent with the state of the IOM. When the MRM switches from the logistic IOM to the aggregate logistic model, the current census of the IOM is $N_t$. The aggregate logistic model's population size must be set to $N_t$ and the individuals in the event list removed and stored in "Limbo."

### 3.5.2 Predator-Prey MRM

The MRM tests, described in Chapter IV, rely on two versions of the predator-prey model. The first version is used to test cross-resolution interactions and the second tests resegmentation. The predator MRM is composed of an aggregate predator-prey model (see section 3.3.4.2) and a predator-prey IOM (see section 3.3.4.3). As with the logistic MRM, a threshold population size is used to determine which model is active. Since the predator and prey populations switch the active models independently, the active models, for both populations will occasionally have different resolutions. That is, at any point, the active models in the predator-prey MRM may have the following combinations: predator IOM-prey IOM, aggregate predator model-prey IOM, predator IOM-aggregate prey model, and aggregate predator model-aggregate prey model. The predator-prey MRM must coordinate interactions between the population models when the active models have different resolutions.

When the predator IOM and aggregate prey model are active, an interaction between the predator and prey models must make the interaction behave like an IOM-to-IOM interaction from the predator's perspective and an aggregate-to-aggregate interaction from the prey's perspective. To accomplish this, the MRM accumulates the predator's requests for prey between population updates. The prey MRM replaces the aggregate prey model's growth method with a version that uses the accumulated predation to update its size, rather than the value produced by the aggregate predator model.

When the aggregate predator model and the prey IOM are active, the predator's interactions with the prey must be translated from an aggregate interaction to an individual-based interaction. To represent aggregate predation, the predator-prey MRM includes a surrogate individual class. The surrogate predator is provided with the number of prey to be consumed over the simulated year after each aggregate predator model update. The surrogate predator determines the distribution of consumption events, based on the consumption behavior of individuals in the predator IOM and schedules itself for execution.

To use the predator-prey MRM as a test of the resegmentation mechanism, the MRM must be modified to allow for model switches at arbitrary times. For the sake of analysis, the resegmentation test for the predator IOM remains active throughout the test, so only the prey MRM involves switching. To create this model, the prey MRM overrides the join and leave methods of the prey IOM to keep track of births and deaths between population updates. If the prey population falls below the threshold value, the prey model switches from the aggregate model to the IOM model. Using the aggregated culling behavior of the predator, the prey model estimates the number of individuals alive at the current time. This number of individuals are inserted into the model. If the prey population rises above the threshold, the number of individuals required to produce the prey population (at the current time) is used as the aggregate prey model's population size.

### 3.5.3 The Yellow Perch Model

The yellow perch MRM couples the yellow perch super-individual model (section 3.3.4.6) and the yellow perch IOM (section 3.3.4.5). As most of the computational effort in the yellow perch IOM is used to maintain young individuals, only the young cohort model switches resolution. The young hatch as a super-individual and remains a super-individual until its size falls below a threshold level, at which point, the size of the super-individual and its weight and length are used to generate individual young perch. As with the predator-prey MRM, when the young are represented as super-individuals, the culling of young by adults is translated to an aggregate request.

## 3.6 Future Refinements

Though the current version of STEMSS implements the design philosophy described in this chapter, the implementation is incomplete. A driving force behind the

STEMSS design is to hide implementation details from the user and the Model and Network classes are largely successful in this attempt. However, the Zmaps are still constructed by hand. Hence, the translation classes must also be constructed by hand. This problem can be overcome by applying the same encoding principles to the Zmap structure applied to the Model and Network classes.

STEMSS currently implements only dynamic networks and hybrid models. Though we can tailor these structures to produce strictly discrete-event, differential equation or discrete time systems, the existing structures require more overhead than the specific structures. A future revision should include a wider range of modeling choices. One of these choices should be a class for adapting models that were not developed with the STEMSS to the STEMSS framework. A wrapper class, called the STEMSSConforming class, is being developed. The STEMSSConforming class inherits the Model class, which allows it to use the STEMSS simulation package. To adapt models to STEMSS, the STEMSSConforming class maps the target model's state information and transition functions to it own state and transition methods, which allows the adapted model to be used in a STEMSS or STEMSS-MRM based simulation.

The STEMSS-MRM and the STEMSSConforming models can be generalized to include a broader model coupling system. The MRM is a specialized model coupler. It is intended to seamlessly resolve state differences between two, possibly disparate, models. There are two, relatively minor differences between an MRM model coupling system and a general model coupling system. First, all models in a coupled model are active. This forces a change in how events are processed. Second, the model coupler does not switch between resolutions. To implement this system, the MRM must be able to distinguish which model's transition function was invoked. The transition function is called when the model's event is processed and the event has a unique identifier. By passing the event's id to the transition function, the MRM can determine which model's transition function was invoked. This will convert STEMSS-MRM to a general model coupling system.

## 3.7 Summary

This chapter described the development of a discrete-event ecological modeling and simulation framework that offers a solution to specific criticisms of ecologists about computer simulation. These solutions include a well-defined structure for model development, an interface for ecological modeling and a facility for developing an MRM. The approach to multiple-resolution modeling developed for this dissertation is novel in its definition of the problem, in its solution to the problem and in its implementation. This chapter developed fundamental definitions to determine what the MRM must do to ensure consistency. These definitions led to a combined state and behavioral representation for ensuring state consistency which was incorporated into STEMSS. The next two chapters will describe the results of the system tests.

# CHAPTER IV

# MRM VALIDATION STUDIES

## 4.1 An Overview of the Tests

The design of the STEMSS framework, the MRM extension to the framework and three sample models were described in the previous chapter. The tests used to validate STEMSS and STEMSS-MRM, and the results of those tests, will be described in this chapter. Each of the validation tests uses one or more model suites (see Table 1), i.e., sets of models consisting of low-resolution, high-resolution and multiple-resolution versions of a particular model. The logistic suite, for example, consists of an aggregate logistic model, a logistic IOM and a logistic MRM. Each test was designed to show the ability of the STEMSS-MRM mechanisms to support the development and use of MRMs and that the MRMs support model consistency. The results of these tests will show three points:

- MRMs can produce time series bounded by their constituent models,
- STEMSS-MRM mechanisms to maintain consistency between constituent models,
- STEMSS supports the development and implementation of ecological models.

Table 1. Test Model Suites

| Model Suite | LRM | HRM | MRM |
|---|---|---|---|
| Logistic | Aggregate finite difference model | IOM equivalent of Logistic model | Switching model, decision based number of reproducers |
| Predator-Prey | Density-dependent Lotka-Volterra, finite difference | IOM predator and prey population models | Switching models for both population using t threshold number of reproducers, models include cross-resolution interactions and resegmentation |
| Yellow Perch | Rose Model of Yellow Perch | Super-individual Yellow Perch | Young cohort is MRM, Yearling and Adults are IOMs. Switching is size based. |

The first test will validate the correctness of the STEMSS-MRM structures using the logistic and predator-prey suites. The test is divided into five sections designed to validate the basic multiple-resolution modeling mechanisms. The first section compares the effect of perturbing the logistic IOM and predator-prey IOM. This test will establish that the models are sensitive to minor perturbations and, therefore, any perturbation induced by the multiple-resolution modeling mechanisms will induce a detectable change in the time series produced by the model. The remaining sections of this test will use the sensitivity of these models to establish that systematic errors are not induced by STEMSS-MRM.

The second test uses the yellow perch suite to test the efficacy of STEMSS in developing realistic multiple-resolution population models. The yellow perch suite consists of the super-individual and IOM versions of the yellow perch model discussed in

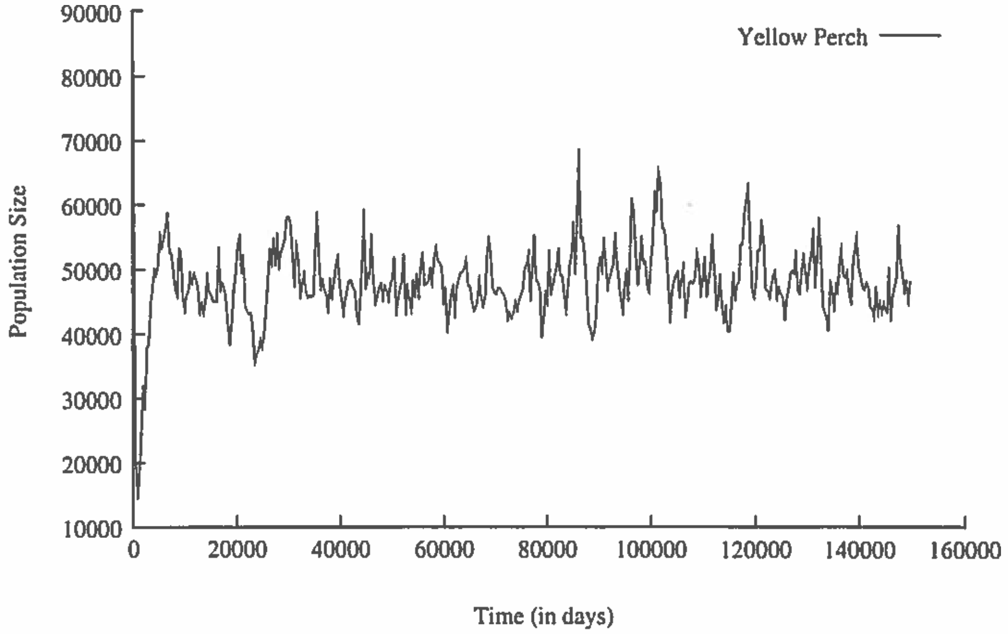

Figure 29. Yellow Perch Simulation

chapter III and an MRM composed of these models. The yellow perch model produces a complex time series (see Figure 29) that potentially require thousands of simulated years to collect a sufficient amount of data for analysis. The 400-year time series shown in Figure 29 took four days to run, therefore, a ten thousand-year simulation may take more than three months to complete. As the complexity, in the time series, is due to stochasticity in the model, the yellow perch test uses deterministic versions of the constituent models to reduce the simulation time. By removing stochasticity, the revised model will not produce the same results as the original model, however, the structure of the revised model, (i.e., the set of behaviors and interactions of the models) are the same as the original model.

Therefore, the revise model should provide a sense of the computation behavior or the yellow perch model.

The third set of tests examines the behavior of the MRM under stochastic conditions using the logistic suite. These tests were designed for two purposes:

- to validate that the MRM is bounded by the LRM and HRM when the time series are stochastic,
- to develop a procedure for predicting relative differences between the HRM and the MRM.

The logistic model suite used in this test is identical to the one used for the first test except that the growth rate in the HRM is stochastic. As a result, the LRM and HRM produce different time series, so if the MRM satisfies the hypothesis stated in Chapter I, the MRM should produce results bounded by these time series. As stochasticity in the HRM alters the nature of time series it produces, the three time series must be statistically compared. Therefore, validating that the MRM is bounded by the LRM and the HRM prompts two questions: what data should be analyzed and how should the data be analyzed?

The time series is composed of two components: amplitude and frequency. This test will focus on the amplitude of the time series, leaving frequency for future work. The MRM time series is bounded by the LRM and HRM time series when the maximum (and minimum) values of the MRM are bounded by the maximum (and minimum) values of the LRM and HRM. The specific relationships between the LRM, HRM and MRM depend on the model, but, without loss of generality, assume that, for some measure of the time series, $HRM_{max} < LRM_{max} < LRM_{min} < HRM_{min}$. Under this condition, the MRM is bounded when, $HRM_{max} < MRM_{max} < LRM_{max} < LRM_{min} < MRM_{min} < HRM_{min}$. As the one or both of the constituent models maybe stochastic, establishing the boundedness relationship requires a statistical technique for comparing the time series measure. The technique employed by in this dissertation is Fischer's Least Significant Difference (LSD) test.

One proposed use for STEMSS-MRM is to analyze a population under emergency conditions. To use the framework effectively, the user must be able to predict execution time and the loss in accuracy. For example, if the model is required to execute within a day, the modeler should be able to find a switching threshold value that allows the simulation to finish execution within the specified time. Then the modeler should be able to use the threshold value to predict accuracy of the MRM. Chapter V will provide a method for predicting the execution time of an MRM and this chapter will provide a method for measuring the consistency of the MRM relative to the HRM (the working assumption in this chapter is that the HRM is the more accurate model).

As a general note, regarding the statistical analysis of simulation results, STEMSS uses a pseudo-random number generator based on Knuth (1998). The sequence of numbers created by the random number generator is autocorrelated, that is, each number in the sequence is dependent on the previous number. As a result, the simulation will not produce a sequence of independent and identically distributed random numbers as required by the statistical analysis techniques used in this chapter. Nevertheless, the use of a PRN is necessary for stochastic simulation and is commonly used in practice.

## 4.2 MRM Testing Environment

The validation tests used in this chapter are the logistic, the predator-prey, and yellow-perch suites (see Table I). Each model was described in Chapter III. When necessary, some of these models were modified to emphasize specific principles related to multiple-resolution modeling. If a model is changed for a test, the changes will be described in that section.

All tests were performed on an HP model Pavillion 7955 with a 1.5 GHz Intel Pentium IV processor and 512 Mb of RAM or Sony Vaio model PCV RX651 with a 1.7 GHz Intel Pentium IV processor and 256 Mb of RAM. Both systems use Mandrake Linux version 8.0 and the code was compiled with gcc version 2.96.

## 4.3 STEMSS-MRM Mechanism Tests

The tests in this section were designed to validate the MRM mechanisms, i.e., the switching, resegmentation functions, and aggregation-disaggregation functions, and cross-resolution interactions described in Chapter III. The logistic model suite was used to validate the switching, and aggregation-disaggregation functions and the predator-prey suite was used to validate the resegmentation functions, aggregation-disaggregation functions and cross-resolution interactions. All models used in this set of tests were deterministic to ensure consistency between the LRM and HRM results and the LRM and HRM were tested to verify that they produce identical time series, under identical parameterizations.

If the LRM and HRM are consistent and produce identical results, the time series produced by the LRM and MRM must be equal to establish consistency in the MRM. If the LRM, or HRM, is sensitive to perturbation and the time series it produces is identical to the time series produces by the LRM, or HRM, then the MRM maintains consistency. The perturbation test will demonstrate that both the logistic and predator-prey models are sensitive to perturbations and therefore, the models are suitable for testing consistency of the STEMSS-MRM consistency mechanisms. The remaining tests validate each mechanism.

## 4.3.1 Models and Methods

### 4.3.1.1 Simulation Models

As described in Table 1, the logistic suite consists of the aggregate logistic model (LRM), the logistic IOM (HRM), and a logistic MRM. The predator-prey suite, consists of the aggregate predator-prey model (LRM) and predator-prey IOM and a predator-prey MRM. These models are discrete state models, that is, their states are discrete valued. The aggregate model uses a real-valued growth rate and a real-valued upper population bound to compute the integer valued population size (see section 1.6 for details regarding the model). The IOM, in contrast, determines population change by counting individuals at periodic intervals and computing the individual birth rate based on the current population size. Each reproducing individual uses this birth rate to determine the number of young it produces.

A numerical error is induced in the IOM due to the discrete nature of the reproduction computation. In the aggregated model, the population size is truncated after the computation is done. In the IOM models, the individual birth rate is truncated then the offspring are produced. The difference in these computational procedures produces a significant difference in model behavior. Consider the following example. Let $R = 1.9$, $U = 5000$ and $N_t = 1000$. Based on the aggregate logistic model, the population size at time $N_{t+1}$ will be 1,520 individuals. To produce the same effect in the IOM, the model culls 200 individuals, leaving 800 to reproduce with a birth rate of 1.9. As 800*1.9 = 1,520, the IOM and the aggregate model should arrive at the same answer. However, each individual uses the truncated birth rate to compute the number of offspring she produces, which means, in this case, each individual produces one offspring. Therefore, the population size predicted by the IOM will be 800 not 1,500.

To impose consistency between the models, the IOM includes a mechanism to collect "fractional individuals." When an individual has a non-integral birth rate, the fractional part of the rate is collected in an accumulator. When the accumulator exceeds one, the accumulator is reduced by one and an additional offspring is produced. As a result, the number of offspring produced by the IOM matches the number of offspring produced by the aggregate model.

#### 4.3.1.1.1 Perturbation Test

The criterion for establishing consistency assumes that any perturbation of the modeled population will induce a change in its time series. A perturbation test using the aggregate logistic and aggregate predator-prey models was developed to establish the sensitivity of both models. In the first step, the aggregate models are compared with their IOM counterparts, which, if they are correct, will produce identical time series with identical parameters sets. In the second step, a single individual is added to either model at simulation time 10. If time series produced by a perturbed model are different from those produced by the unperturbed models, the models are sensitive to perturbation. As this is the smallest possible perturbation, the test shows that any perturbation should change the behavior of the model. If, in the process of switching active models or during a cross-resolution interaction, the MRM produces an inconsistent state (i.e., the number of individuals in the active model is different from the number of individuals in the passive model after the switch), the test will detect it. So, if the MRM is inconsistent with the single-resolution model, its results will differ from the unperturbed model.

#### 4.3.1.1.2 Switch Test

State, behavioral and organizational (aggregation-disaggregation) consistency requires that the state of an MRM's constituent models match at specified times. This allows the MRM to switch between the models while ensuring that the state of the system is not altered by the switch. To test consistency during a switch, the MRM uses the deterministic aggregate logistic and the logistic IOM models to form the MRM. As the models are deterministic, their states should match at each point in time (this is confirmed by the perturbation test) so the MRM's state should match regardless of model switching. In this test, the MRM is configured to switch between its constituent models on every time step. The switching mechanism maintains state and behavioral consistency if the time series produced by the MRM is identical to the time series produced by the aggregate logistic model.

#### 4.3.1.1.3 Resegmentation Test

Segmentation consistency requires that the MRM maintains consistency when a model switch happens between allowable transitions for the passive model (i.e., at a simulation time when the transition function is not defined). For example, if the MRM switches from an IOM population to the aggregate population model, between annual time steps, the state and transition function for the aggregate model are undefined. To compensate for this problem, the user specifies a resegmentation function using the STEMSS MRM structures.

This test uses the predator-prey MRM in which the predator is configured as an IOM and the prey is allowed to switch between the aggregate population model and IOM. The MRM monitors the prey IOM's size between population updates and, when the population rises above a threshold size, the resegmentation function is called and the aggregate prey model is activated. To maintain consistency, when the MRM switches from the prey IOM to the aggregate prey model, the prey IOM tracks the number of offspring produced before the switch, $N_t$, then counts the number of individuals with pending reproduction events, $N$. The pending individuals are treated as an aggregate population, i.e., the number of offspring is computed as $N*R$. The next population size is $N_t + N*R$.

#### 4.3.1.1.4 Cross-Resolution Interaction Test

When a model is composed of two or more models, for example the predator-prey model, and the models are not forced to have the same resolution, it is possible for the models to have different combinations of resolutions. In the predator-prey models, the combinations may be $\text{Pred}_{\text{IOM}}$-$\text{Prey}_{\text{IOM}}$, $\text{Pred}_{\text{agg}}$-$\text{Prey}_{\text{IOM}}$, $\text{Pred}_{\text{IOM}}$-$\text{Prey}_{\text{agg}}$ and $\text{Pred}_{\text{agg}}$-$\text{Prey}_{\text{agg}}$. The predator-prey MRM resolves the problem by either aggregating individual predator actions, when the predator IOM and aggregate prey model are active or determining the aggregated predator's culling behavior then temporally distributing these actions when the aggregate predator model and prey IOM are active.

When the models may have different combinations of resolutions and these combinations can change on every time step, maintaining consistency requires the model to account for several possible types of interactions and resolution changes. To ensure that the MRM maintains consistency, the cross-resolution test examines the following conditions:

1. the predator and prey models switch in phase on every time step,
2. the predator and prey models switch out-of-phase on every step,
3. the predator and prey switch out-of-phase on every other step, so they have the same resolution every other step,
4. the predator and prey switch every fourth step, so they have the same resolution for two consecutive steps,
5. switching occurs at arbitrary switching thresholds so the switching pattern is not predetermined.

The first test forces the models to switch to the same resolution on each time step during the simulation. This tests the ability of STEMSS to coordinate resolution changes. The second, third and forth tests force the models to switch out-of-phase, increasing the phase differences in each test, which validates the ability of STEMSS to coordinate arbitrary sequences of resolution change. The final test validates the system under the expected conditions of use.

### 4.3.1.2 Logistic Model Parameters

Each of the logistic models relies on three parameters, $R$, the average per capita growth rate, $U$, the upper population bound and $N_o$, the initial population size. The value of $R$ determines whether the population declines, grows to a stable size, oscillates about a population size or becomes chaotic. To test the range of behaviors (except chaotic dynamics) the values for $R$ were 2.5, 3.0, 3.5 and 3.7, which produce damped, single cycle, 2-cycle and 4-cycle oscillations respectively. The upper population bound was set to 10, 100, 1000 and the initial population size was set to half the upper population bound.

### 4.3.1.3 Predator-Prey Model Parameter

The predator-prey model relies on six parameters ($U, R, A, B, h, k$). $U$ and $R$ are the upper population bound and average per capita growth rate for the prey species. The parameters $A$ and $B$ determine the impact of predator species on the prey population. The parameters $h$ and $k$ are the density-dependent upper population bound and average per capita growth rate of the predator species. This test uses two parameters sets: the set (1000, 3.5, 1, 2, 0.7, 2.7) causes both populations to reach a stable population size and the set (3000, 3.5, 1, 2, 0.8, 2.0) causes both populations to oscillate.

### 4.3.1.4 Data Collection and Analysis Procedures

For this set of tests, the analysis consists of two procedures. The first is to compare models by superimposing their time series and visually checking for changes. This procedure is most useful when the difference between the time series is visually obvious. The second procedure is to compute the sum of the squared differences (*SS*) between the models. That is, for an MRM composed of a low-resolution model, LRM, and a high resolution model, HRM $SS = \Sigma(N_{xrm,i} - N_{yrm,i})^2$, where $xrm \in \{\text{LRM, HRM, MRM}\}$, $yrm \in \{\text{HRM, MRM}\}$ and $xrm \neq yrm$, and i ranges across elements of the time series. When the time series tested by this procedure are the same, the value is zero. If the differences between the models are visually obvious, then the value will be large.

To test the effect of perturbation on the models, establishing that the aggregate model and IOM are identical is necessary. Once the equivalence of the models is established, it is necessary to establish that the models are sensitive to perturbation. The effect of perturbation is validated by adding a single individual to the population (for the predator-prey model, the individual is added to the prey population) and visually comparing the perturbed population with the unperturbed population.

After establishing the sensitivity of the logistic and predator-prey models, both models can be used to test the specific MRM mechanisms. Each test isolates a specific mechanism (see sections 4.3.1.1.2 -4.3.1.1.4) and a 500-year simulation was run for each combination of model parameters. The data was analyzed as follows:

- each time series was grouped according to parameter set and mechanism test type,
- for each pair of models (i.e., LRM-HRM, LRM-MRM, HRM-MRM) the sum of the squared difference for each step of the time series was computed,
- the results of the comparisons and each mechanism were then tabulated.

### 4.3.2 Results

Simulations using the aggregate model and IOM for both the logistic and predator-prey models were run and compared. For each parameter set, the sum of the squared differences was zero. That is, aggregate model and IOM produce identical results, so they are suitable for the perturbation and mechanism tests.

Figure 30 shows the effect of perturbation on the logistic models and Figures 31 and 32 show the effect of perturbation on the predator and prey models. For the logistic model, the effect is visually obvious by year 20, approximately 10 simulated years after the perturbation. Direct analysis of the simulation data shows that the perturbation has an immediate numerical impact on the model. The impact of perturbation on the predator-prey model is visually obvious immediately on the prey population after the prey model was perturbed and the impact on the predator population lagged by a year, but the effect was obvious. Therefore, both models are sensitive to perturbation.

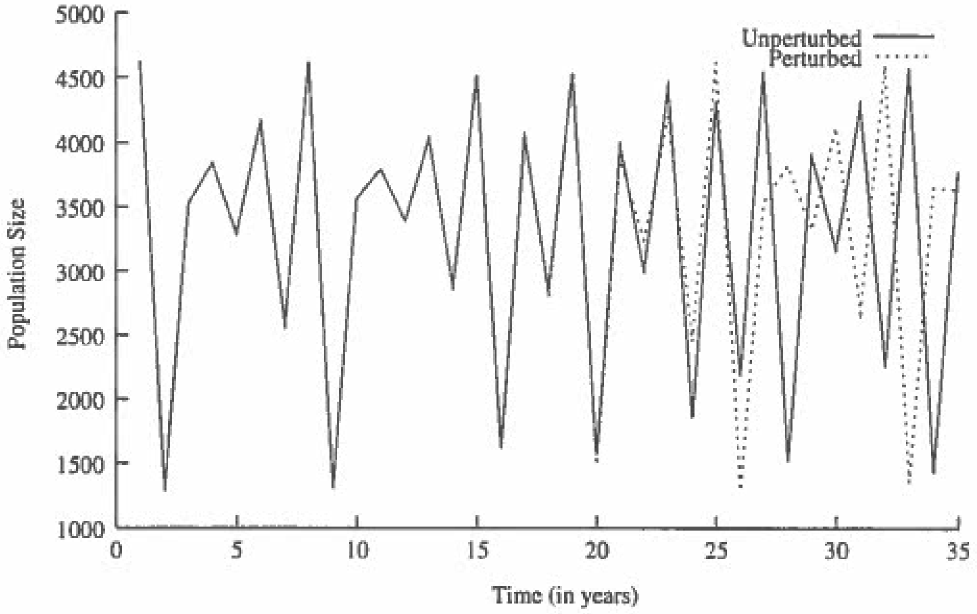

Figure 30. Unperturbed and Perturbed Logistic Population

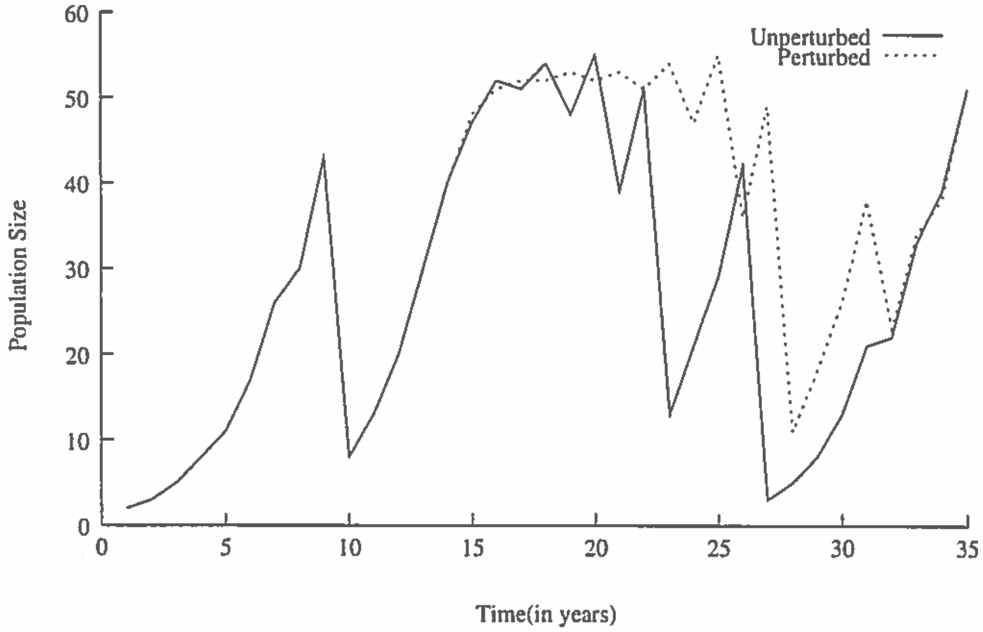

Figure 31. Unperturbed and Perturbed Predator Population

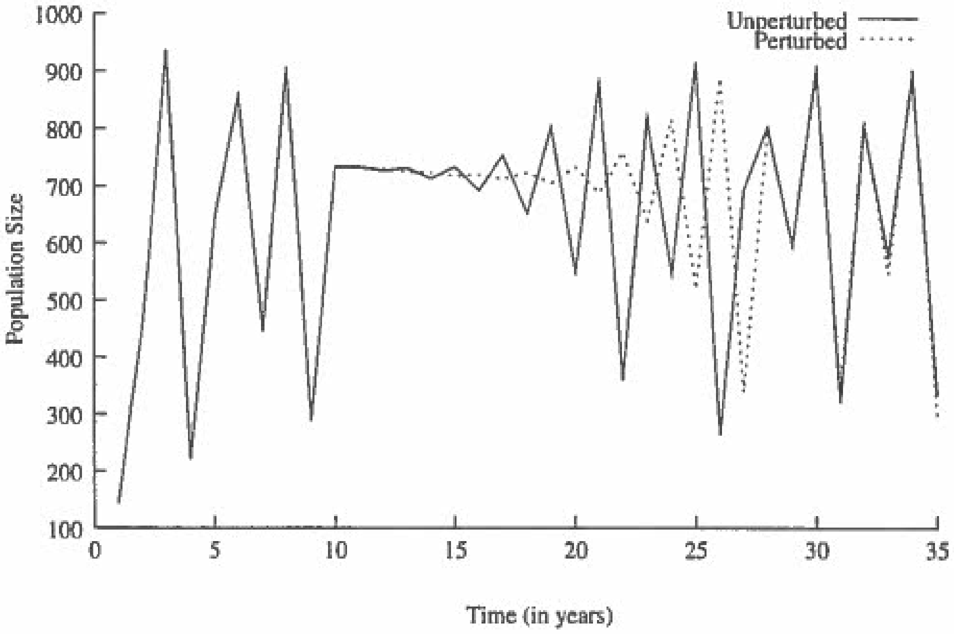

Figure 32. Unperturbed and Perturbed Prey Population

Table 2. Switching Test Results

| <U,R> | (HRM - LRM)$^2$ | (HRM - MRM)$^2$ | (LRM - MRM)$^2$ |
|---|---|---|---|
| <100, 2.5> | 0 | 0 | 0 |
| <1000, 2.5> | 0 | 0 | 0 |
| <10,3.0> | 0 | 0 | 0 |
| <100,3.0> | 0 | 0 | 0 |
| <1000,3.0> | 0 | 0 | 0 |
| <10,3.5> | 0 | 0 | 0 |
| <100,3.5> | 0 | 0 | 0 |
| <1000,3.5> | 0 | 0 | 0 |
| <10,3.7> | 0 | 0 | 0 |
| <100,3.7> | 0 | 0 | 0 |
| <1000,3.7> | 0 | 0 | 0 |

Table 2 presents the results for the switching test. The table shows the pairwise sums of the squared differences for the aggregate model, IOM and MRM. In each instance the tests confirm that the time series produced by the MRM is identical to the time series produced by the aggregate logistic model (LRM) and the logistic IOM (HRM). These tests validate the switching mechanism for the logistic MRM. The time series produced by each

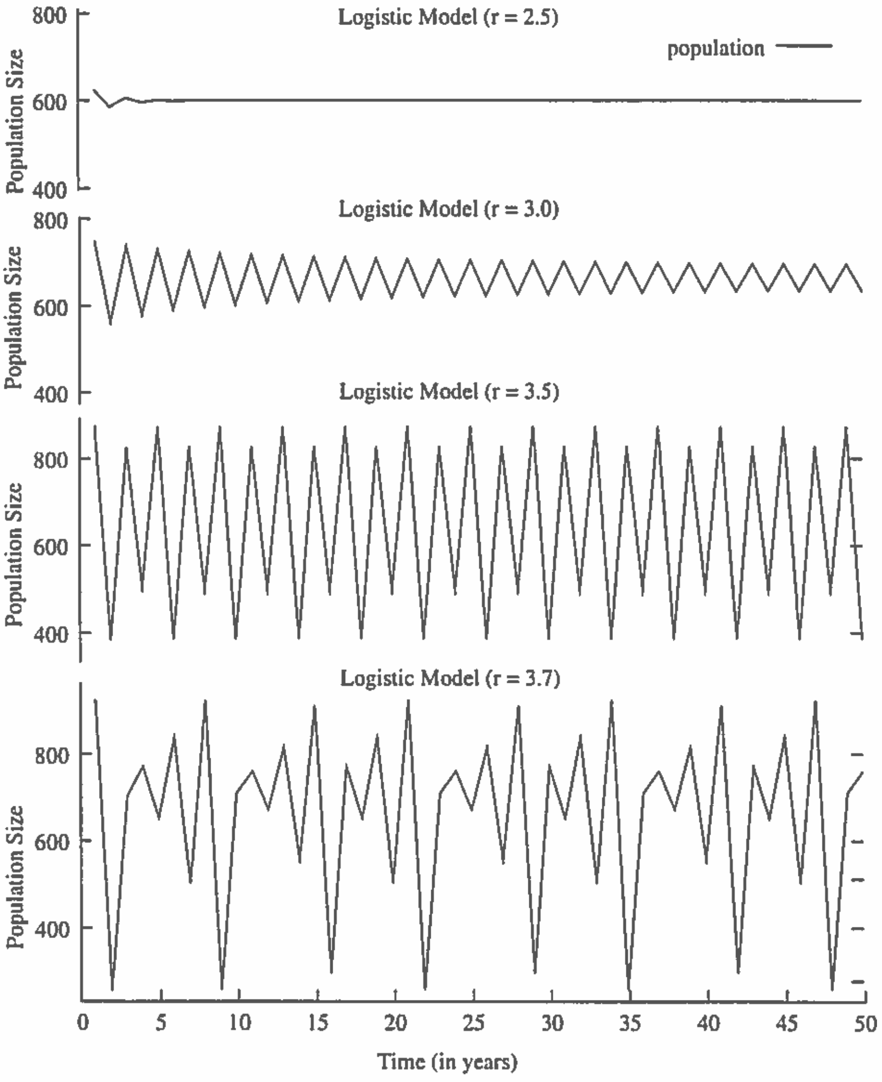

Figure 33. Logistic Population Model Time Series

parameter set tested are presented in Figure 33 to provide a sense of the complexity of the time series produce by the logistic model.

Table 3 presents the results for the resegmentation and cross-resolution tests. The sum of the squared differences confirm that aggregate predator-prey model (LRM), the predator-prey IOM (HRM) and the predator-prey MRM produce identical time series. These tests validate that the resegmentation and cross-resolution mechanisms for the predator-prey MRM. The time series produced the <1000, 3.5, 2, 0.8> parameter set is shown in Figures 34 and the series produced by the parameter set <3000, 3.5, 2.7, 0.7> is show in Figure 35. These figures provide a sense of the complexity of the predator-prey model.

Table 3. Resegmentation and Cross-Resolution Interaction Results

| Type | <U, R, k, h> | $(HRM - LRM)^2$ | $(HRM - MRM)^2$ | $(LRM - MRM)^2$ |
|---|---|---|---|---|
| CRI | <3000, 3.5, 2.7, 0.7> | 0 | 0 | 0 |
| CRI | <1000, 3.5, 2.0, 0.8> | 0 | 0 | 0 |
| Reseg | <3000, 3.5, 2.7, 0.7> | 0 | 0 | 0 |
| Reseg | <1000, 3.5, 2, 0.8> | 0 | 0 | 0 |

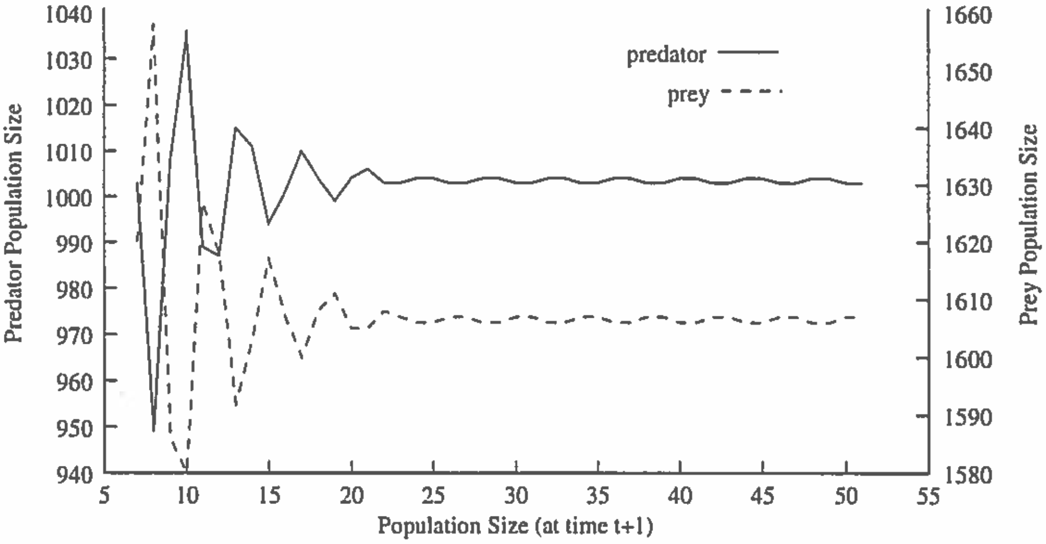

Figure 34. Stable Predator and Prey Populations

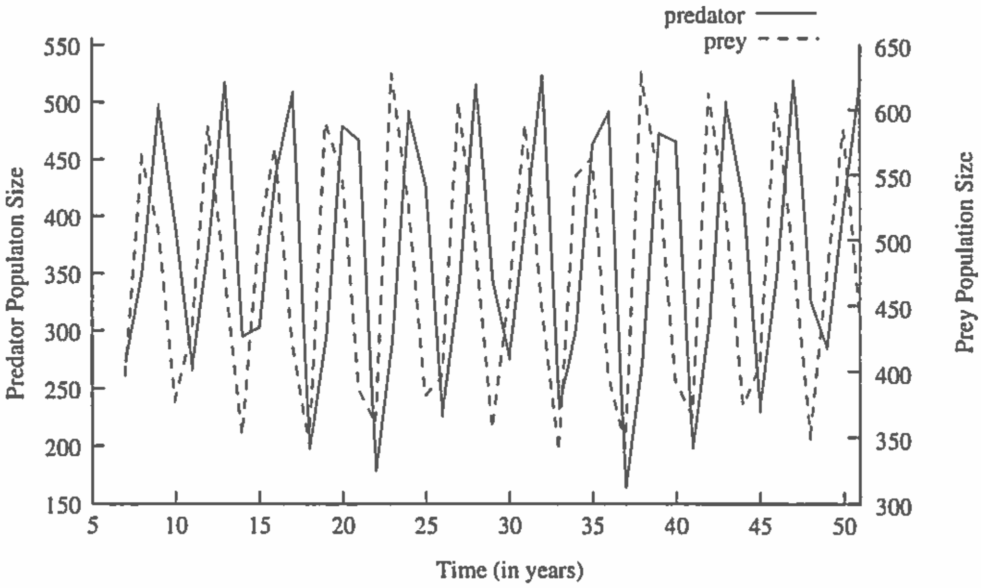

Figure 35. Cyclical Predator and Prey Populations

### 4.3.3 Discussion

The tests confirm that the aggregate and IOM versions of both the logistic and predator-prey models are identical and are sensitive to perturbation. These models were used to confirm that the mechanisms required by MRM and implemented by STEMSS-MRM do not induce perturbations in the model's state. Each of the mechanism tests validated the MRM mechanisms, supported by STEMSS-MRM and implemented by the user, maintained consistency.

The importance of consistency maintenance by these models is twofold. First, they are examples of valid MRMs. The MRMs are bounded by the aggregate models (LRM) and the IOMs (HRM). Second, the logistic model, after some modification, will be used as the source model for the stochastic MRM tests.

## 4.4 Yellow Perch Model Suite

The second validation test uses the yellow perch suite to demonstrate the ability to construct complex ecological models using STEMSS and the ability to construct an MRM using these models. The model, using the parameter set suggested by Rose (1995), requires 2000-2400 computer hours to generate a single 10,000-year time series. To avoid this cost, the models used in this test are deterministic and produce analyzable results in less than 24 hours. The modifications required to make the model deterministic alter the results of the stochastic model, so the two models are not comparable. Nevertheless, the altered model retains the same set of transition functions and state value as the original model, so the test does validate STEMSS' ability to produce valid MRMs.

## 4.4.1 Method

### 4.4.1.1 Simulation Model

The yellow perch IOM (high-resolution model) and super-individual model (low-resolution model) are described in chapter III. For both of these models, the stochastic terms were replaced by constants to remove the confounding effects of stochasticity. The yellow perch MRM, represents each cohort by an MRM composed of the IOM and super-individual cohort models. Experience with the IOM version indicates that individuals in the young cohort dominate the computation time, so it is the only switching model in this test. The remaining cohort models are represented by MRMs, but the switching mechanism in these models is inhibited, so they use the IOM exclusively.

### 4.4.1.2 Model Parameters

The parameters used for the model were adapted from Rose et al. (1992) and the details regarding the parameters were discussed in section 1.5. The values listed in Table 4 are the parameters associated with individual consumption and growth (Equations 4 and 5); Table 5 lists the parameters associated with reproduction; Table 6 lists the set of predation parameters (see Equation 6); Table 7 lists the set of thermal modifiers for consumption and growth; Table 8 lists the set of initial conditions for each individual.

Table 4. Consumption and Metabolic Parameters

| Stage | AC | BC | AL | BL | AR | BR | ACT | KE | Mortality |
|---|---|---|---|---|---|---|---|---|---|
| FF | 0.5 | -0.4 | 45.9 | 0.33 | 0.0005 | -0.20 | 4.4 | 0.43 | 0.9 |
| YoY | 0.06 | -0.3 | 45.9 | 0.33 | 0.002 | -0.20 | 1.0 | 0.4 | 0.9 |
| Yearling | 0.13 | -0.2 | 39.1 | 0.33 | 0.002 | -0.20 | 1.0 | 0.43 | 0.9 |
| Adult | 0.01 | -0.1 | 39.1 | 0.33 | 0.04 | -0.25 | 1.0 | 0.5 | 0.5 |

Table 5. Fecundity Parameters

| | AF | BF | Sex Ratio(M:F) |
|---|---|---|---|
| Adult | 180.0 | 3658.0 | 0.5 |

Table 6. Predation Parameters

| Stage | KZ | KB | KY |
|---|---|---|---|
| FF | 0.005 | 0 | 0 |
| YoY | 4. | 2500. | 0 |
| Yearling | 4. | 1000. | 0 |
| Adult | 5000.0 | 5000. | 0.00003 |

Table 7. Thermal Metabolic and Consumption Parameters

| Stage | Tco | Tcm | ThetaC | Tro | Trm | ThetaR |
|---|---|---|---|---|---|---|
| Young | 29 | 32 | 2.3 | 32 | 35 | 2.1 |
| Adult | 23 | 28 | 2.3 | 28 | 133 | 2.1 |

Table 8. Initial Size

| AC | young | yearling | adult2 | adult3 | adult4 | adult5 | adult6 | adult7 | adult8 | adult9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Len | 50.6 | 120.6 | 155.6 | 167.8 | 182.3 | 197.9 | 214.4 | 230. | 248.8 | 268.6 |
| Size | 20005 | 5000 | 200 | 100 | 520 | 24 | 12 | 8 | 4 | 1 |

#### 4.4.1.3 Data Collection and Analysis Procedures

The yellow perch simulations used the parameter set and initial conditions found in Tables 4-8. Simulations using the yellow perch super-individual model (LRM) and yellow perch IOM (HRM) eventually reach a fixed point, so, rather than setting an end time for the simulation, all simulations were terminated when the population stabilized. The simulation output was collected and the time series superimposed on a single graph. The graphs were visually compared and the stable population sizes were quantitatively compared to establish boundedness.

### 4.4.2 Results

Figure 39 shows results of the population simulation. The steady state population size was reached by day nine thousand, and the population size produced by the MRMs, is bounded by the sizes produced by IOM and aggregate model. The assumption that the super-individual model (LRM) and IOM (HRM) represent the same system is, however, false. There is a 124% discrepancy between the steady state population size (i.e., the population size on day nine thousand) of the aggregate model and the IOM, so the two models are different.

The time series produced by each model are visually similar for the first three years then they begin to diverge. The appearance of the divergence, in Figure 36, coincides

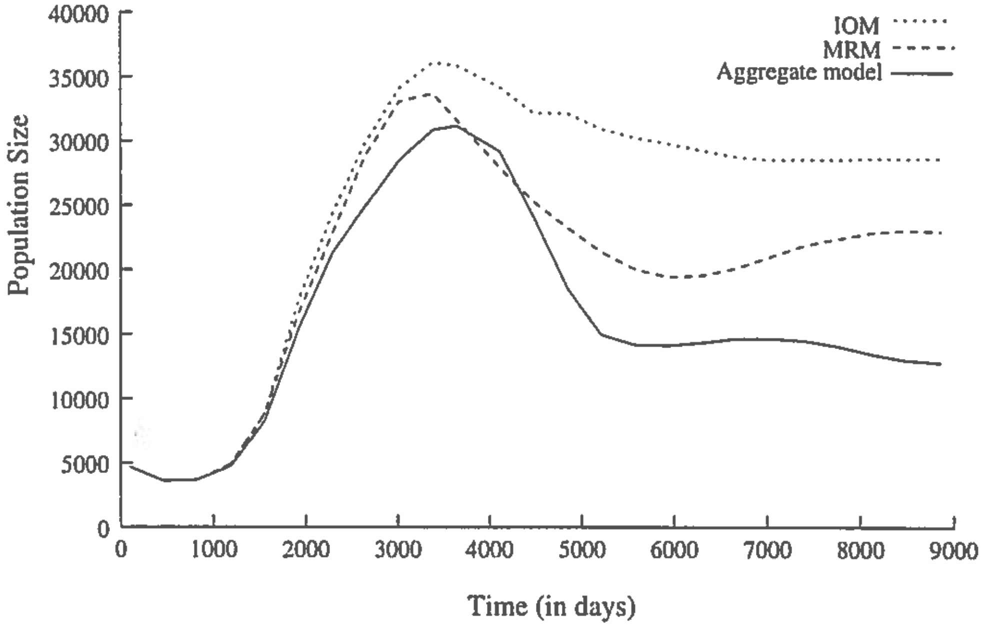

Figure 36. Yellow Perch MRM Test

with the maturation of the first cohort produced by the simulated adults, suggesting that discrepancy arises from the individuals generated by the simulation, as opposed to those generated at initialization. To confirm this suggestion, a separate model was run to track the growth of individuals from birth to one year, which showed that the first-feeder and young-of-the-year stages produced minor differences in growth for each day in the stage. These differences were accumulated, causing the growth of the super-individual to lag behind the growth of the IOM individuals by approximately 10 simulated days. As adults have to reach a critical size before they can cull young and reproduce, the ten-day lag creates a significant change in the model dynamics.

### 4.4.3 Discussion

The results of these simulations are consistent with the hypothesis that the MRM is bounded by the IOM and aggregate models. The stable population size confirms that the MRM is bounded by the super-individual model and the IOM. Despite the discrepancies in individual growth and reproduction between the constituent models, the MRM performed as expected. The steady state population sizes produced by the super-individual model and the IOM are qualitatively similar. That is, all three models reach a stable population size that suggests that the underlying dynamics for each model are similar.

The discrepancy between the models adds another concern beyond questions related to MRM. This test demonstrates that subtle numerical differences between models can affect the time series produced by the models. In this example, numerical deviations led to delays in discrete state changes (e.g., the ability to cull young) and the delay induced changes in per capita growth rate, fecundity and egg laying.

The error in growth rate is associated with determinism in the model. Modeled individuals grow in lockstep, so discontinuities in model behavior, such as culling ability or reproduction, are simultaneously encountered by all individuals. In a stochastic system, changes in behavior are not, usually, precisely synchronized. That is, individuals are not generally identical, so individual size is not expected to be constant in a cohort. As there is a distribution of sizes within a cohort, size-dependent changes will happen over a range of times, which reduces the effect of state discontinuities. This effect is similar to antialiasing (Foley, Van Dam, Feiner and Hughes, 1990).

## 4.5 Stochastic Logistic Experiment

The third set of validation tests was designed to quantify the effect of model switching on stochastic MRMs by comparing the time series produced by the LRM, HRM and MRM. The validated logistic suite, described in section 4.2, was used for this experiment because the time series produced by these models are easy to understand. The logistic IOM was modified to produce a stochastic time series, consequently, the MRM is stochastic. To control which model is active, the MRM compares the population size with a switching threshold. When the size falls below the switching threshold, the IOM is activated when it rises above the switching threshold, the aggregate model is activated. The switching threshold determines the amount of IOM activity in a simulation and, therefore, it determines the accuracy of the MRM.

The results of these tests will be used to estimate the accuracy of an MRM as a function of switching threshold and model parameters. The relationship between the switching threshold and model accuracy suggests that user can select a level of accuracy by selecting a threshold value. If the results validate this suggestion, then switching threshold gives the modeler some control over the accuracy of the model. The next chapter will explore the use of the switching threshold for controlling the execution time of the simulation. Assuming both sets of tests validate the ability of the switching threshold to control accuracy and execution time, then a user can estimate of the accuracy and execution time of the simulation based on the switching threshold.

## 4.5.1 Models and Methods

### 4.5.1.1 Simulation Model

The deterministic models in the logistic suite are characterized by two parameters, $r$ and $U$. The stochastic model includes a uniform random number, $\Delta$, to the individual birth rate, $R$. The remaining parameters form specific test groups that will be described as $<U, \Delta>$. The MRM switches between the aggregate logistic model (LRM) and the logistic IOM (HRM) based on a switching threshold that is a fraction of the upper population bound. That is, the threshold parameter, $th$, is a value between 0 and 1.

### 4.5.1.2 Parameters

For all tests in this experiment, the average growth rate per individual is, $R = 3.0$. This value was chosen because it produces interesting dynamics while avoiding chaos. The upper population bound, $U$, was selected from the set {100, 1000, 2500, 5000} and the initial population was set to $U/2$. To test the effect of stochasticity on MRM performance, a uniform random term in the range, +/- $\Delta$, was added to $r$, which, in this model, is interpreted as the individual birth rate. Experience indicated that $\Delta = 1.0$ and 2.5 produced sufficiently random simulations for the purposes of this experiment. To test the effect of the switching threshold on model performance and to determine the relationships between *th* and accuracy, simulations for each set of model parameters were run for switching thresholds, $th \in \{0.23, 0.235, 0.24\}$.

### 4.5.1.3 Data to be Collected

The test determines whether the MRM is bounded by the aggregate logistic model (LRM) and the stochastic logistic IOM (HRM). The specific meaning of bounded is not defined, however. For the logistic model suite, logistic MRM is bounded by the aggregate logistic and the logistic IOM when it meets the following criteria:

- $N_{MRM,max} < N_{IOM,max}$, where $N_{MRM,max}$ is the maximum population size from the time series generated by the MRM and $N_{IOM,max}$ is the maximum population size from the time series generated by the IOM
- $N_{agg,max} < N_{MRM,max}$, where $N_{agg,max}$, is the maximum population size from the time series generated by the aggregate logistic model,
- $N_{MRM,min} > N_{agg,min}$, where $N_{agg,min}$, is the minimum population size from the time series generated by the aggregate logistic model,
- $N_{IOM,min} > N_{MRM,min}$, where $N_{IOM,min}$, is the minimum population size from the time series generated by the IOM.

When a model satisfies this boundedness criteria, it is "amplitude bounded."

One shortcoming of this definition is that it ignores frequency. Consider a low-resolution model and a high-resolution model that produce a square-wave time series. If an MRM, composed of these models, produces a triangular-wave time series, then the models are not bounded, even if the two series are amplitude-bounded. Therefore, declaring the models bounded would be inaccurate. Nevertheless, amplitude-boundedness does provide a measurable quantity for analyzing boundedness and, if the data do not exhibit poor frequency behavior, it should stand as a reasonable measure of boundedness.

#### 4.5.1.4 Statistical Tests

The goal of this experiment is to show that the upper bound of the MRM is bounded by the upper bounds of aggregate logistic (LRM) and logistic IOM (HRM) and that the lower bound of the MRM is bounded by the lower bounds of the aggregate logistic model and logistic IOM. Let $N_{max,IOM}$ and $N_{min,IOM}$ be the maximum and minimum IOM population sizes, $N_{max,agg}$ and $N_{min,agg}$ be the maximum and minimum aggregate model population sizes and $N_{max,MRM}$ and $N_{min,MRM}$ be the maximum and minimum MRM population sizes. If, for the version of the logistic model used in this experiment, $N_{max,agg} < \mathrm{N}_{\mathrm{max,MRM}} < N_{max,IOM}$, then the upper bound of the MRM is bounded by the upper bounds of the aggregate logistic model and logistic IOM. If $N_{min,agg} > N_{min,MRM} > N_{min,IOM}$, given that one set of tests is used to compare the upper bound and the other set is used to compare the lower bound, the two sets can be combined into a single set of relations: $N_{min,IOM} < N_{min,MRM_{0.24}} < N_{min,MRM_{0.235}} < N_{min,MRM_{0.23}} < N_{min,agg} < N_{max,agg} < N_{max,MRM_{0.23}} < N_{max,MRM_{0.235}} < N_{max,MRM_{0.24}} < N_{max,IOM}$.

where $MRM_{0.23}$, $MRM_{0.235}$ and $MRM_{0.24}$ refer to the MRMs using switching thresholds of 0.23, 0.235 and 0.24 respectively.

A common statistical test designed to establish the relationships above is the Fisher least significant difference (LSD) test (Freund & Wilson, 1997). The LSD test is a two-step process. In the first step, an ANOVA is done to test the equality of a set of population means (Freund & Wilson, 1997). In the second step, the LSD value is computed and, for each pairwise comparison, the difference is significant if the difference between the pair is greater than the LSD value. The ANOVA and LSD tests were computed using SPSS 11.5 (SPSS, 2003).

By definition, pseudo-random number (PRN) generators do not produce random number sequences, so the sequence of numbers produced by the PRN are autocorrelated. The STEMSS PRN generator uses the algorithm found in (Knuth, 1998), which attempts to reduce the effect of autocorrelation. Nevertheless, the PRN sequence is autocorrelated and, therefore, it influences the validity of the statistical tests.

Since the aggregate logistic model is deterministic, it does not produce normally distributed data and its variance is zero. Consequently, it does not meet the assumptions of the LSD procedure. Therefore, it will not be included as part of the statistical tests. Instead, the maximum and minimum population sizes of the aggregate logistic model will be compared to the mean maximum and mean minimum population sizes from each MRM. The aggregate value will be included for reference.

### 4.5.1.5 Data Collection and Analysis

Each combination of parameters and switching threshold values was used to produce 700 hundred year simulations and each simulation was repeated 100 time with different random number seeds each time. After collecting the simulation results, the following tasks were performed:

- the first 200 years of data were removed from each simulation to eliminate the initial simulation transient,
- data were grouped according to parameter type (e.g., all time series generated using <100, 1> were grouped together),
- the maximum and minimum value of each time series were collected for each simulation and grouped according to model type (i.e., for each $<U, \Delta>$).

For each grouping, the SPSS one-way ANOVA function with the LSD option was used to analyze the data.

## 4.5.2 Results

The average minimum population sizes, for each experiment, are presented in Table 9 and the average maximum population sizes are presented in Table 10 and are graphically presented in Figures 37-40. In general, the results satisfy the constraint,

$N_{min,IOM} < N_{min,MRM_{0.24}} < N_{min,MRM_{0.235}} < N_{min,MRM_{0.23}} < N_{min,agg} < N_{max,agg} <$

$N_{max,MRM_{0.23}} < N_{max,MRM_{0.235}} < N_{max,MRM_{0.24}} < N_{max,IOM}$, although there are some exceptions. Specifically, some simulations, using the parameter set <100,2.5> produced population extinctions that skewed the results. Also, the $MRM_{0.24}$ with $\Delta = 1.0$, produced a time series that was similar to the IOM time series.

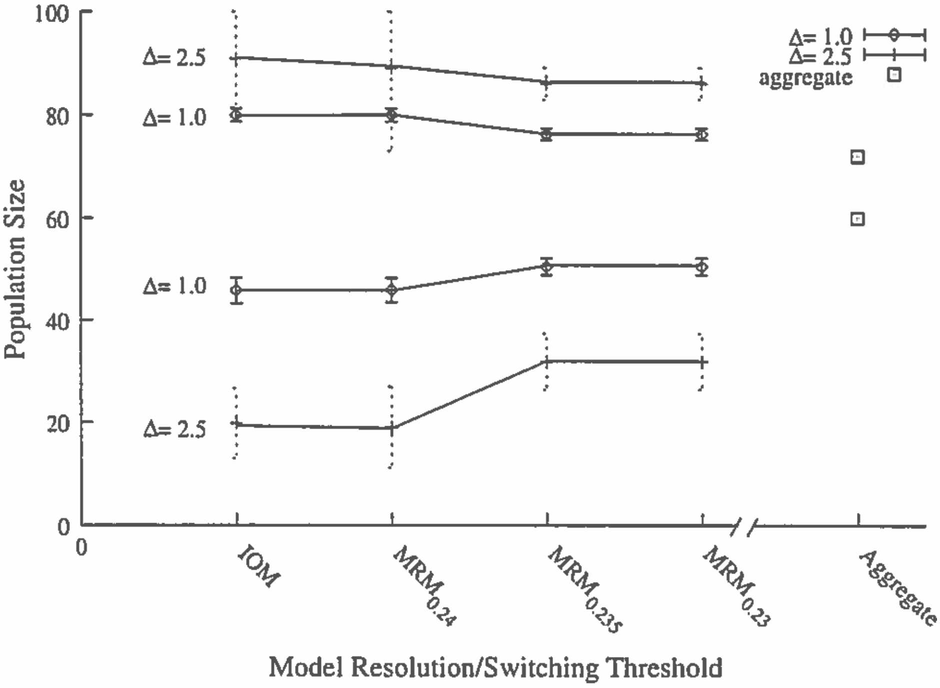

Figure 37. Average Maximum and Minimum Population Sizes, U = 100

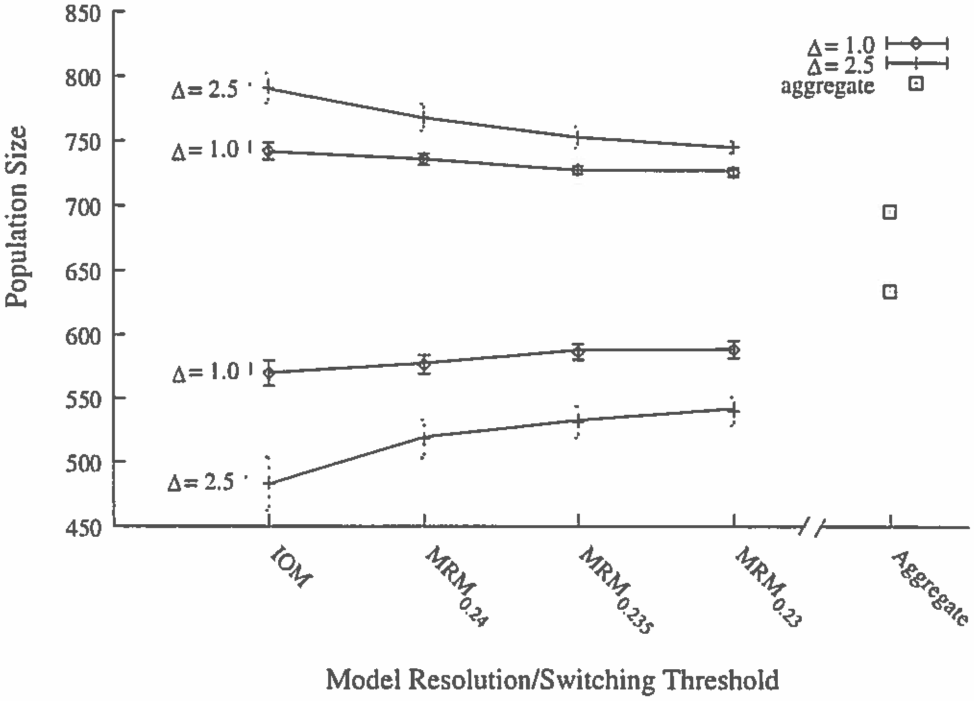

Figure 38. Average Maximum and Minimum Population Sizes, U = 1000

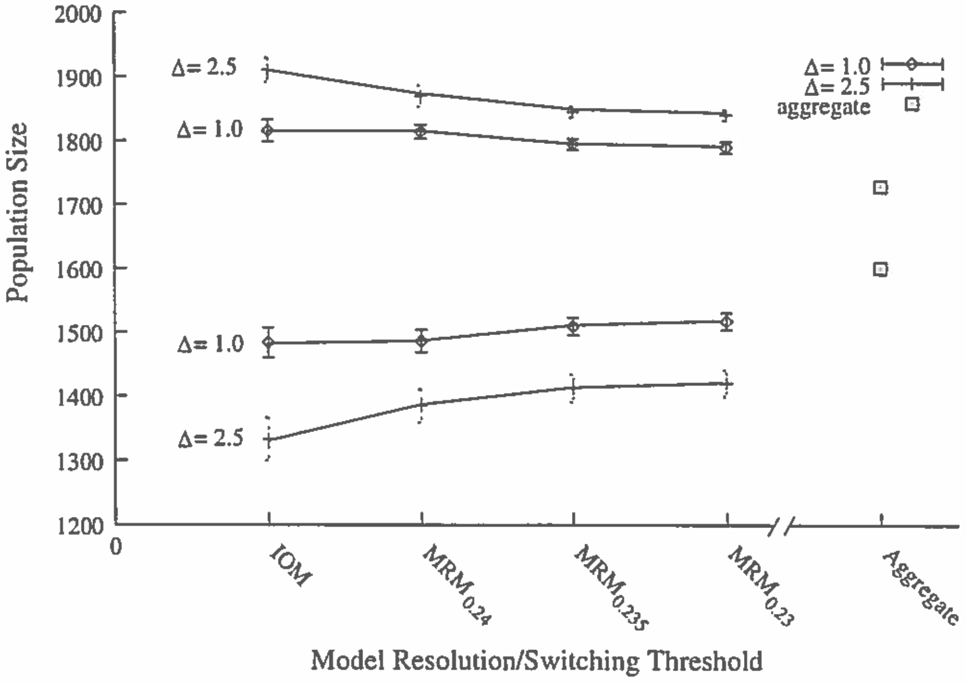

Figure 39. Average Maximum and Minimum Population Sizes, U = 2500

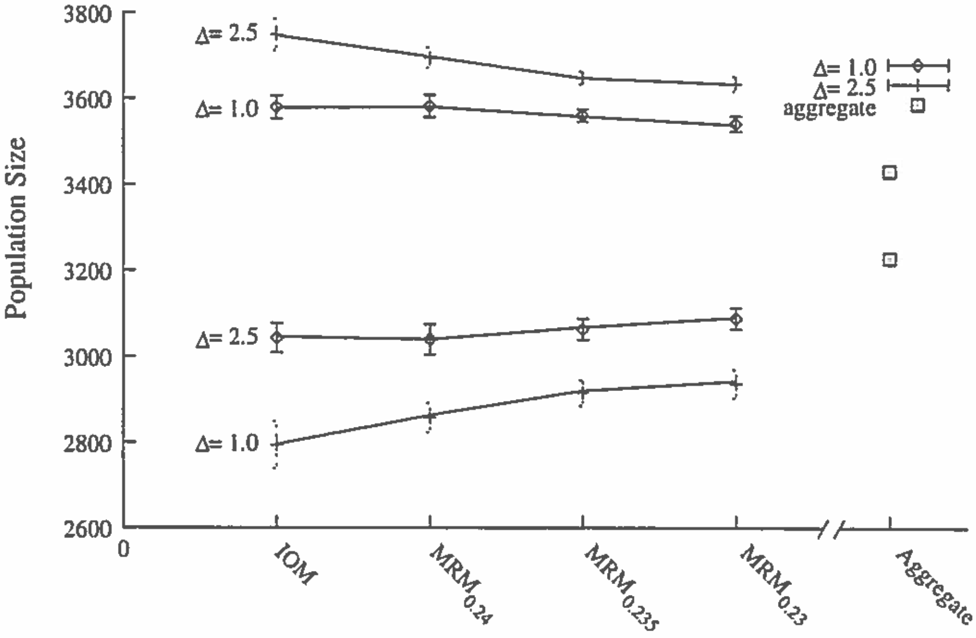

Figure 40. Average Maximum and Minimum Population Sizes, U = 5000

The ANOVA and LSD results presented in Tables 11-26 support the claim that, in general, the logistic MRM is bounded by the aggregate logistic model and the logistic IOM. The tests also reinforce the claim that the $MRM_{0.24}$ is the same as the IOM, when $\Delta = 1.0$, though the tests are not conclusive because of variability in the standard deviation of each model type (see Tables 11-26). As the upper population bound increases from $U = 1000$ to $U = 5000$, the probability that models are equivalent increases, that is, the time series produced by the $MRM_{0.24}$ appears to converge to the time series produced by the IOM. There is, however, insufficient evidence to state this claim with statistical confidence.

The ANOVA and LSD tests also indicate that the $MRM_{0.235}$ and $MRM_{0.23}$ diverge as *U* increases. Figures 41 and 42 show the impact of the switching threshold with increasing *U* for a given Δ. Each point on the x-axis shows the maximum and minimum population sizes (as ratios of upper population bound, *U*). For $\Delta = 1$ (Figure 41), the difference between $MRM_{0.235}$ and $MRM_{0.23}$, as a ratio of *U*, declines as *U* increases, which is unexpected since the models diverge as *U* increases. Since the variation in maximum and minimum values, for all models, declines as *U* increases, concluding that the divergence is more a loss of variation is reasonable than it is an actual divergence. Moreover, the data suggest that all of the models are converging to the aggregate model as *U* increases.

Table 9. Average Minimum Population Size.

| | N | Minimum | Maximum | Mean | Std. Dev. |
|---|---|---|---|---|---|
| IOM<100,1> | 100 | 37 | 50 | 45.81 | 2.54 |
| MRM$_{0.24}$<100,1> | 100 | 39 | 51 | 45.89 | 2.36 |
| MRM$_{0.235}$<100,1> | 100 | 45 | 53 | 50.47 | 1.65 |
| MRM$_{0.23}$<100,1> | 100 | 45 | 53 | 50.47 | 1.65 |
| AGG<100> | | | | 60 | |
| IOM<1000,1> | 100 | 545 | 598 | 569.7 | 9.98 |
| MRM$_{0.24}$<1000,1> | 100 | 554 | 596 | 576.47 | 7.32 |
| MRM$_{0.235}$<1000,1> | 100 | 570 | 597 | 586.36 | 6.12 |
| MRM$_{0.23}$<1000,1> | 100 | 572 | 605 | 588.49 | 6.83 |
| AGG<1000> | | | | 634 | |
| IOM<2500,1> | 100 | 1407 | 1543 | 1483.31 | 23.84 |
| MRM$_{0.24}$<2500,1> | 100 | 1445 | 1542 | 1485.75 | 17.90 |
| MRM$_{0.235}$<2500,1> | 100 | 1473 | 1558 | 1508.76 | 13.60 |
| MRM$_{0.23}$<2500,1> | 100 | 1479 | 1554 | 1516.57 | 13.45 |
| AGG<2500> | | | | 1600 | |
| IOM<5000,1> | 100 | 2963 | 3117 | 3042.80 | 34.35 |
| MRM$_{0.24}$<5000,1> | 100 | 2967 | 3125 | 3039.24 | 34.94 |
| MRM$_{0.235}$<5000,1> | 100 | 3011 | 3155 | 3063.20 | 24.46 |
| MRM$_{0.23}$<5000,1> | 100 | 3035 | 3141 | 3088.03 | 24.86 |
| AGG<5000> | | | | 3228 | |
| IOM<100,2.5> | 100 | 0 | 33 | 19.76 | 6.79 |
| MRM$_{0.24}$<100,2.5> | 100 | 0 | 32 | 19.01 | 7.89 |
| MRM$_{0.235}$<100,2.5> | 100 | 16 | 44 | 31.79 | 5.50 |
| MRM$_{0.23}$<100,2.5> | 100 | 16 | 44 | 31.79 | 5.50 |
| AGG<100> | | | | 60 | |
| IOM<1000,2.5> | 100 | 428 | 523 | 483.08 | 20.75 |
| MRM$_{0.24}$<1000,2.5> | 100 | 469 | 542 | 518.01 | 14.91 |
| MRM$_{0.235}$<1000,2.5> | 100 | 478 | 552 | 531.22 | 12.25 |
| MRM$_{0.23}$<1000,2.5> | 100 | 512 | 561 | 539.72 | 11.33 |
| AGG<1000> | | | | 634 | |
| IOM<2500,2.5> | 100 | 1246 | 1404 | 1333.20 | 33.67 |
| MRM$_{0.24}$<2500,2.5> | 100 | 1252 | 1430 | 1384.45 | 25.55 |
| MRM$_{0.235}$<2500,2.5> | 100 | 1342 | 1448 | 1411.55 | 21.23 |
| MRM$_{0.23}$<2500,2.5> | 100 | 1368 | 1461 | 1419.75 | 20.47 |
| AGG<2500> | | | | 1600 | |
| IOM<5000,2.5> | 100 | 2640 | 2911 | 2793.66 | 54.66 |
| MRM$_{0.24}$<5000,2.5> | 100 | 2762 | 2935 | 2857.31 | 33.53 |
| MRM$_{0.235}$<5000,2.5> | 100 | 2828 | 2978 | 2914.88 | 30.14 |
| MRM$_{0.23}$<5000,2.5> | 100 | 2847 | 3035 | 2936.02 | 33.23 |
| AGG<5000> | | | | 3228 | |

Table 10. Average Maximum Population Size

| | N | Minimum | Maximum | Mean | Std. Dev |
|---|---|---|---|---|---|
| IOM<100,1> | 100 | 77 | 83 | 79.94 | 1.29 |
| $\text{MRM}_{0.24}$<100,1> | 100 | 78 | 85 | 79.95 | 1.31 |
| $\text{MRM}_{0.235}$<100,1> | 100 | 74 | 79 | 76.23 | 1.14 |
| $\text{MRM}_{0.23}$<100,1> | 100 | 64 | 79 | 76.23 | 1.14 |
| AGG<100> | | | | 72 | |
| IOM<1000,1> | 100 | 728 | 760 | 742.02 | 6.46 |
| $\text{MRM}_{0.24}$<1000,1> | 100 | 728 | 751 | 735.77 | 4.35 |
| $\text{MRM}_{0.235}$<1000,1> | 100 | 721 | 735 | 727.51 | 3.16 |
| $\text{MRM}_{0.23}$<1000,1> | 100 | 716 | 734 | 725.92 | 3.59 |
| AGG<1000> | | | | 696 | |
| IOM<2500,1> | 100 | 1768 | 1863 | 1815.67 | 6.86 |
| $\text{MRM}_{0.24}$<2500,1> | 100 | 1773 | 1836 | 1813.78 | 10.95 |
| $\text{MRM}_{0.235}$<2500,1> | 100 | 1759 | 1815 | 1794.19 | 8.49 |
| $\text{MRM}_{0.23}$<2500,1> | 100 | 1764 | 1812 | 1788.97 | 8.59 |
| AGG<2500> | | | | 1728 | |
| IOM<5000,1> | 100 | 3526 | 3652 | 3579.73 | 26.43 |
| $\text{MRM}_{0.24}$<5000,1> | 100 | 3526 | 3642 | 3582.34 | 25.76 |
| $\text{MRM}_{0.235}$<5000,1> | 100 | 3508 | 3593 | 3560.96 | 14.66 |
| $\text{MRM}_{0.23}$<5000,1> | 100 | 3503 | 3578 | 3541.68 | 17.57 |
| AGG<5000> | | | | 3432 | |
| IOM<100,2.5> | 100 | 85 | 100 | 90.88 | 9.55 |
| $\text{MRM}_{0.24}$<100,2.5> | 100 | 96 | 100 | 89.00 | 16.00 |
| $\text{MRM}_{0.235}$<100,2.5> | 100 | 78 | 93 | 86.00 | 3.065 |
| $\text{MRM}_{0.23}$<100,2.5> | 100 | 78 | 93 | 86.000 | 3.065 |
| AGG<100> | | | | 72 | |
| IOM<1000,2.5> | 100 | 769 | 829 | 790.66 | 11.70 |
| $\text{MRM}_{0.24}$<1000,2.5> | 100 | 748 | 791 | 768.090 | 10.21 |
| $\text{MRM}_{0.235}$<1000,2.5> | 100 | 741 | 787 | 752.92 | 8.24 |
| $\text{MRM}_{0.23}$<1000,2.5> | 100 | 738 | 767 | 745.380 | 4.60 |
| AGG<1000> | | | | 696 | |
| IOM<2500,2.5> | 100 | 1866 | 1948 | 1910.48 | 18.81 |
| $\text{MRM}_{0.24}$<2500,2.5> | 100 | 1842 | 1929 | 1869.43 | 16.80 |
| $\text{MRM}_{0.235}$<2500,2.5> | 100 | 1827 | 1868 | 1844.34 | 8.72 |
| $\text{MRM}_{0.23}$<2500,2.5> | 100 | 1821 | 1858 | 1839.36 | 8.20 |
| AGG<2500> | | | | 1728 | |
| IOM<5000,2.5> | 100 | 3663 | 3836 | 3748.45 | 36.82 |
| $\text{MRM}_{0.24}$<5000,2.5> | 100 | 3645 | 3764 | 3695.03 | 23.32 |
| $\text{MRM}_{0.235}$<5000,2.5> | 100 | 3616 | 3698 | 3648.45 | 14.88 |
| $\text{MRM}_{0.23}$<5000,2.5> | 100 | 3578 | 3677 | 3634.78 | 17.41 |
| AGG<5000> | | | | 3432 | |

Table 11. ANOVA and LSD Results for Minimum Population Size <100, 1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <100,1> | Between Groups | 2134.76 | 3 | 711.59 | 163.36 | <.001* |
| | Within Groups | 1725.00 | 396 | 4.35 | | |
| | Total | 3859.76 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <100,1> | IOM | MRM$_{0.24}$ | -0.08 | .2952 | .787 |
| | IOM | MRM$_{0.235}$ | -4.66 | .2952 | <.001* |
| | IOM | MRM$_{0.23}$ | -4.66 | .2952 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | -4.58 | .2952 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | -4.58 | .2952 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | .00 | .2952 | 1.000 |

* The mean difference is significant at the .05 level.

Table 12. ANOVA and LSD Results for Minimum Population Size <1000, 1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <1000,1> | Between Groups | 23060.94 | 3 | 7686.98 | 129.68 | <.001* |
| | Within Groups | 23473.53 | 396 | 59.27 | | |
| | Total | 46534.48 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <1000,1> | IOM | MRM$_{0.24}$ | -6.76 | 1.08882 | <.001* |
| | IOM | MRM$_{0.235}$ | -16.65 | 1.08882 | <.001* |
| | IOM | MRM$_{0.23}$ | -18.78 | 1.08882 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | -9.89 | 1.08882 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | -12.02 | 1.08882 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | -2.13 | 1.08882 | .051 |

* The mean difference is significant at the .05 level.

Table 13. ANOVA and LSD Results for Minimum Population Size <2500,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| 2500,1> | Between Groups | 82505.307 | 3 | 27501.769 | 87.686 | <.001* |
| | Within Groups | 124200.890 | 396 | 313.639 | | |
| | Total | 206706.197 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <2500,1> | IOM | MRM$_{0.24}$ | -2.44 | 2.50455 | .331 |
| | IOM | MRM$_{0.235}$ | -25.45 | 2.50455 | <.001* |
| | IOM | MRM$_{0.23}$ | -33.26 | 2.50455 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | -23.01 | 2.50455 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | -30.82 | 2.50455 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | -7.81 | 2.50455 | .002* |

* The mean difference is significant at the .05 level.

Table 14. ANOVA and LSD Results for Minimum Population Size <5000,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <5000,1> | Between Groups | 151141.528 | 3 | 50380.509 | 55.711 | <.000* |
| | Within Groups | 358113.150 | 396 | 904.326 | | |
| | Total | 509254.678 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <5000,1> | IOM | MRM$_{0.24}$ | 3.56 | 4.25283 | .403 |
| | IOM | MRM$_{0.235}$ | -20.40 | 4.25283 | <.001* |
| | IOM | MRM$_{0.23}$ | -45.23 | 4.25283 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | -23.96 | 4.25283 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | -48.79 | 4.25283 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | -24.83 | 4.25283 | <.001* |

* The mean difference is significant at the .05 level.

Table 15 ANOVA and LSD Results for Minimum Population Size <100, 2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <100,2.5> | Between Groups | 15416.527 | 3 | 5138.842 | 121.677 | <.001* |
| | Within Groups | 16724.410 | 396 | 42.233 | | |
| | Total | 32140.938 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <100,2.5> | IOM | $MRM_{0.24}$ | .750 | .9191 | .415 |
| | IOM | $MRM_{0.235}$ | -12.03 | .9191 | <.001* |
| | IOM | $MRM_{0.23}$ | -12.03 | .9191 | <.001*S |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | -12.78 | .9191 | <.001*S |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | -12.78 | .9191 | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | .000 | .9191 | 1.000 |

* The mean difference is significant at the .05 level.

Table 16 ANOVA and LSD Results for Minimum Population Size <1000,2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <1000,2.5> | Between Groups | 186593.308 | 3 | 62197.769 | 267.158 | <.001* |
| | Within Groups | 92193.670 | 396 | 232.812 | | |
| | Total | 278786.978 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <1000,2.5> | IOM | $MRM_{0.24}$ | -34.9300 | 2.15783 | <.001* |
| | IOM | $MRM_{0.235}$ | -48.14 | 2.15783 | <.001* |
| | IOM | $MRM_{0.23}$ | -56.64 | 2.15783 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | -13.2100 | 2.15783 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | -21.71 | 2.15783 | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | -8.50 | 2.15783 | <.001* |

* The mean difference is significant at the .05 level.

Table 17 ANOVA and LSD Results for Minimum Population Size <2500,2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <2500,2.5> | Between Groups | 457598.187 | 3 | 152532.729 | 229.692 | .000\ |
| | Within Groups | 262974.250 | 396 | 664.076 | | |
| | Total | 720572.437 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <2500,2.5> | IOM | $MRM_{0.24}$ | -51.25 | 3.64438 | <.001* |
| | IOM | $MRM_{0.235}$ | -78.35 | 3.64438 | <.001* |
| | IOM | $MRM_{0.23}$ | -86.55 | 3.64438 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | -27.10 | 3.64438 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | -35.30 | 3.64438 | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | -8.20 | 3.64438 | .025* |

* The mean difference is significant at the .05 level.

Table 18 ANOVA and LSD Results for Minimum Population Size <5000,2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <5000,2.5> | Between Groups | 1224211.228 | 3 | 408070.409 | 266.529 | .000 |
| | Within Groups | 606298.35 | 396 | 1531.056 | | |
| | Total | 1830509.578 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <5000,2.5> | IOM | $MRM_{0.24}$ | -63.65 | 5.53364 | <.001* |
| | IOM | $MRM_{0.235}$ | -121.22 | 5.53364 | <.001* |
| | IOM | $MRM_{0.23}$ | -142.36 | 5.53364 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | -57.57 | 5.53364 | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | -78.71 | .53364 | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | -21.14 | 5.53364 | <.001* |

* The mean difference is significant at the .05 level.

Table 19 ANOVA and LSD Results for Maximum Population Size <100,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <100,1> | Between Groups | 1380.127 | 3 | 460.042 | 307.830 | <.001* |
| | Within Groups | 591.810 | 396 | 1.494 | | |
| | Total | 1971.937 | 399 | | | |
| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | | Sig. |
| <100,1> | IOM | $MRM_{0.24}$ | -.010 | .1729 | | .954 |
| | IOM | $MRM_{0.235}$ | 3.710 | .1729 | | <.001* |
| | IOM | $MRM_{0.23}$ | 3.710 | .1729 | | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | 3.720 | .1729 | | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | 3.720 | .1729 | | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | .000 | .1729 | | 1.000 |

* The mean difference is significant at the .05 level.

Table 20 ANOVA and LSD Results for Maximum Population Size, <1000,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <1000,1> | Between Groups | 16914.770 | 3 | 5638.257 | 269.720 | <.001* |
| | Within Groups | 8278.020 | 396 | 20.904 | | |
| | Total | 25192.790 | 399 | | | |
| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | | Sig. |
| <1000,1> | IOM | $MRM_{0.24}$ | 6.25 | .64659 | | <.001* |
| | IOM | $MRM_{0.235}$ | 14.5100 | .64659 | | <.001* |
| | IOM | $MRM_{0.23}$ | 16.1000 | .64659 | | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.235}$ | 8.2600 | .64659 | | <.001* |
| | $MRM_{0.24}$ | $MRM_{0.23}$ | 9.8500 | .64659 | | <.001* |
| | $MRM_{0.235}$ | $MRM_{0.23}$ | 1.5900 | .64659 | | .014* |

* The mean difference is significant at the .05 level.

Table 21 ANOVA and LSD Results for Maximum Population Size, <2500,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <2500,1> | Between Groups | 55110.127 | 3 | 18370.042 | 133.552 | <.001* |
| | Within Groups | 54469.570 | 396 | 137.549 | | |
| | Total | 109579.697 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <2500,1> | IOM | $\text{MRM}_{0.24}$ | 1.89 | 1.65861 | .255 |
| | IOM | $\text{MRM}_{0.235}$ | 21.48 | 1.65861 | <.001* |
| | IOM | $\text{MRM}_{0.23}$ | 26.70 | 1.65861 | <.001* |
| | $\text{MRM}_{0.24}$ | $\text{MRM}_{0.235}$ | 19.59 | 1.65861 | <.001* |
| | $\text{MRM}_{0.24}$ | $\text{MRM}_{0.23}$ | 24.81 | 1.65861 | <.001* |
| | $\text{MRM}_{0.235}$ | $\text{MRM}_{0.23}$ | 5.22 | 1.65861 | .002* |

* The mean difference is significant at the .05 level.

Table 22 ANOVA and LSD Results for Maximum Population Size, <5000,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <5000,1> | Between Groups | 107224.648 | 3 | 35741.549 | 75.821 | <.001* |
| | Within Groups | 186671.750 | 396 | 471.393 | | |
| | Total | 293896.398 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <5000,1> | IOM | $\text{MRM}_{0.24}$ | -2.61 | 3.07048 | .396 |
| | IOM | $\text{MRM}_{0.235}$ | 18.77 | 3.07048 | <.001* |
| | IOM | $\text{MRM}_{0.23}$ | 38.05 | 3.07048 | <.001* |
| | $\text{MRM}_{0.24}$ | $\text{MRM}_{0.235}$ | 21.38 | 3.07048 | <.001* |
| | $\text{MRM}_{0.24}$ | $\text{MRM}_{0.23}$ | 40.66 | 3.07048 | <.001* |
| | $\text{MRM}_{0.235}$ | $\text{MRM}_{0.23}$ | 19.28 | 3.07048 | <.001* |

* The mean difference is significant at the .05 level.

Table 23 ANOVA and LSD Results for Maximum Population Size <100,2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <100,2.5> | Between Groups | 3330.270 | 3 | 1110.090 | 127.182 | <.001* |
| | Within Groups | 3456.440 | 396 | 8.728 | | |
| | Total | 6786.710 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <100,2.5> | IOM | MRM$_{0.24}$ | .1400 | .41781 | .738 |
| | IOM | MRM$_{0.235}$ | 5.8400 | .41781 | <.001* |
| | IOM | MRM$_{0.23}$ | 5.8400 | .41781 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | 5.7000 | .41781 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | 5.7000 | .41781 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | .0000 | .41781 | 1.000 |

* The mean difference is significant at the .05 level.

Table 24 ANOVA and LSD Results for Maximum Population Size, <1000,2.5>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <1000,2.5> | Between Groups | 119667.887 | 3 | 39889.296 | 483.188 | <.001* |
| | Within Groups | 32691.55 | 396 | 82.554 | | |
| | Total | 152359.438 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <1000,2.5> | IOM | MRM$_{0.24}$ | 22.57 | 1.28495 | <.001* |
| | IOM | MRM$_{0.235}$ | 37.74 | 1.28495 | <.001* |
| | IOM | MRM$_{0.23}$ | 45.28 | 1.28495 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | 15.17 | 1.28495 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | 22.71 | 1.28495 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | 7.54 | 1.28495 | <.001* |

* The mean difference is significant at the .05 level

Table 25 ANOVA and LSD Results for Maximum Population Size, <2500,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <2500,2.5> | Between Groups | 316904.248 | 3 | 105634.749 | 542.272 | <.001* |
| | Within Groups | 77140.950 | 396 | 194.800 | | |
| | Total | 394045.198 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <2500,2.5> | IOM | MRM$_{0.24}$ | 41.05 | 1.97383 | <.001* |
| | IOM | MRM$_{0.235}$ | 66.14 | 1.97383 | <.001* |
| | IOM | MRM$_{0.23}$ | 71.12 | 1.97383 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | 25.09 | 1.97383 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | 30.07 | 1.97383 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | 4.98 | 1.97383 | .012* |

* The mean difference is significant at the .05 level.

Table 26 ANOVA and LSD Results for Maximum Population Size, <5000,1>

| | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| <5000,2.5> | Between Groups | 794029.827 | 3 | 264676.609 | 436.717 | <.001* |
| | Within Groups | 239999.570 | 396 | 606.060 | | |
| | Total | 1034029.397 | 399 | | | |

| Dependent Variable | Model I | Model J | Mean Diff (I-J) | Std. Error | Sig. |
|---|---|---|---|---|---|
| <5000,2.5> | IOM | MRM$_{0.24}$ | 53.42 | 3.48155 | <.001* |
| | IOM | MRM$_{0.235}$ | 100.00 | 3.48155 | <.001* |
| | IOM | MRM$_{0.23}$ | 113.67 | 3.48155 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.235}$ | 46.58 | 3.48155 | <.001* |
| | MRM$_{0.24}$ | MRM$_{0.23}$ | 60.25 | 3.48155 | <.001* |
| | MRM$_{0.235}$ | MRM$_{0.23}$ | 13.67 | 3.48155 | <.001* |

* The mean difference is significant at the .05 level

### 4.5.3 Discussion

Figures 41 and 42 show that the MRM is bounded by the aggregate logistic model (LRM) and the logistic IOM (HRM). Moreover, the graphs show that the MRMs are

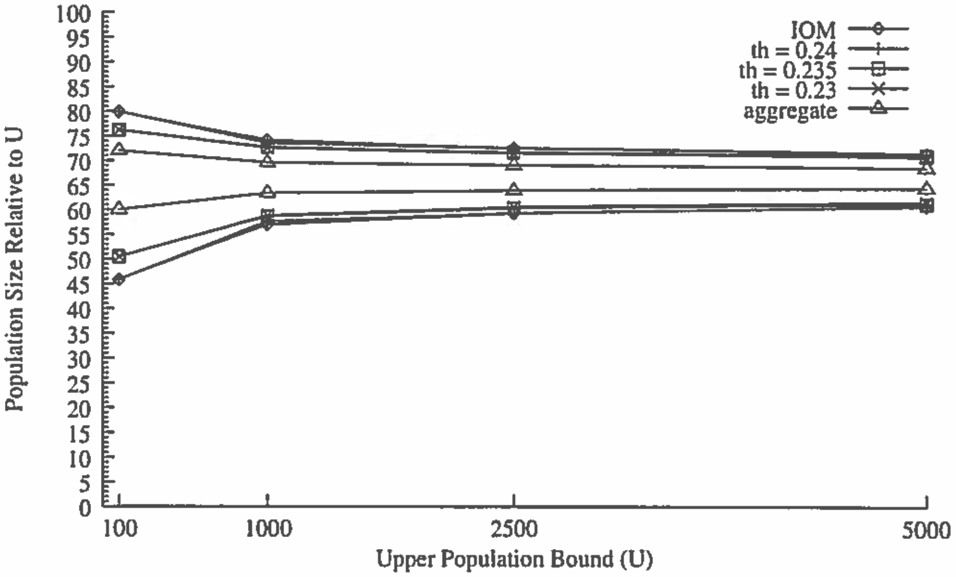

Figure 41. Ratio of Population Maximum and Minimum Population Sizes Relative to upper population bound, Δ = 1.0

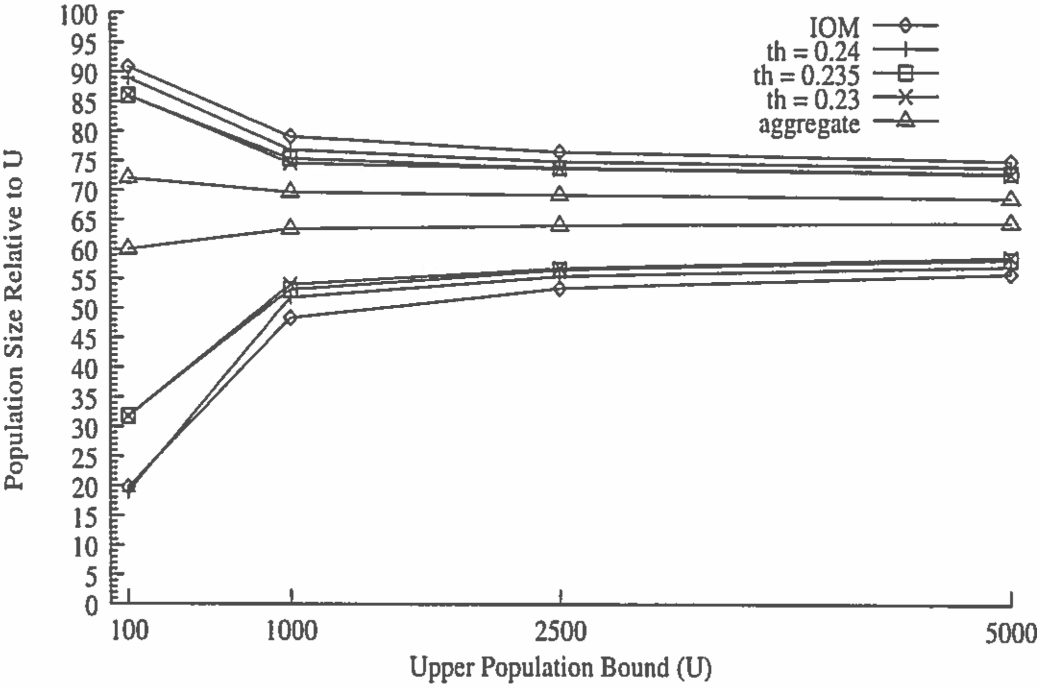

Figure 42. Ratio of Population Maximum and Minimum Population Sizes Relative to upper population bound, Δ = 2.5

bounded by the switching threshold value. That is, larger threshold values are bounded by smaller threshold values and the IOM. As the threshold increases, the time series produced by the MRM changes from an LRM-like time series to an HRM-like time series. If increasing the threshold causes a monotonic change in time series behavior, then the threshold value gives us a mechanism for control the accuracy of the MRM.

The data suggest that the switching threshold has a direct bearing on the consistency of the MRM time series and the HRM time series, which may provide a means of predicting the accuracy of the MRM. For example, Figures 43 - 46 are plots of the maximum and minimum population sizes, relative to the upper population bound, *U*.

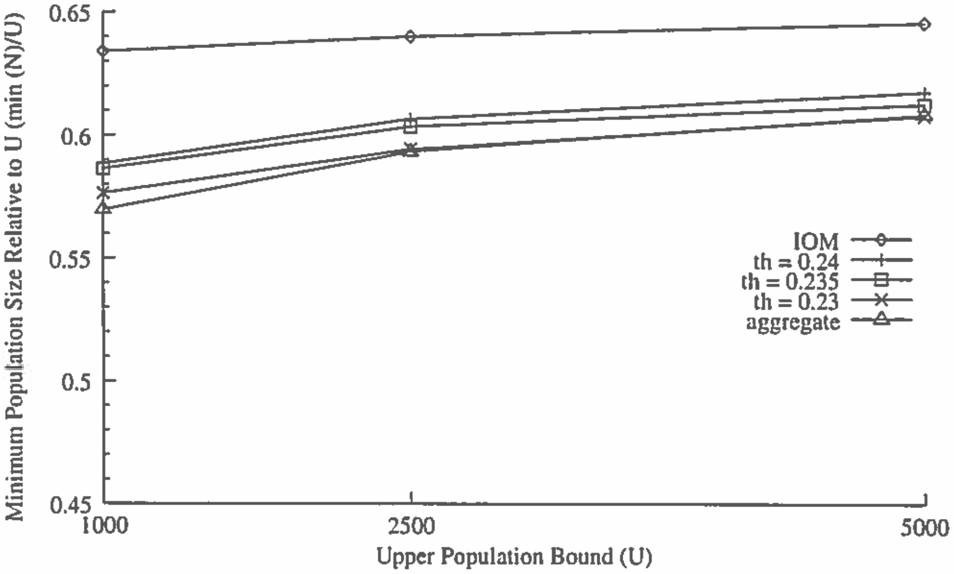

Figure 43. Mimimum Population Size Relative to *U*, for Δ = 1.0

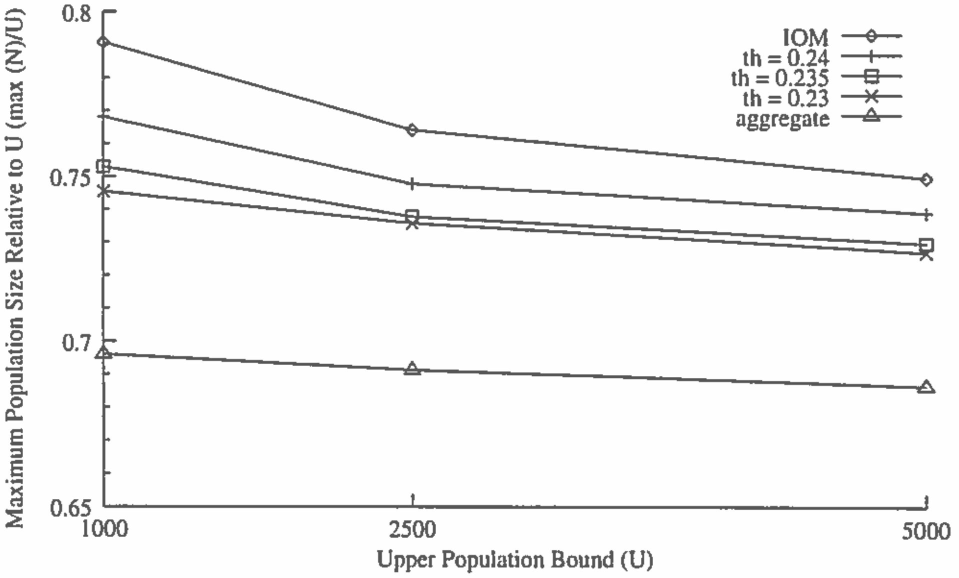

Figure 44. Maximum Population Size Relative to $U$, for $\Delta = 2.5$

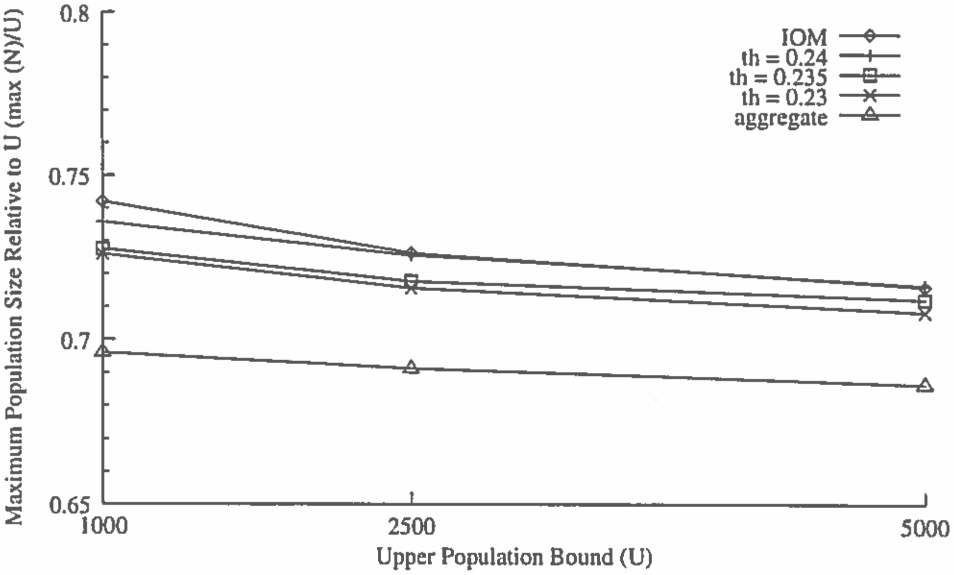

Figure 45. Maximum Population Size Relative to $U$, for $\Delta = 1.0$

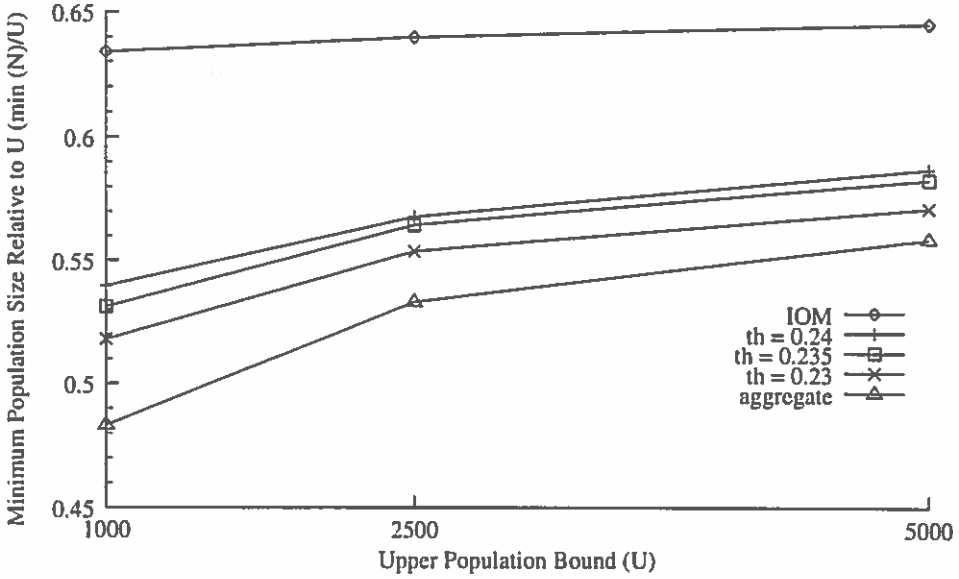

Figure 46. Minimum Population Size Relative to *U*, for Δ = 2.5

Each plot shows the family of model curves (i.e. IOM, $MRM_{0.23}$, $MRM_{0.235}$ and $MRM_{0.24}$) for upper population bound in the set, U = {1000, 2500, 5000}. If a modeler decides to use a particular threshold value, for a given problem, he or she can predict the accuracy of a simulation that uses that threshold value.

Besides the boundedness of the MRM, the experiment also suggests that the MRMs are ordered by threshold values. This result is stronger than boundedness as it suggests that the time series changes from an LRM-like series to an HRM-like series as the threshold increases. Specifically, if increasing the threshold causes a monotonic change in time series behavior, then the threshold value gives us a mechanism for control the accuracy of the MRM. With the current set of tests, we cannot affirm the correctness of this assertion, however, the possibility warrants further investigation.

The threshold value is an important factor in controlling MRM results. The data suggest that switching threshold has a direct bearing on the similarity between the MRM time series and the IOM time series, which may provide a means of predicting the accuracy of the MRM. For example, a model of "similarity to IOM" versus switching threshold can be developed using the maximum population size. Figure 46-49 shows a series of curves plotting the relative difference between the IOM and MRM over the range of threshold values tested. Each graph consistently shows a monotonic decrease in error as the threshold value increases and a monotonic decrease in error as the population size increases. Therefore, MRM accuracy is likely to improve as the switching threshold or upper population bound increases. A superficial analysis of these graphs suggests that the decrease in relative error with respect to the threshold may be linear or possibly asymptotic. If this is true, the threshold value may give modelers fine control over the accuracy of the model. A more detailed analysis of the characteristics of the relative error is warranted.

## 4.6 General Discussion

The three tests described in this chapter illustrated two significant points. First, valid population models can be constructed with STEMSS and STEMSS-MRM. Second, the MRMs tested in this chapter confirmed the hypothesis stated in Chapter I. The logistic and the predator-prey models demonstrated the use of STEMSS in developing basic population models and the yellow perch models showed its use in developing complex models. Each of these models produced simulations bounded by the constituent high-resolution and low-resolution models. Consequently, the models were consistent with the central hypothesis of the dissertation, that the results of the multiple resolution population models were bounded by and arbitrarily close to the results of constituent single resolution models.

The affirmation of the hypothesis, based on the results of the experiments discussed in this chapter, implies that further investigation into multiple-resolution modeling is warranted. Some possible topics for further studies include the development and analysis of more complex models, the development and analysis of other model switching policies and techniques, the investigation of boundedness in other time series characteristics and the impact of resegmentation on model accuracy.

# CHAPTER V

# MRM PERFORMANCE STUDIES

## 5.1 Introduction

The results reported in Chapter IV described the relationship between the model switching threshold and MRM accuracy. The data showed that as the switching threshold increased, the accuracy of the MRM increased. Given that an increase in the switching threshold value implies an increase in HRM activity, the results from Chapter IV imply that as HRM activity increases the accuracy of the model increases. Unfortunately, an increase in HRM activity causes an increase in simulation execution time. Therefore, an increase in accuracy implies an increase in execution time. This chapter will investigate the impact of the switching threshold on the MRM execution time.

The relationship between accuracy and execution time is particularly important when the model is used to decide a course of action during an environmental crisis. Consider the yellow perch model, described in section 4.5. The IOM model required 20 hours to simulate 30 years of population data while the aggregate model required 20 minutes for the same simulation. The steady-state population size, predicted by the IOM, is approximately 250% of the population size predicted by the aggregate model. The

MRM required 10 hours to execute the same simulation and reduced this error to 120%, so the MRM simulation results in a loss of accuracy and a decrease in execution time. If the results from this model must be collected within 10 hours, then a 120% loss in accuracy is expected. If this loss in accuracy is acceptable, then the model is successful. That is, it delivers an acceptable simulation within the time constraints of the user.

To use STEMSS-MRM effectively, the modeler must be able to predict, with confidence, the accuracy and execution time of the MRM. For example, during the initial phases of an oil spill in the coastal United States, the U. S. Coast Guard is responsible for directing the federal government's response to the spill (EPA, 2002). The response involves deciding where to place specific resources and doing so in a timely manner. Any method designed to aid in making this decision must be able to produce results quickly and with acceptable accuracy.

To understand how to predict execution time, consider the intent of an MRM. In a sense, the desired outcome of an MRM is the creation of a model that produces the most accurate results in a specified amount of time. That is, given all possible traces for the HRM, LRM and every possible model switch and enough time, the traces could be combined to construct every possible MRM. The accuracy of the MRMs, relative to the HRM, can be plotted against execution time. The modeler can use these plots to determine the minimum execution time required to produce for a given level of accuracy. Though such an analysis is impractical, due to the cost of collecting and analyzing the data, it does provide some insight into the factors that affect performance. In essence, the approach is designed to find the sequence of model switches that produces the minimum execution time across a range of model accuracies. As the sequence cannot be known *a priori*, an approach to estimating the performance cost of an arbitrary switching policy is necessary if STEMSS is to be used effectively.

To estimate the cost of a switching policy, it is necessary to estimate the overhead cost due to STEMSS, the constituent models and the user-defined switching methods. This estimate is expressed as:

$$T_{exe} = T_{STEMSS} + N_{HRM}T_{HRM} + N_{LRM}T_{LRM} + N_{swh}T_{swh} + N_{swl}T_{swh} \tag{22}$$

where:

$T_{exe}$ is the execution time of the model,

$T_{STEMSS}$ is the overhead cost of the STEMSS methods,

$N_{HRM}$ is the number of HRM transitions performed in a simulation,

$N_{LRM}$ is the number of LRM transitions performed in a simulation,

$T_{HRM}$ is the time to execute the HRM,

$T_{LRM}$ is the time to execute the LRM,

$N_{swh}$ is the number of HRM switches performed in a simulation,

$N_{swl}$ is the number of LRM transitions performed in a simulation,

$T_{swh}$ is the time to execute the switch from the LRM to the HRM, and

$T_{swl}$ is the time to execute the switch from the LRM to the HRM.

The timing parameters (i.e., $T_{exe}$, $T_{STEMSS}$, $T_{HRM}$, $T_{LRM}$, $T_{swh}$ and $T_{swl}$) can be estimated by profiling the simulation to measure the time required by each routine and associating those times with the performance equation parameters. These parameters may be constants, they may be dependent on the number of models associated with the active model, $N_{tr}$ or the may depend on the number of model switches made by the simulation, $N_{sw}$. For example, in a population model, when the IOM is active, $N_{tr}$ is the population size. When the aggregate model is active, $N_{tr}$ is the number of modeled populations. Since the timing parameters depend on $N_{tr}$ or $N_{sw}$, profile data can be used to compute the average value of the timing parameters as a function of $N_{tr}$ or $N_{sw}$.

The frequency parameters (i.e., $N_{tr}$, $N_{sw}$, $N_{HRM}$, $N_{LRM}$, $N_{swl}$, $N_{swl}$) can also be estimated from the profiling data, but this approach may be counterproductive. Each parameter can be associated with a specific function call. For example, the number of switches from the HRM to the LRM, NLRM, is the number of calls to the user-defined switching function. Unlike the timing data, however, the frequency data is dependent on model execution. The number of calls to a given function depends on the model and the duration of the simulation. This approach to estimating the frequency parameters has two potential drawbacks with respect to this analysis. First, these parameters are model dependent, so they require a complete characterization of the model's behavior to estimate. For a simple model, with a small parameter space, the characterization should be tractable. However, for a model with a large parameter space and complex behaviors and interactions, a complete characterization will be difficult. The second drawback, actually a benefit from the modeler's perspective, is that a complete characterization would make the need for further simulation unnecessary.

Since characterizing complex models is a difficult problem, an alternate method for estimating frequency parameters is needed. One technique is to rely on the domain knowledge of the user to estimate the number of model transitions and switches. A second technique is the incorporation of the performance equation into the running simulation. The MRM, as it is currently implemented, uses a specific state value to decide when to switch the active model. Conceivably, the performance equation might be used to determine optimal switching times by using a system monitor to estimate the performance parameters while the simulation is running. A third technique is to adapt the switch detection logic to a steering mechanism. With a monitor providing performance information, the user can manually intervene in (or steer) a running simulation by reducing accuracy to speed up the execution, or to slow down the simulation to improve accuracy. Both of these alternatives require an accurate performance equation.

The development of a performance equation assumes that the MRM will, in fact, improve execution time. To validate this assumption, this chapter includes a performance study in which the logistic model suite, described and analyzed in Chapter IV, is used to validate the central hypothesis of this dissertation, that the MRM can improve the execution time of the HRM. Without this validation, a performance equation is irrelevant.

The second study discussed in this chapter is designed to develop the performance equation for the logistic MRM and to demonstrate its use. In the first step of this study, the STEMSS control flow is used to specify the performance equation of the logistic MRM and profile data is collected and analyzed to derive parameters for this equation. The logistic MRM performance equation is then validated by using it to predict the execution times of the simulations used in the first step. In the final step of this study the validated performance equation is used to predict the execution time of the logistic MRM for a previously untested parameter set. The final step is a demonstration of the use of STEMSS in a crisis.

## 5.2 Timing Analysis of Logistic Equation

### 5.2.1 Purpose

The central hypothesis of this dissertation asserts that, in part, an MRM can improve model execution time relative to the execution time of the LRM. This assertion cannot be proven, in general, as the performance characteristics of each model are unique and some model characteristics may defy any attempt to produce an acceptable MRM. Nevertheless, the assertion can be validated (or invalidated), for a particular model, by characterizing the execution time of the model with respect to the model's parameters and switching policies.

In this analysis, the hypothesis is validated with the logistic MRM. The experiment compares the execution time with the upper population bound, $U$, the stochastic birth rate parameter, $\Delta$ and the model's switching threshold, *th*. The performance study also provides an analysis of the changes in execution time due to HRM activity. HRM activity is the ratio of the number of times the HRM is active to the number of MRM events processed by the simulation. This analysis shows that switching threshold is a valid parameter for controlling MRM execution. That is, an increase in the switching threshold will increase the execution time.

### 5.2.2 Procedure

The logistic model suite was used in this experiment (see Table 27) and the simulation procedures are identical to experiment 4.5, except the model was compiled using the TAU performance system (TAU, 2002). TAU was used to collect execution time data and HRM activity. For the logistic model, HRM activity is:

$$H_a = \frac{N_{iom}}{N_{iom} + N_{agg}} \tag{23}$$

where $H_a$ is HRM activity, $N_{iom}$ is the number of times the IOM model is active and $N_{agg}$ is the number of times the aggregate model is active. For this test, $H_a$ is IOM activity.

Table 27: Logistic Model Performance Data

| <U, Δ> | avg | std | %HRM activity | avg | std | %HRM activity |
|---|---|---|---|---|---|---|
| <100, 1.0> | | | | <100, 2.5> | | |
| MRM$_{0.23}$ | 0.121 | 0.0058 | 73.5 | 0.108 | 0.0055 | 67.9 |
| MRM$_{0.235}$ | 0.122 | 0.0061 | 75.6 | 0.110 | 0.0037 | 67.9 |
| MRM$_{0.24}$ | 0.158 | 0.0050 | 99.4 | 0.148 | 0.0046 | 98.5 |
| IOM | 0.149 * | 0.0270 | 100 | 0.144 * | 0.0259 | 100 |
| <1000, 1.0> | | | | <1000, 2.5> | | |
| MRM$_{0.23}$ | 2.588 | 0.2120 | 67.4 | 2.267 | 0.1479 | 61.5 |
| MRM$_{0.235}$ | 3.158 | 0.2092 | 83.1 | 2.595 | 0.1779 | 70.6 |
| MRM$_{0.24}$ | 3.730 | 0.1918 | 96.3 | 3.085 | 0.2169 | 82.7 |
| IOM | 3.752 | 0.3036 | 100 | 3.586 | 0.0587 | 100 |
| <2500, 1.0> | | | | <2500, 2.5> | | |
| MRM$_{0.23}$ | 22.221 | 4.6768 | 69.7 | 20.03 | 3.2569 | 63.4 |
| MRM$_{0.235}$ | 30.681 | 5.7099 | 90.0 | 25.41 | 5.2474 | 76.3 |
| MRM$_{0.24}$ | 34.954 | 5.3828 | 98.5 | 31.81 | 4.8771 | 88.3 |
| IOM | 33.240 * | 4.7261 | 100 | 34.58 | 5.1511 | 100 |
| <5000, 1.0> | | | | <5000, 2.5> | | |
| MRM$_{0.23}$ | 280.35 | 32.591 | 72.2 | 243.70 | 22.721 | 66.1 |
| MRM$_{0.235}$ | 379.63 | 13.725 | 93.1 | 304.33 | 28.737 | 79.4 |
| MRM$_{0.24}$ | 408.78 | 6.0192 | 98.9 | 376.90 | 12.565 | 94.7 |
| IOM | 422.81 | 10.459 | 100 | 416.55 | 8.8088 | 100 |

* execution time for the IOM model is less than the execution time of the MRM$_{0.24}$ simulation.

For all tests in this simulation, the average growth rate for individuals was set at $R = 3.0$, the upper population bound was selected from the set $U = \{100, 1000, 2500, 5000\}$, the stochastic term for birth rate was selected from $\Delta = \{1.0, 2.5\}$ and the threshold value was selected from $th = \{0.23, 0.235, 0.24\}$. Thirty simulations for each combination of parameters were run and the timing data for specific functions was collected. The average execution times for each parameter set were collected and plotted as families of threshold value curves.

### 5.2.3 Results

Table 27 presents the average and standard deviation of the execution time and HRM activity, as a percentage, for each parameter set. In general, the average execution time increases as the threshold value increases except the <100, 1.0>, <100, 2.5> and <2500, 1.0> data sets show a decrease between $MRM_{0.24}$ and IOM. The decrease is small (6.4%, 2.8% and 5.1% of the IOM time for <100, 1.0>, <100, 2.5> and <2500, 1.0> respectively), so the decrease may be an artifact of the stochasticity in the model or it may be the result of simulation overhead, these assertions are still under investigation. In either case, these inconsistencies suggest that there may be a limiting level of IOM activity in an MRM that, when crossed, causes an increase in MRM, relative to the IOM, execution

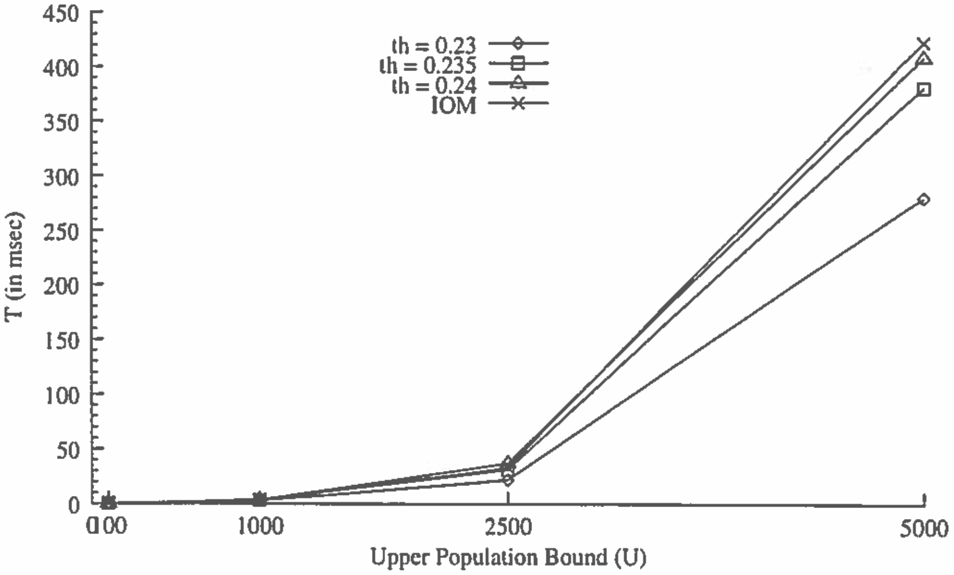

Figure 47. Execution Time of Logistic MRM with $\Delta = 1.0$

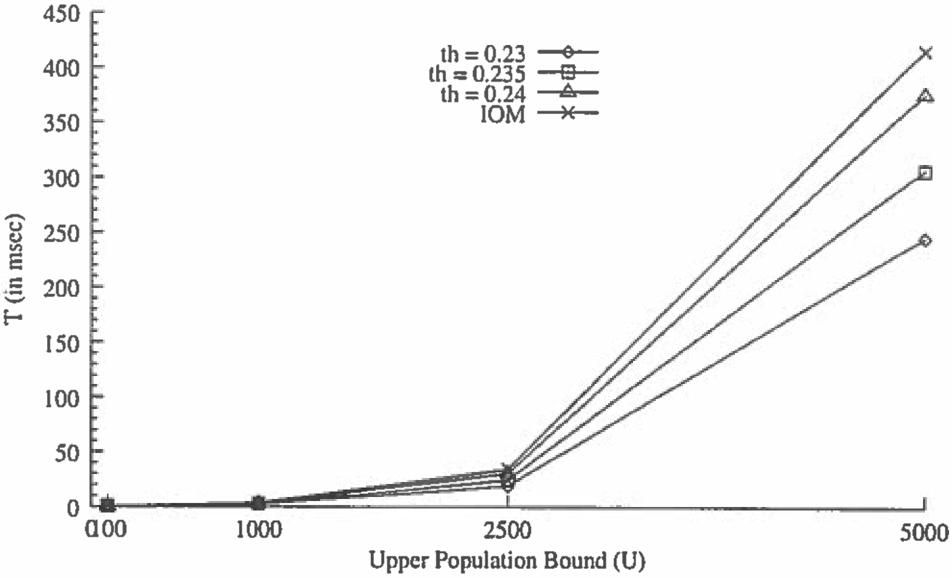

Figure 48. Execution Time of Logistic MRM with $\Delta = 2.5$

time. The execution times for the $\Delta = 1.0$ and 2.5 switching thresholds are shown in Figures 47 and 48, respectively.

### 5.2.4 Discussion

The effect of the model parameters and threshold values on the execution time of the simulation reaffirms the central hypothesis of this dissertation. That is, the data show that the execution time of an MRM can be less then that of the high-resolution model. The data also suggest that execution time of an MRM can exceed that of the HRM when the amount of IOM activity is high. This is an undesirable result which will be characterized in future work. Nevertheless, the observations from this study suggest that the switching threshold can control execution time. This suggestion, coupled with the observations regarding model accuracy made in Chapter IV, indicate that a user can predict the execution time and accuracy of a particular parameterization based on the switching threshold value.

## 5.3 Logistic MRM Performance Equation

The analysis in section 5.2 suggests that an MRM can improve the execution time of simulation and that a user can control the execution time, but it does not identify the factors that influence execution time. These factors, if properly exploited, could provide the user with more control over the execution time of the simulation and may form the basis for automating MRM switching control. The study performed in this section uses a detailed analysis of the simulator's control flow and profile data of the logistic MRM to identify the significant performance factors and to develop a performance equation based on these factors. It is important to keep in mind that the logistic model represents one of the simplest possible models to construct and analyze, so the performance equation must provide good results in this case or it cannot be expected to work in other, more complex, cases.

Figure 49 shows flow of control in a STEMSS-MRM simulation. The simulation begins with the event loop removing the head of the future event list (*fel*) and invoking the MRM transition function associated with that event. The event may be reinserted into the *fel*, depending on the outcome of the transition function. The MRM transition function

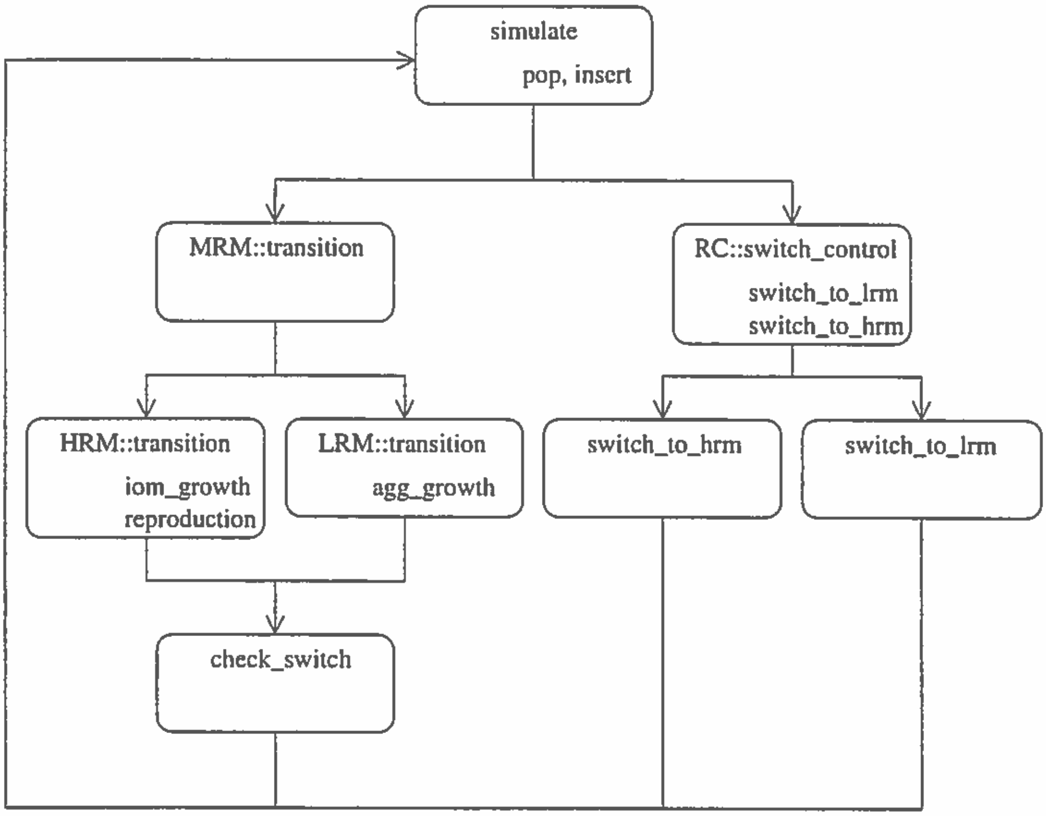

Figure 49. MRM Control Flow

will invoke either an LRM transition function or an HRM transition function, depending on which model is active, then test the model switching condition. If the condition is met, the MRM will set up the switch, by activating the resolution controller. When the MRM transition completes, control is returned to the event loop and the process repeats.

The execution time of a simulation is the sum of the cumulative cost of executing the high and low-resolution models, the cumulative cost of model switching and cumulative cost of the simulation overhead. The cost of executing the high and low-resolution models is the sum of cumulative execution time of each transition function, $T_{tr,i}$. The cost of model switching is the cumulative execution time required by the switching function, $T_{sw}$. The simulation overhead costs consisting of the time for each iteration of the event loop, $T_{so}$, the overhead costs of event processing, $T_{po}$ and model switching, $T_{wo}$. For an arbitrary model, the performance equation can be rewritten as:

$$T_{exe} = T_{so} + T_{po} + T_{wo} + \sum_{i \in \{\mathrm{trans}\}} T_{tr,i} + T_{sw} + \varepsilon \qquad (24)$$

where *{trans}* is the set of model transition functions and ε is the setup and cost which is negligible compared with the rest of the simulation, so it will be ignored. To use equation 24, the specific performance behavior of each factor must be carefully investigated.

The processing overhead, $T_{po}$, is the overhead cost of processing model transition events. As there are only two types of transitions in STEMSS, network and model events, the total overhead cost of event processing, during the course of a simulation, is the sum of the network and model transition overhead costs, or:

$$T_{po} = N_m H_m + N_n H_n \qquad (25)$$

where $H_m$ is the exclusive time required to process a model transition, per method call and $H_n$ exclusive time required to process a network transition, per method call. Both $H_m$ and $H_n$ are model independent values. The value, $T_{po}$, is the total overhead cost of model execution. To determine the overhead cost of event processing per event, and thus the model independent cost of event processing, $T_{po}$ is reported as an average cost per event, $F_{po}$, and is computed as:

$$F_{po} = \frac{T_{po}}{(N_m + N_n)} \tag{26}$$

After processing a model event, the MRM transition function invokes a switch detection method, defined by the user. If the switching condition is met, the MRM will switch the active model, using the STEMSS switch configuration mechanism. The total overhead cost of switching, $T_{wo}$, is the cost of switch detection plus the cost of setting up the switch. The MRM tests for a switch-to-low condition when the HRM is active and the switch-to-high condition when the LRM is active. Therefore, the number of switch-to-low tests is the same as the number of HRM transitions and the number of switch-to-high tests is the same as the number of LRM transitions. Given the relationship between the switching tests and the resolution of the active model, the overhead cost of model switching is:

$$T_{wo} = N_{tr}H_aT_{swl} + N_{tr}L_aT_{swh} + N_{swl}T_{sswl} + N_{swh}T_{sswh} \tag{27}$$

where,

$N_{tr}$ is the number of transition functions invoked by the event loop, over the course of the simulation,

$H_a$ is the ratio of high-resolution models to $N_{tr}$,

$L_a$ is the ratio of low-resolution models to $N_{tr}$,

$T_{sswl}$ is the cost for setting up the switch to the high-resolution model and,

$T_{sswh}$ is the cost for setting up the switch to the low-resolution model,

$N_{swh}$ is the number of switches to the high-resolution model,

$N_{swl}$ is the number of switches to the low-resolution model,

$T_{swh}$ is the cost of making the high-resolution model active, and

$T_{swl}$ is the cost of making the low-resolution model active.

The times required to set up a switch, $T_{swh}$ and $T_{swl}$, are model dependent. If the MRM uses a switching threshold policy, as in the logistic MRM, then both values should be constant, though not necessarily equal. The number of switches to the high-resolution model and low-resolution model are such that $N_{swh} = N_{swl} \quad 1$, and$N_{sw} = N_{swh} + N_{swl}$, where $N_{sw}$ is the total number of switches. With some algebraic manipulation, equation 27 can be written as:

$$T_{wo} = N_{tr}(H_a T_{swl} + L_a T_{swh}) + \frac{1}{2} N_{sw}(T_{sswl} + T_{sswh}) \tag{28}$$

The cost of executing the constituent model is the cumulative time required to execute each model transition. For an arbitrary transition function, $i$, the cumulative time required to execute $i$ is the total time required to execute the function, $T_{tr,i}$, times the number of times the method is invoked, $N_{tr,i}$. The number times the method is invoked can be stated as a ratio, $R_{tr,i} = N_{tr,i}/N_{tr}$, and the total cost of model execution can be stated, in an algebraically convenient form as,

$$N_{tr} \sum_{i \in \{\text{trans}\}} R_{tr,i} T_{tr,i} \tag{29}$$

The time required to switch the active model depends on the STEMSS method for coordinating the switch and the switching methods defined by the user. The cost of model switching therefore is the cumulative time required by the STEMSS resolution switching function, and the time required by the user-defined high-to-low-resolution switch and low-to-high-resolution switch. The cumulative time required by the STEMSS switching method is $H_{sw}*N_{sw}$, where $H_{sw}$ is the time required to switch models, excluding the invocation of the user-defined switching method and $N_{sw}$ is the number of switches performed by the simulation. Recall from the discussion about computing $T_{wo}$, that $N_{swl} = N_{swh}$ and $N_{swl} + N_{swh} = N_{sw}$. The cumulative time required to switch active models is $N_{swl}T_{rstl} + N_{swh}T_{rsth}$, where $N_{swh}$ and $N_{swl}$ are as previously defined and $T_{rsth}$ and $T_{rstl}$ are the inclusive times of a single reset of the IOM and aggregate model respectively. With some algebraic manipulation, the performance equation becomes:

$$
\begin{aligned}
T_{exe} = {} & T_{so} + N_{tr}\left[F_{po} + L_a T_{swl} + H_a T_{swh} + \sum_{i \in \{\text{trans}\}} R_{tr,i} T_{tr,i}\right] + \\
& N_{sw}\left[\frac{1}{2}(T_{sswl} + T_{sswh} + T_{rstl} + T_{rsth}) + H_{sw}\right]
\end{aligned} \tag{30}
$$

$T_{so}$ is the overhead cost of executing the simulation loop, which includes event insertion and removal. The inclusive cost of executing the simulation loop includes the event processing costs and the exclusive cost ignores the list manipulation costs, therefore, $T_{so}$ cannot be directly measured. Instead, $T_{so}$ is estimated as the difference between the inclusive cost of executing the event loop minus the cost of event processing. In this case, event processing includes both model and switching event processing, so $T_{so}$ is:

$$T_{so} = T_{sim} - \left(T_{po} + T_{wo} + \sum_{i \in \{\text{trans}\}} T_{tr,i} + T_{sw}\right) \tag{31}$$

where $T_{sim}$ is the total cost of executing the event loop, including the invocation of transition methods. The cost of executing the event loop per iteration is, $F_{so} = T_{so}/N_{tr}$

By estimating the time parameters, in terms of time per event or time per switch, these parameters are model independent, in the sense that the values are not determined by the execution of a specific model. This gives the user some latitude in measuring the parameters. For example, the user could create a set of deterministic tests that provide precise measurements of each parameter. In this study, profile data from several simulations using the logistic MRM are used to estimate these values.

Unlike the timing parameters, the frequency parameters depend on the duration of the duration of the simulation and the specific model. The frequency parameters can be restated as $N_{tr} = T{*}n_{tr}$ and $N_{sw} = T{*}n_{sw}$, where $T$ is the simulation duration, $n_{tr}$ is the number of transitions per unit of simulation time and $n_{sw}$ is the number of switches per unit of transition time. The restated parameters lead to a general equation for a STEMSS-MRM:

$$\begin{aligned} T_{exe} = T\Big\{ & n_{tr}\Big[F_{so} + F_{po} + L_a T_{swl} + H_a T_{swh} + \sum_{i \in \{\text{trans}\}} R_{tr,i} T_{tr,i}\Big] + \\ & n_{sw}\Big[\frac{1}{2}(T_{sswl} + T_{sswh} + T_{rstl} + T_{rsth}) + H_{sw}\Big]\Big\} \end{aligned} \tag{32}$$

To use the general performance equation (Equation 32) for a specific model, the user must identify each parameter. The parameters, $F_{so}$, $F_{po}$, $T_{sswh}$, $T_{sswl}$ and $H_{sw}$ depend on STEMSS only while the parameters $n_{tr}$, $n_{sw}$, $H_a$, $L_a$, $R_{tr,i}$, $T_{tr,i}$, $T_{swh}$, $T_{swl}$ and $H_{sw}$ are model dependent.

To determine the cost of model processing the specific transition functions must be identified. In the logistic MRM, for example, the model transitions are reproduction, IOM growth and aggregate model growth. The reproduction functions are associated exclusively with the IOM, so the number of reproduction events depends on IOM activity. That is, the cost of the reproduction events is $N_{tr}\mathrm{H_a}R_{repro}T_{repro}$, where $T_{repro}$ is the total cost of executing the reproduction method and $R_{repro}$ is the number of reproduction events per IOM event. The cost of model execution, for the logistic MRM, is:

$$H_a R_{repro} T_{repro} + L_a T_{iom} + H_a T_{agg} \tag{33}$$

where $T_{iom}$ is the cost of executing a single IOM growth event and $T_{agg}$ is the cost of executing a single aggregate model growth event.

Profiling information is used to derived values for the performance equation parameters. The information supplied by the profile data includes the number of calls made to specific methods, the cost of executing each function, excluding the cost of subroutine calls (exclusive time) and the cost of executing each function, including the cost of subroutine calls (inclusive time). The profile information does not map to the parameters directly, so Table 28 is provided to show the association between the performance equation parameters and simulation method calls.

Table 28: Association of STEMSS Methods and Performance Equation Parameters

| Function | number of calls | inclusive time | exclusive time |
|---|---|---|---|
| main | | Texe, program execution time and is used to validate the performance equation | ---- |
| simulate | --- | Tsim, simulation loop execution time, used to estimate Fso | ---- |
| Model::transition | Nm, number of model transitions and used in the estimation of Tpo | --- | Hm, overhead cost of model transition call and is used in the estimation of Tpo |
| MRM::transition | Ntr, number of MRM transitions | --- | Hmrm, overhead cost of MRM transition |
| Network::transition | Nn, number of network transitions and used in the estimation of Tpo | | Hn, overhead cost network transitions call estimation of Tpo |

### 5.3.1 Procedures

The logistic MRM was recompiled with TAU and executed with $U = 100, 1000, 2500, 5000$ and $\Delta = 1.0, 2.5$ and threshold values, $th = 0.23, 0.235$ and $0.24$ and $R = 3.0$. The simulations were run 30 times for each parameter set and switching threshold combination. The performance data generated by TAU was collected and organized according to $\Delta$, $U$ and $th$. The data includes the number of calls to a given method, the cumulative time required to execute the method, including the function calls (inclusive time), and the time to execute a method excluding function calls (exclusive time).

The average value of each performance equation parameter, for each model parameter set, was computed and tabulated. These values were used in the performance equation, which was used to estimate the average execution time of simulations. To validate the equation, and to estimate its accuracy, the error between the estimate average execution time and measured execution time, relative to the measured execution time was computed for each parameter set. The errors were plotted as a function of $U$ and are used to estimate the error in the performance equation.

To demonstrate how a user can predict model accuracy and execution time, the logistic MRM was executed with an additional set of simulations. The new set of equations used $U = 3500$, $\Delta = 1.0, 2.5$ and $th = 0.23, 0.235$ and $0.24$. The performance equation was used to predict the execution time of these simulations and the error versus switching threshold values was graphed. This demonstration shows how a modeler can use STEMSS in a crisis when there are constraints on the execution time.

## 5.3.2 Results

### 5.3.2.1 Performance Equation Parameters

The average value for each parameter in the performance equation will be presented as tabular and graphical form. Each graph plots a parameter value as a family of curves with constant switching threshold values plotted against the upper population bound. Recall that this analysis is for the logistic MRM, other models may require different methods for presenting the data.

As the estimation of parameters relies on TAU, the overhead for TAU was estimated by comparing the execution of model with and without TAU instrumentation. The measured overhead was between 0.1% and 2%. These results suggest that bias in parameter values, caused by TAU, is not significant.

#### 5.3.2.1.1 $N_{tr}$ and $N_{sw}$

Table 29 shows the average number of model transitions and model switches per simulation for parameter set and switching threshold. Figures 50 and 51 show the graphs of $N_{tr}$ with $\Delta = 1.0$ and 2.5, respectively. Both graphs show a roughly linear increase in the number of events as $U$ increases, so the modeler can estimate the number of model transitions by interpolation with this graph.

Figures 52 and 53 are graphs of the average number of switching events, for $\Delta =$ 1.0 and 2.5 respectively. The curves decline as a function of U when $U > 1000$, though it is unclear whether the decline is asymptotic or the model eventually ceases to switch. For values of $U$ less than 1000, the behavior of the curves appears to be uncorrelated. This may be due population extinctions for smaller values of $U$ but the matter needs to be investigated. Though the specific form of the curves can only be learned by further analysis, a modeler can use these curves to estimate the number of transitions and switchesin a simulation. As the impact of switching, on the logistic model is small, a linear interpolation for $U$ greater than 1000 should be sufficiently accurate. ,

Table 29: $N_{tr}$ and $N_{sw}$ values

| | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| *th* | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $N_{tr}$ | | | | | | |
| 100 | 8182 | 8182 | 11300 | 7356 | 7356 | 10910 |
| 1000 | 72680 | 90900 | 106600 | 64830 | 75160 | 89440 |
| 2500 | 188600 | 248000 | 272900 | 168700 | 205800 | 241000 |
| 5000 | 392800 | 514000 | 548200 | 354500 | 431000 | 515400 |
| $N_{sw}$ | | | | | | |
| 100 | 264 | 265 | 5.6 | 322 | 322 | 15. |
| 1000 | 326 | 169 | 37 | 385 | 295 | 173. |
| 2500 | 304 | 99 | 15 | 366 | 237 | 117. |
| 5000 | 278 | 69 | 11 | 339 | 206 | 64 |

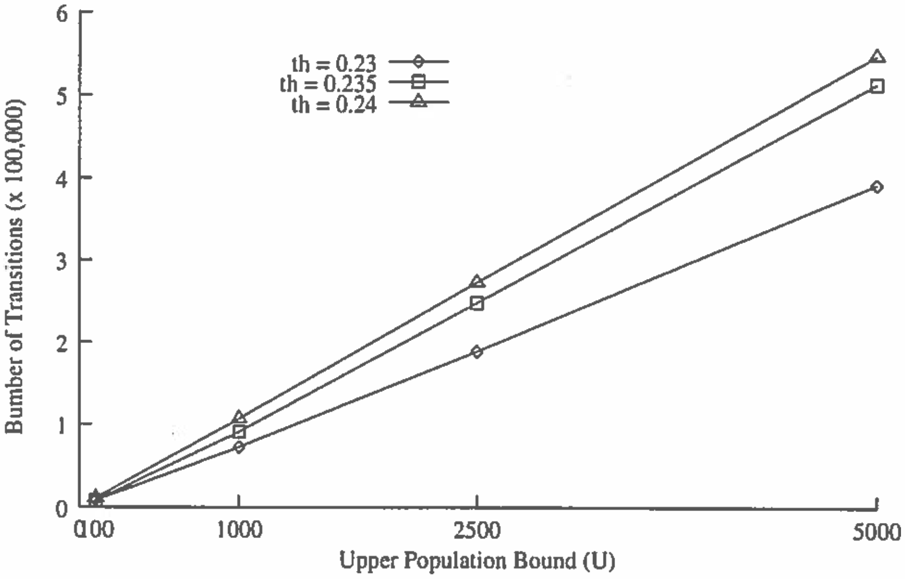

Figure 50. $n_{tr}$ versus $U$ for $\Delta = 1.0$

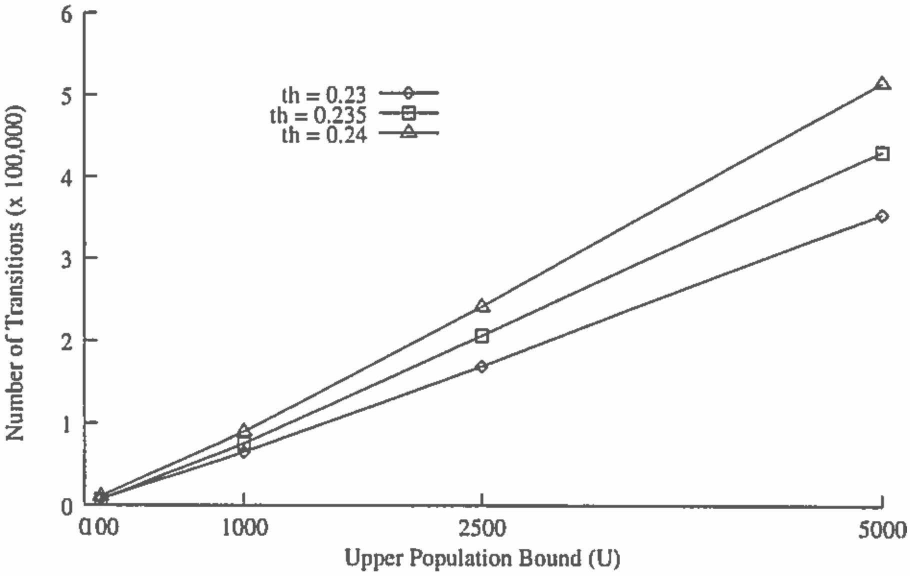

Figure 51. $n_{tr}$ versus $U$ for $\Delta = 2.5$

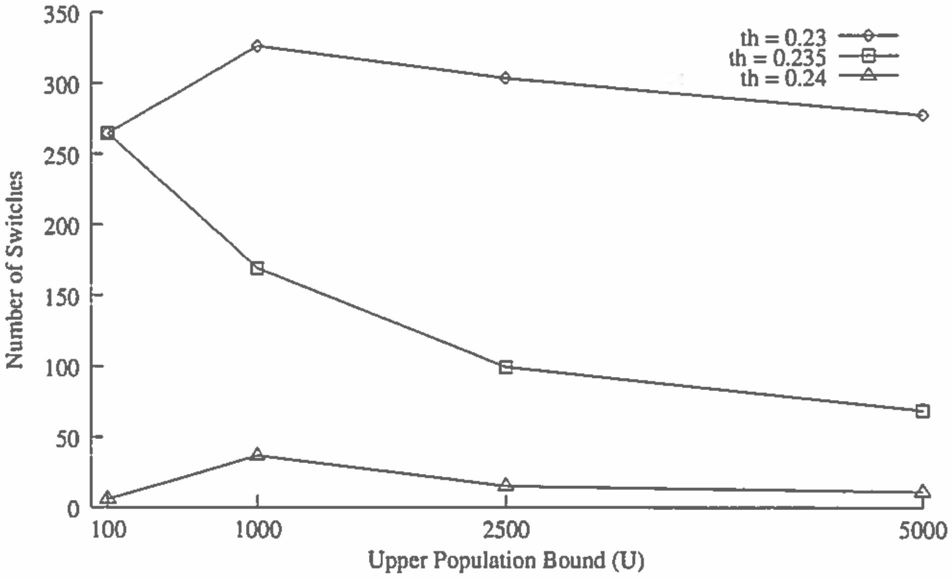

Figure 52. $n_{sw}$ versus $U$ for $\Delta = 1.0$

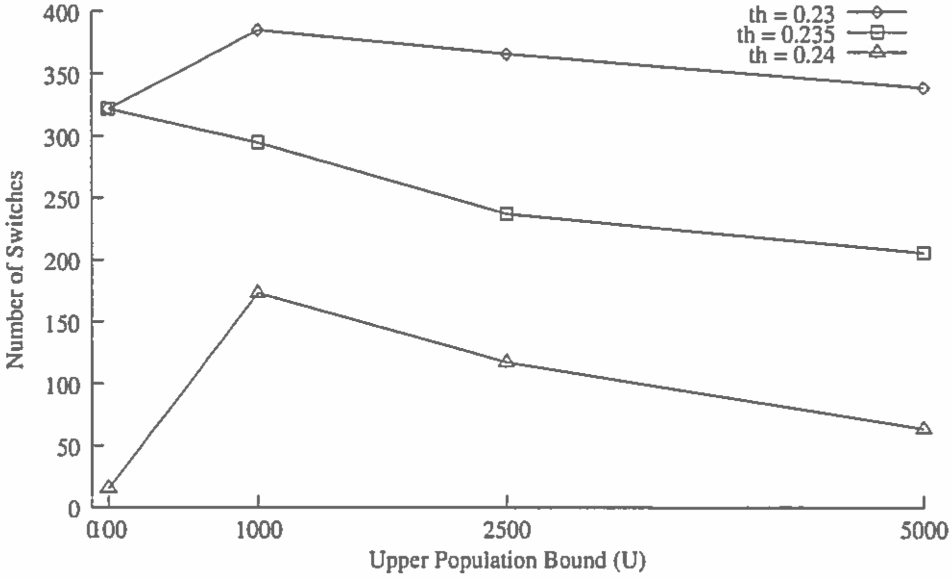

Figure 53. $n_{sw}$ versus $U$ for $\Delta = 2.5$

### 5.3.2.1.2 $N_{agg}$, $N_{iom}$, $H_a$ and $L_a$

The average number of aggregate model and IOM growth events are presented in Table 30. This data is use to compute IOM activity ($H_a$) and aggregate model activity ($L_a$)

Table 30: $N_{agg}$ and $N_{iom}$

| | $\Delta = 1.0$ | | | $\Delta = 2.5$ | | |
|---|---|---|---|---|---|---|
| th | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| Niom | | | | | | |
| 100 | 369 | 369 | 498 | 340 | 340 | 493 |
| 1000 | 338 | 416 | 483 | 308 | 353 | 414 |
| 2500 | 349 | 451 | 493 | 318 | 382 | 442 |
| 5000 | 362 | 467 | 496 | 331 | 398 | 469 |
| Nagg | | | | | | |
| 100 | 132 | 132 | 2.8 | 161 | 161 | 7.67 |
| 1000 | 163 | 85 | 18 | 192. | 148 | 87 |
| 2500 | 152 | 50 | 7.6 | 183 | 119 | 59 |
| 5000 | 139 | 34 | 5.5 | 169 | 103 | 32 |

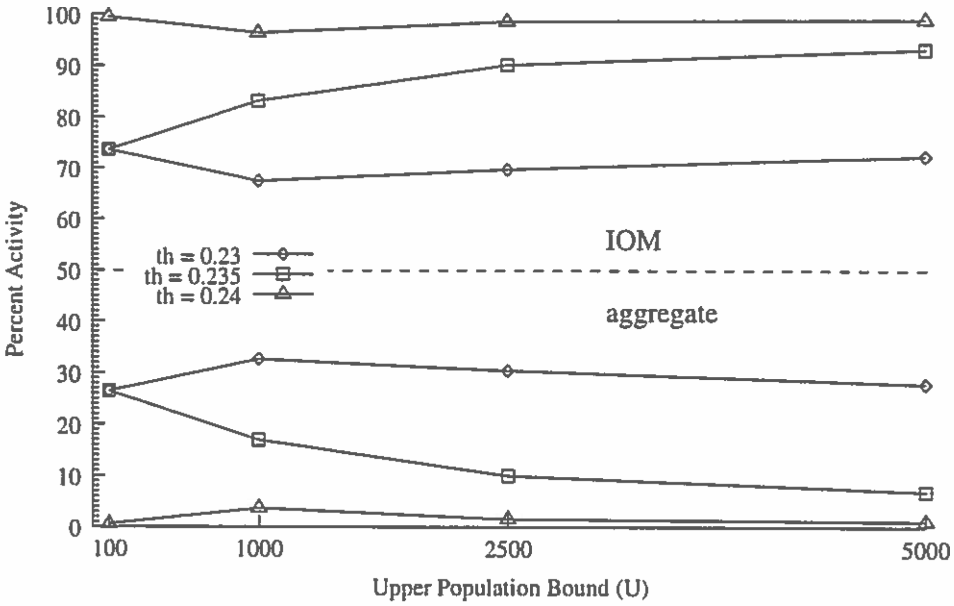

Figure 54. IOM and Aggregate Model activity versus $U$, with $\Delta = 1.0$.

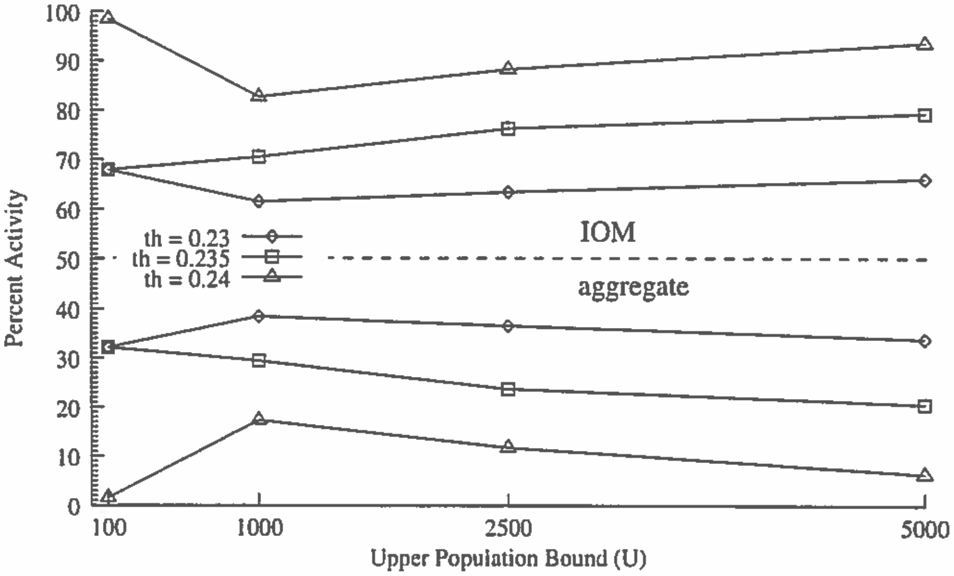

Figure 55. IOM and Aggregate Model activity versus $U$, with $\Delta = 2.5$.

and the resulting plots are shown in Figures 54 and 55, respectively. The curves for both $H_a$ and $L_a$ appear to be either linear or asymptotic. In either case, a modeler can use these graphs as a basis for estimating $H_a$ and $L_a$.

#### 5.3.2.1.3 $T_{po}$

The overhead cost of model transition processing is the cost of invoking the transition functions. In STEMSS, the invocation of a transition function requires the identification of the appropriate transition function and the invocation of that function. This cost is the exclusive cost of executing the model and network transitions, $H_m$ and $H_n$,

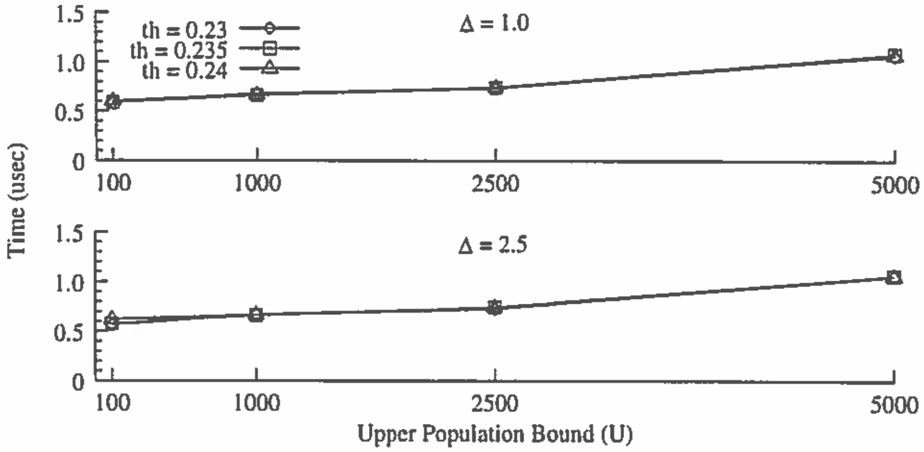

Figure 56. $F_{po}$ versus $U$, with $\Delta = 1.0$ on bottom and $\Delta = 2.5$ on top.

times the number of model and network transitions, $N_m$ and $N_n$. Figure 56 shows the graphs of processing overhead per event as a function of $U$ for $\Delta = 1.0$ and 2.5.

### 5.3.2.1.4 $T_{wo}$

The overhead cost of switching transitions is the cost of detecting and setting up a model switch. These values are determined by the number model transition events, the number of switches and the time to process each of the related methods. Table 31 presents the times associated with the switching parameters, $T_{swh}$, $T_{swl}$, $T_{sswh}$ and $T_{sswl}$ and Figures 57-60 show the graphs of $T_{swh}$, $T_{swl}$, $T_{sswh}$ and $T_{sswl}$ versus U. Each figure shows the graphs for the $\Delta = 1.0$ and 2.5. The graphs show an increase in each parameter value as $U$ increases, and the parameter values for each $th$ are generally consistent, i.e.,

$th = 0.24$ produces the highest results for $T_{swh}$ and the lowest for $T_{swl}$ and $T_{sswl}$. The trend for $T_{sswh}$ is unclear. The significance for each parameter is small, so linear interpolation is used to approximate the values of these parameters.

Table 31: Switch Detection Parameters

| | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| th | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $T_{swl}$ | | | | | | |
| 100 | 0.0011 | 0.0011 | 0.00025 | 0.0014 | 0.0014 | 0.00028 |
| 1000 | 0.0020 | 0.0011 | 0.00040 | 0.0024 | 0.0018 | 0.0011 |
| 2500 | 0.0029 | 0.0014 | 0.00088 | 0.0032 | 0.0025 | 0.0017 |
| 5000 | 0.0027 | 0.0012 | 0.0007 | 0.0030 | 0.0023 | 0.0011 |
| $T_{swh}$ | | | | | | |
| 100 | 0.0017 | 0.0017 | 0.0038 | 0.0017 | 0.0017 | 0.0024 |
| 1000 | 0.0018 | 0.0021 | 0.0024 | 0.0018 | 0.0018 | 0.0019 |
| 2500 | 0.0024 | 0.0025 | 0.0041 | 0.0021 | 0.0022 | 0.0024 |
| 5000 | 0.003 | 0.0034 | 0.0056 | 0.0030 | 0.0031 | 0.0034 |
| $T_{sswl}$ | | | | | | |
| 100 | 0.0013 | 0.0013 | 0.0014 | 0.0013 | 0.0013 | 0.0013 |
| 1000 | 0.0019 | 0.0019 | 0.0019 | 0.0019 | 0.0018 | 0.0018 |
| 2500 | 0.0029 | 0.0029 | 0.0028 | 0.0027 | 0.0028 | 0.0028 |
| 5000 | 0.0034 | 0.0035 | 0.0034 | 0.0033 | 0.0033 | 0.0034 |
| $T_{sswh}$ | | | | | | |
| 100 | 0.00027 | 0.00027 | 0.0003 | 0.00027 | 0.00027 | 0.00028 |
| 1000 | 0.00027 | 0.00027 | 0.00028 | 0.00027 | 0.00027 | 0.00028 |
| 2500 | 0.00030 | 0.00028 | 0.00028 | 0.00027 | 0.00028 | 0.00028 |
| 5000 | 0.00031 | 0.00030 | 0.00030 | 0.00031 | 0.00034 | 0.00030 |

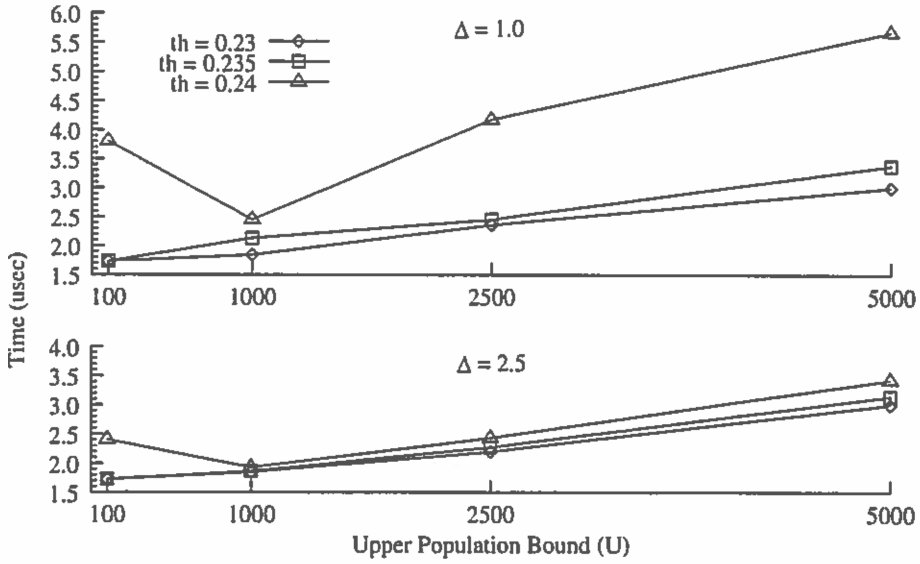

Figure 57. $T_{swh}$ versus $U$, with Δ = 1.0 on bottom and Δ = 2.5 on top.

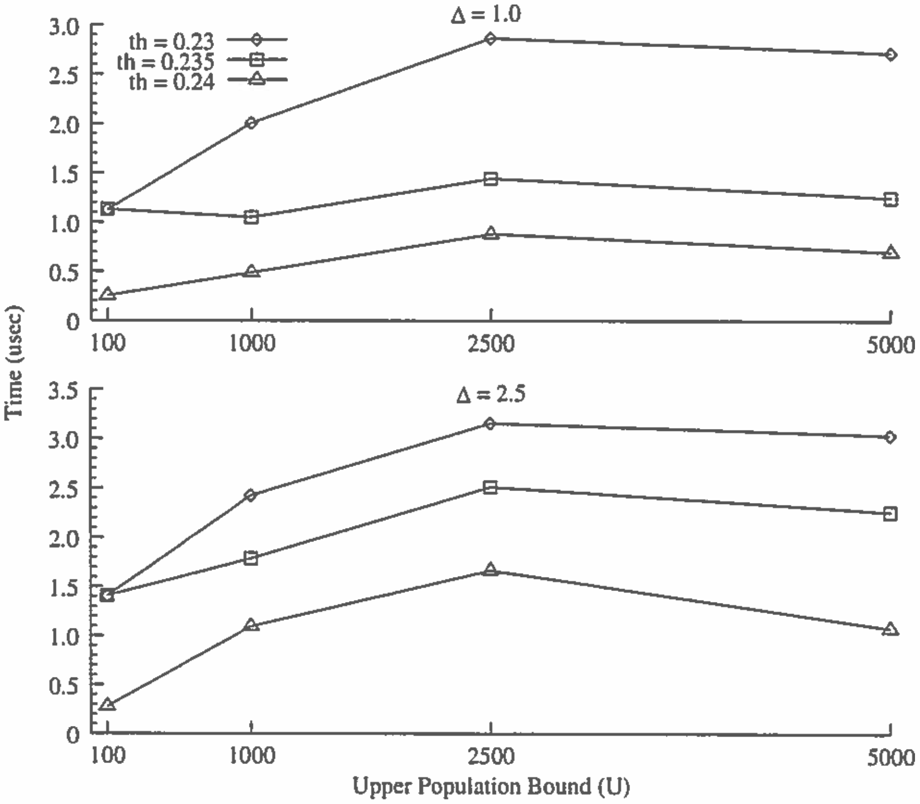

Figure 58. $T_{swl}$ versus $U$, with Δ = 1.0 on bottom and Δ = 2.5 on top.

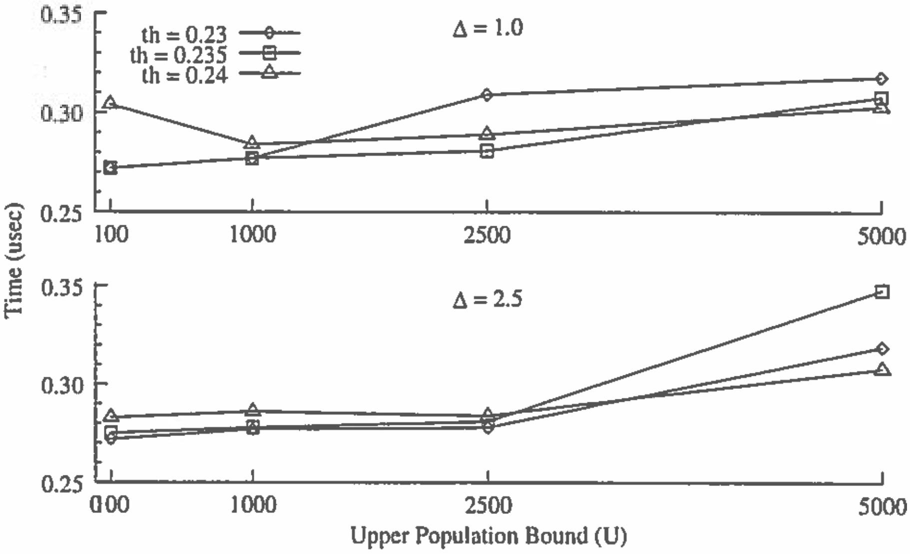

Figure 59. $T_{sswh}$ versus $U$, with $\Delta = 1.0$ on bottom and $\Delta = 2.5$ on top.

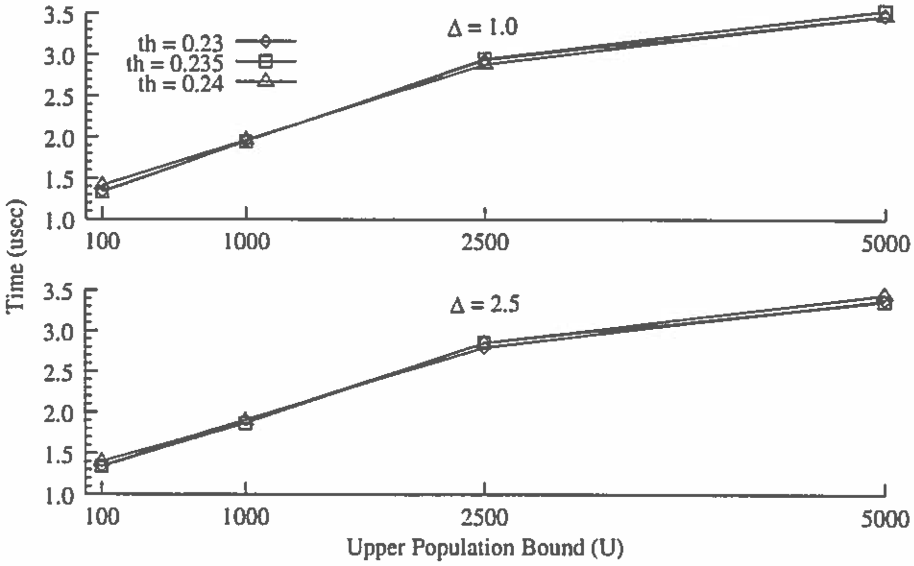

Figure 60. $T_{sswl}$ versus $U$, with $\Delta = 1.0$ on bottom and $\Delta = 2.5$ on top.

### 5.3.2.1.5 $T_{so}$

$T_{so}$ is the difference between the inclusive simulation time and the inclusive cost of processing events (see Table 32). The performance equation requires the use of $F_{so}$, rather

Table 32: STEMSS Overhead Cost, $T_{so}$

| | $\Delta = 1.0$ | | | $\Delta = 2.5$ | | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $T_{so}$ | | | | | | |
| 100 | 285.97 | 276.67 | 408.57 | 250.98 | 240.50 | 362.39 |
| 1000 | 2231.65 | 2803.78 | 3554.04 | 2020.31 | 2289.91 | 2688.64 |
| 2500 | 5887.88 | 7831.91 | 9114.29 | 5515.77 | 6673.17 | 7513.57 |
| 5000 | 12090.40 | 15665.32 | 18665.63 | 12726.83 | 15105.16 | 17171.34 |

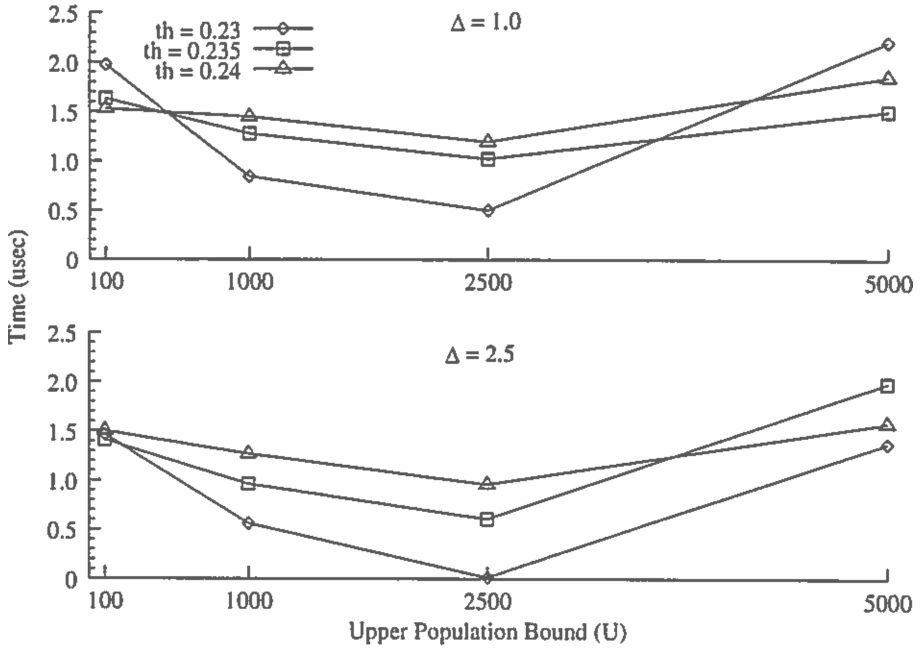

Figure 61. $F_{so}$ versus $U$, with $\Delta = 1.0$ on bottom and $\Delta = 2.5$ on top.

Table 33: Logistic Reproduction Parameters

| | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $N_{repro}$ | | | | | | |
| 100 | 7681 | 7681 | 10800 | 6855 | 6855 | 10410 |
| 1000 | 72180 | 90400 | 106100 | 64330 | 74660 | 88940 |
| 2500 | 188100 | 247500 | 272400 | 168200 | 205300 | 240500 |
| 5000 | 392200 | 513500 | 547700 | 354000 | 430500 | 514800 |
| $T_{repro}$ | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| 100 | 0.0062 | 0.0062 | 0.0063 | 0.006 | 0.0060 | 0.0062 |
| 1000 | 0.013 | 0.013 | 0.013 | 0.012 | 0.012 | 0.012 |
| 2500 | 0.038 | 0.041 | 0.042 | 0.038 | 0.040 | 0.043 |
| 5000 | 0.21 | 0.22 | 0.22 | 0.20 | 0.21 | 0.22 |

than $T_{so}$, so Figures 61 show the graphs of $F_{so}$ versus $U$, for $\Delta = 1.0$ and 2.5, respectively. The curves in both graphs show an apparent minimum, which makes estimating the parameter value difficult. Nevertheless, as the impact of $F_{so}$ is small, the value is estimated using linear interpolation.

#### 5.3.2.1.6 Logistic IOM Parameters

##### 5.3.2.1.6.1 Reproduction Events

The time required to process reproduction events is the number of reproduction events times the time required to process a single event. The data related to the cost of processing reproduction events is presented in Table 33. The performance equation uses the ratio of reproduction events to IOM events, $R_{repro}$, rather than $N_{repro}$. The graphs of

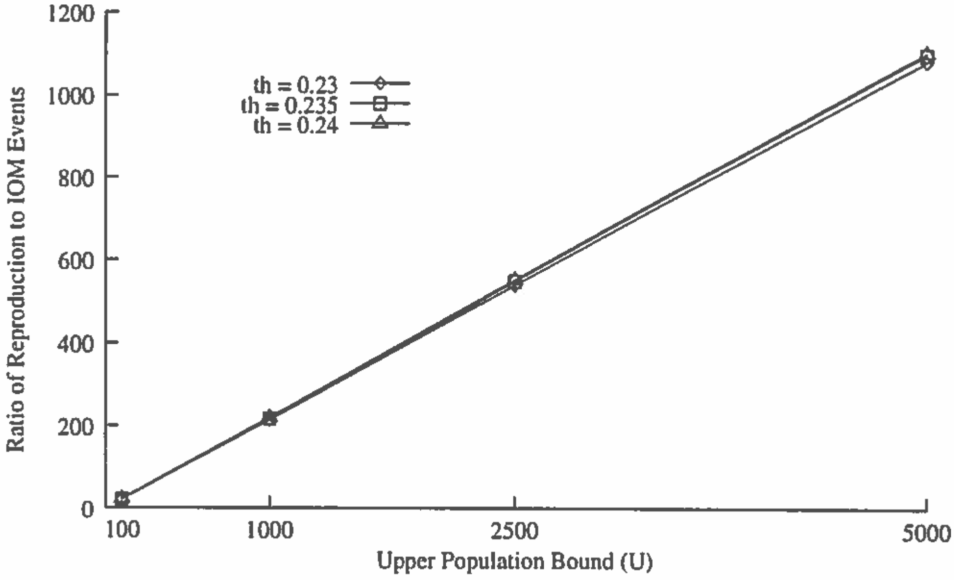

Figure 62. $R_{repro}$ versus $U$, with $\Delta = 1.0$

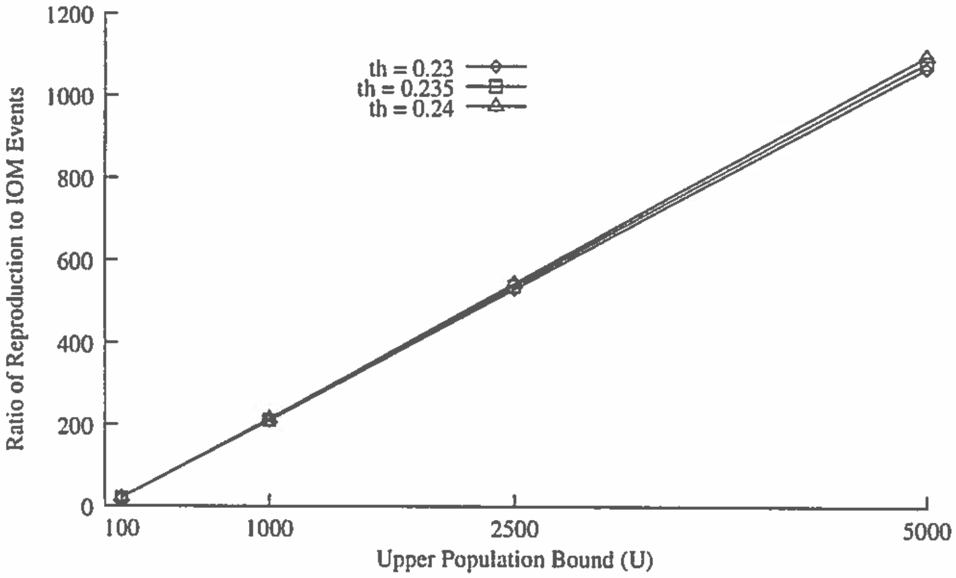

Figure 63. $R_{repro}$ versus $U$, with $\Delta = 2.5$

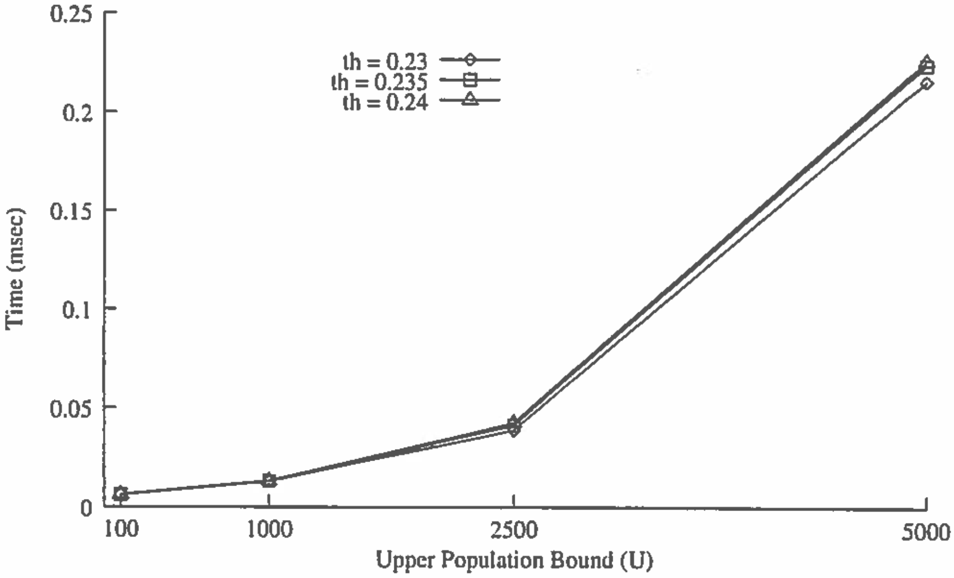

Figure 64. $T_{repro}$ versus $U$, with $\Delta = 1.0$

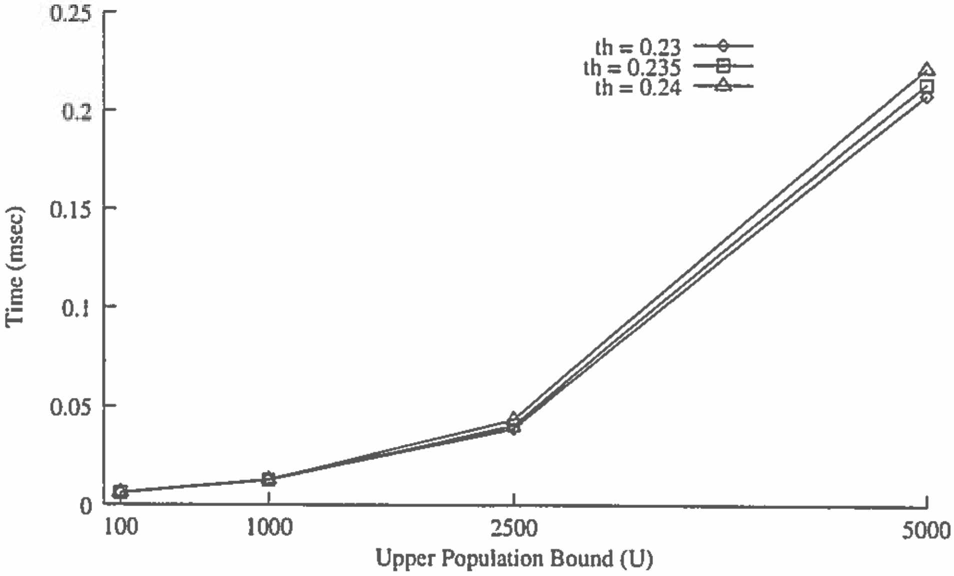

Figure 65. $T_{repro}$ versus $U$, with $\Delta = 2.5$

$R_{repro}$ for $\Delta = 1.0$ and 2.5 are shown in Figures 62 and 63, respectively. Both graphs appear to be linear, so a linear interpolation should be sufficient for estimating the parameter values. The graphs of $T_{repro}$ are shown in Figures 64 and 65, for $\Delta = 1.0$ and 2.5, respectively. These graphs show a nonlinear trend in the time required to process events and the importance of the measurement requires a visual approximation of the parameter's value.

#### 5.3.2.1.6.2 Aggregate Growth Events

The time required to execute the aggregate model's growth method is the product of the time to execute a single invocation of the method, $T_{agg}$, and the number of times the method is called, $N_{agg}$. The data for $N_{agg}$ are shown in Table 30. Table 34 presents the average execution time for a single invocation of the method. These data suggest a constant cost for invoking the aggregate model's growth method. To estimate this parameter, linear interpolation should be sufficient.

Table 34: Aggregate Logistic Growth Parameters

| | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $T_{agg}$ | | | | | | |
| 100 | 0.00029 | 0.00029 | 0.00029 | 0.00029 | 0.00029 | 0.00029 |
| 1000 | 0.00030 | 0.00030 | 0.00030 | 0.00030 | 0.00037 | 0.00030 |
| 2500 | 0.00037 | 0.00036 | 0.00033 | 0.00038 | 0.00037 | 0.00038 |
| 5000 | 0.00040 | 0.00041 | 0.00039 | 0.00042 | 0.00040 | 0.000405 |

### 5.3.2.1.6.3 IOM Growth Events

The time required to execute the IOMs growth method is the product of the time to execute a single invocation of the method, $T_{iom}$, and the number of times the method is called, $N_{iom}$. The data for $N_{iom}$ are shown in Table 30. Average execution time for each parameter set is shown in Table 35 and in Figures 66 and 67. As with $T_{repro}$, $T_{iom}$ has a significant impact on the performance equation, so a linear interpolation is not sufficient. Instead, the equation is approximated visually.

Table 35: Logistic IOM Growth Parameters

| | | $\Delta = 1.0$ | | $\Delta = 2.5$ | | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $T_{iom}$ | | | | | | |
| 100 | 0.095 | 0.096 | 0.11 | 0.088 | 0.088 | 0.10 |
| 1000 | 3.68 | 3.83 | 4.034 | 3.43 | 3.56 | 3.77 |
| 2500 | 39.52 | 43.20 | 45.35 | 39.15 | 42.00 | 46.06 |
| 5000 | 531.91 | 561.60 | 569.56 | 504.50 | 526.04 | 554.47 |

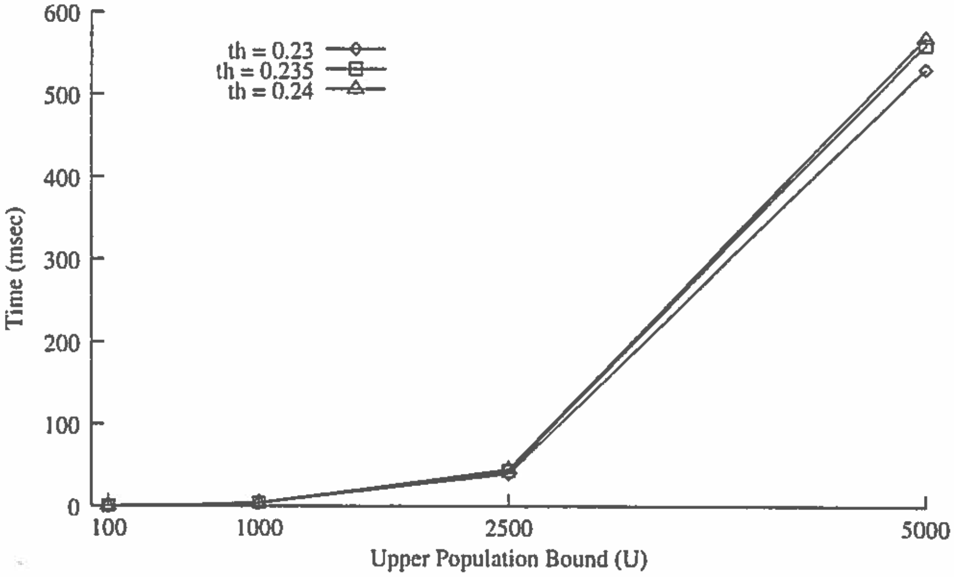

Figure 66. $T_{iom}$ versus $U$, with $\Delta = 1.0$

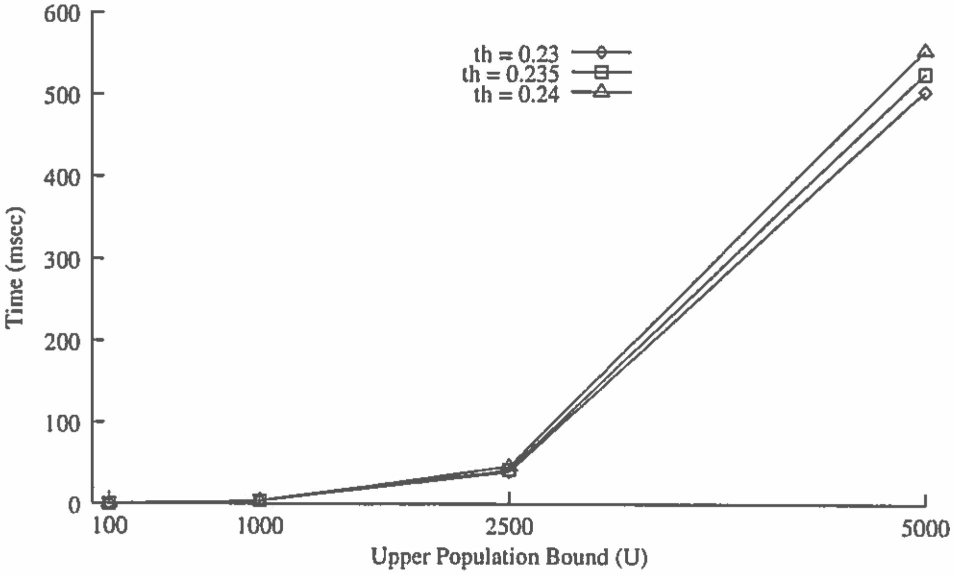

Figure 67. $T_{iom}$ versus $U$, with $\Delta = 2.5$

#### 5.3.2.1.7 Switching Cost

The cost of switching the active model is divided between the cost of the STEMSS resolution switching method, $H_{sw}$, and the user-defined methods for performing the switches, $T_{rstl}$ and $T_{rsth}$. The performance data for $H_{sw}$ is presented in Table 36 and the data for $T_{rstl}$ and $T_{rsth}$ are shown in Table 37. Figure 68 shows the graphs of $H_{sw}$ versus $U$,

Table 36: $H_{sw}$

| | | | $\Delta = 1.0$ | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $H_{sw}$ | | | | | | |
| 100 | 0.002 | 0.002 | 0.003 | 0.002 | 0.002 | 0.002 |
| 1000 | 0.0025 | 0.0025 | 0.0025 | 0.0025 | 0.0025 | 0.0025 |
| 2500 | 0.0029 | 0.0029 | 0.0027 | 0.0027 | 0.0030 | 0.0031 |
| 5000 | 0.0033 | 0.0039 | 0.0030 | 0.0029 | 0.0036 | 0.0036 |

Table 37: Switching Times

| | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
|---|---|---|---|---|---|---|
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| $T_{rstl}$ | | | | | | |
| 100 | 0.032 | 0.032 | 0.030 | 0.030 | 0.030 | 0.032 |
| 1000 | 0.46 | 0.45 | 0.39 | 0.43 | 0.43 | 0.43 |
| 2500 | 1.20 | 1.16 | 0.86 | 1.12 | 1.12 | 1.12 |
| 5000 | 2.46 | 2.38 | 1.68 | 2.30 | 2.29 | 2.23 |
| $T_{rsth}$ | | $\Delta = 1.0$ | | | $\Delta = 2.5$ | |
| | 0.23 | 0.235 | 0.24 | 0.23 | 0.235 | 0.24 |
| 100 | 0.034 | 0.033 | 0.032 | 0.033 | 0.0329 | 0.032 |
| 1000 | 0.516 | 0.494 | 0.415 | 0.486 | 0.473 | 0.454 |
| 2500 | 1.36 | 1.28 | 0.982 | 1.263 | 1.238 | 1.191 |

for $\Delta = 1.0$ and 2.5, respectively. The trends in these graphs are unclear, however, the impact of $H_{sw}$ is small, so a linear interpolation is used to approximate this parameter.

The graphs of the $T_{rstl}$, for $\Delta = 1.0$ and 2.5 are shown in Figures 69 and 70 respectively and the plot of $T_{rsth}$, for $\Delta = 1.0$ and 2.5 are shown in Figures 71 and 72. respectively. The graphs for $T_{rstl}$ and $T_{rsth}$ are apparently linear, so a linear interpolation is used to approximate these parameters.

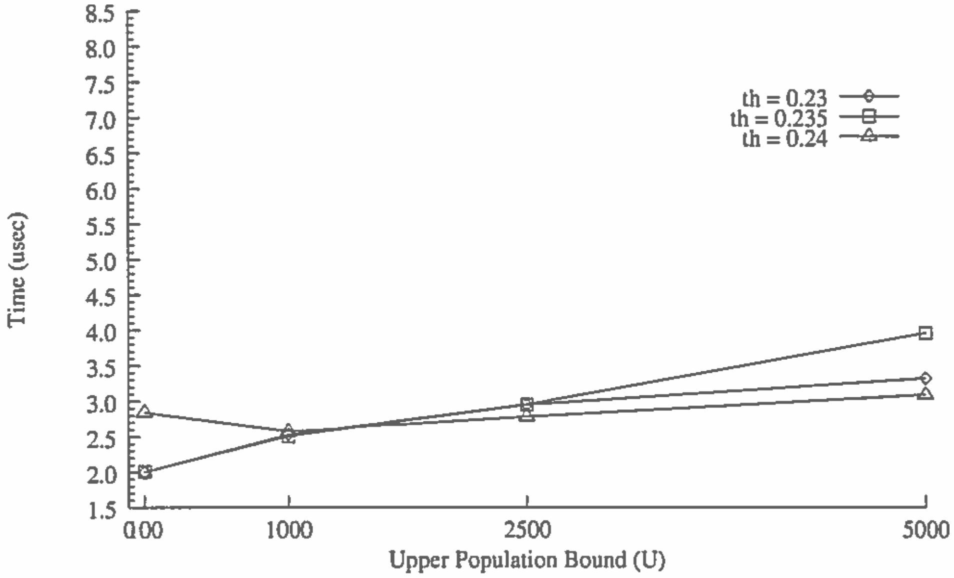

Figure 68. $H_{sw}$ versus $U$, with $\Delta = 1.0$

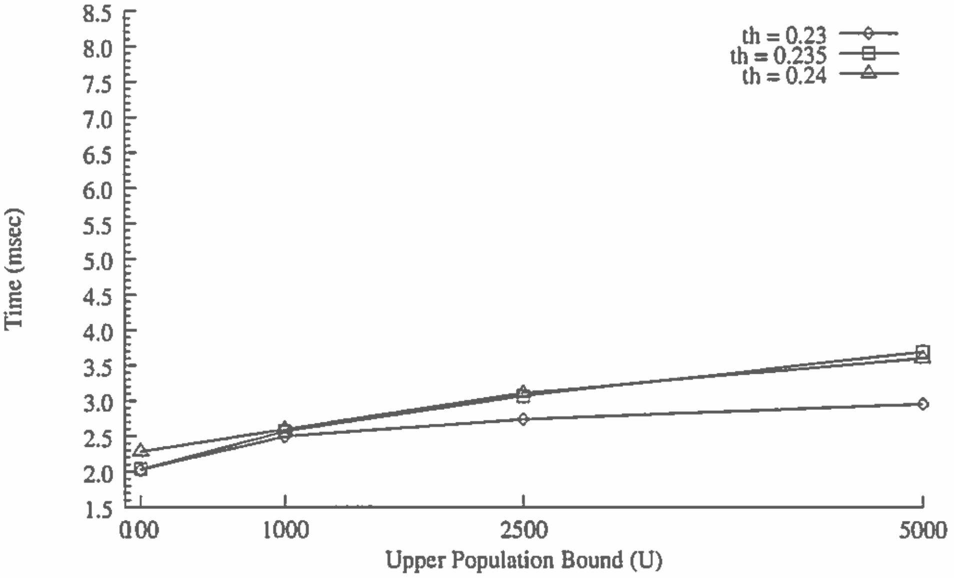

Figure 69. $H_{sw}$ versus $U$, with $\Delta = 2.5$

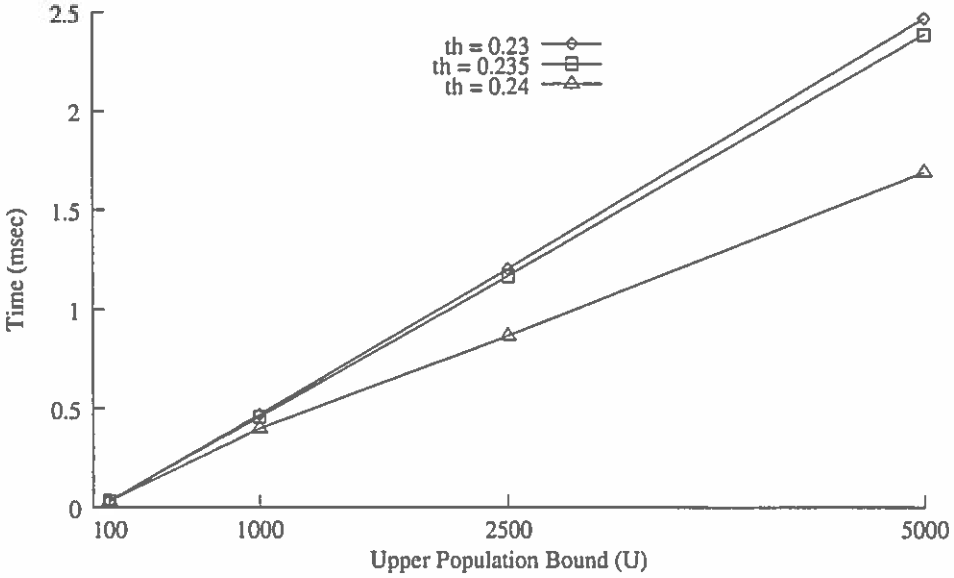

Figure 70. $T_{rstl}$ versus $U$, with $\Delta = 1.0$

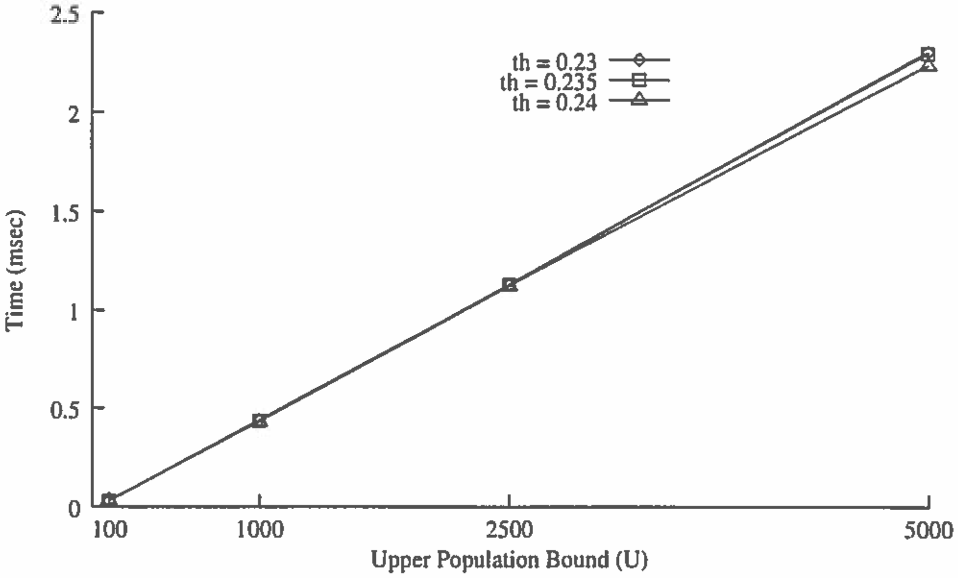

Figure 71. $T_{rstl}$ versus $U$, with $\Delta = 2.5$

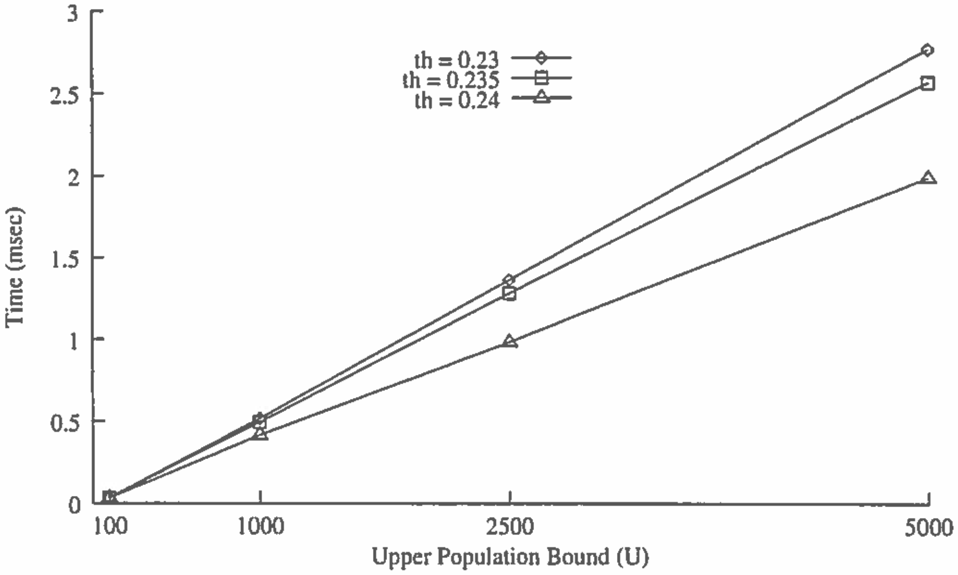

Figure 72. $T_{rsth}$ versus $U$, with $\Delta = 2.5$

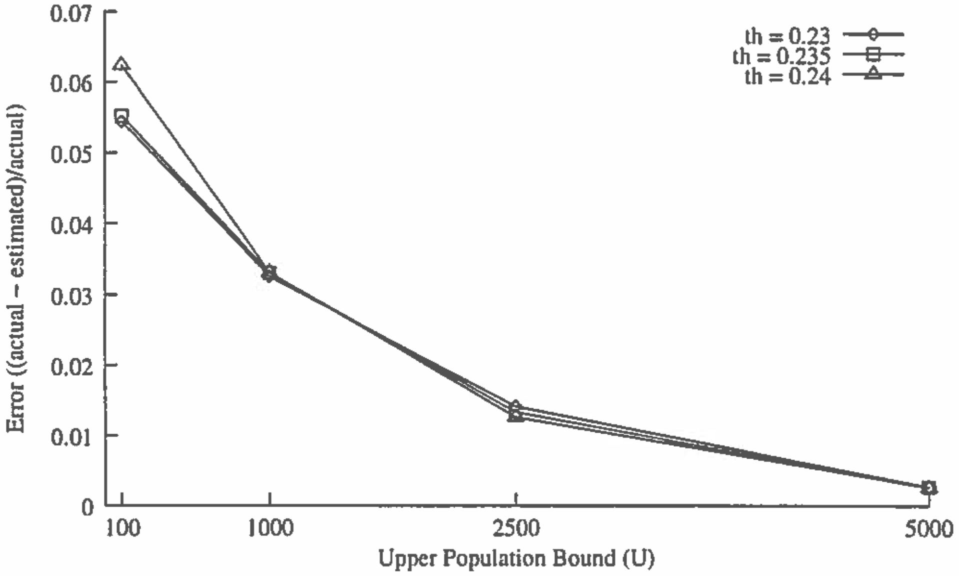

Figure 73. Percent error in Predicted Performance versus Actual Performance, with $\Delta = 1.0$

### 5.3.2.2 Validation of Performance Equation

The performance equation was used to estimate the execution time based on the simulation parameters used to produce the performance equation parameters. The relative error between the predicted and actual time, calculated as $(T_{exe,pred} - T_{exe})/T_{exe}$, was plotted versus $U$ and computed for each for each value of $T_{exe,pred}$. The plots for $\Delta = 1.0$ and 2.5 are shown in Figures 73 and 74 respectively. The error between the predicted and the actual value does not exceed 5%, is not substantial. Therefore, the relative error between the predicted and actual time is not significant.

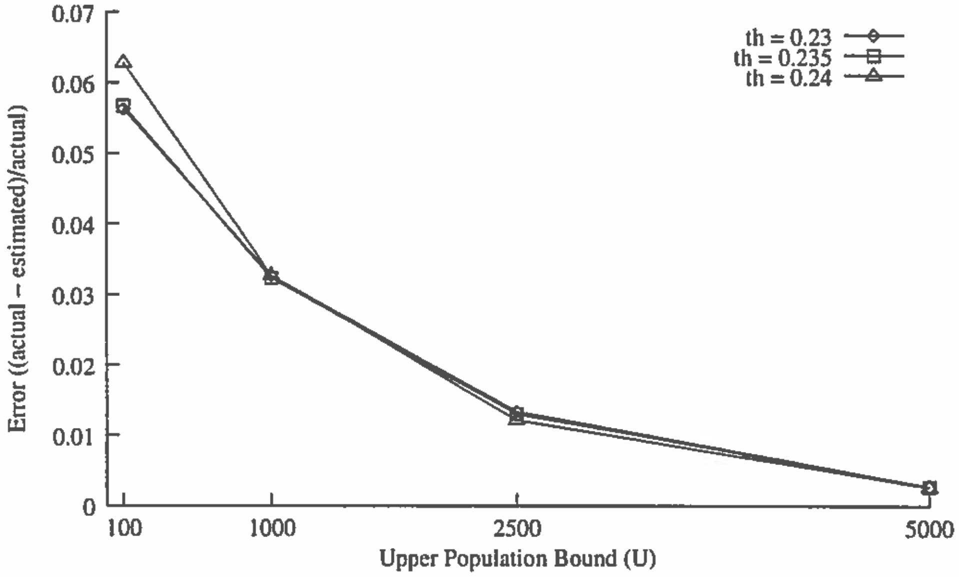

Figure 74. Percent error in Predicted Performance versus Actual Performance, with Δ = 2.5

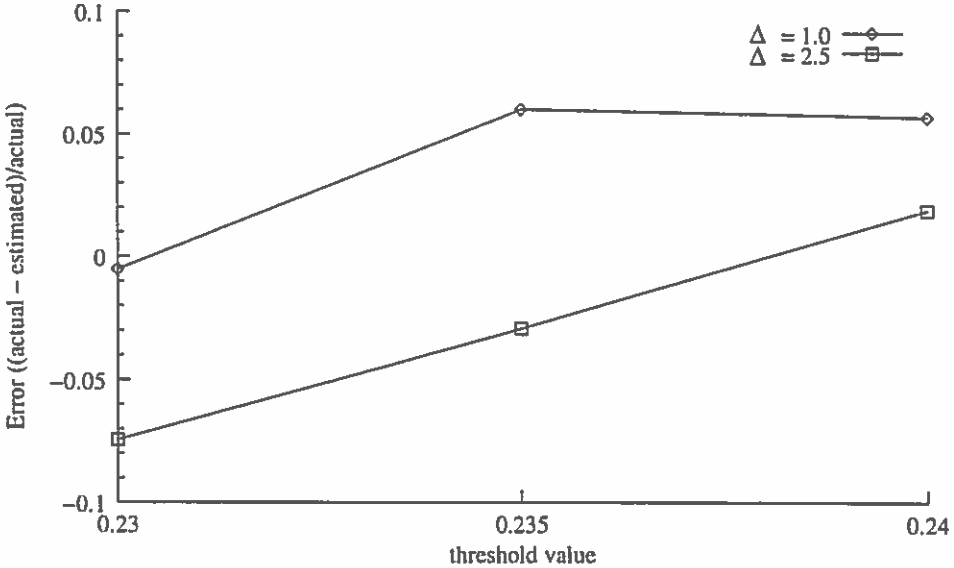

Figure 75. Percent error in Predicted Performance versus Actual Performance, $U = 3500$

To validate the equation, an additional set of simulations was run using $U = 3500$, $\Delta = 1.0$ and 2.5 and $th = 0.23$, 0.235 and 0.24. The performance equation, using parameter values from graphs describe above, was used to estimate the runtime for each of these simulations. Figures 75 shows the relative error between the predicted and actual runtimes. The error in the predicted execution time for the program is less than 10%. As the IOM activity increases, the prediction becomes more accurate.

### 5.3.3 Discussion

This experiment validates the general MRM performance equation (equation 9) and describes how a model can predict both accuracy and execution time for an arbitrary simulation. Figures 73 and 74 show that, for the cases tested, the equation maintains a prediction error of less than 5%. If a modeler were required to produce a simulation, within a specified time limit, using the logistic MRM, he or she can select a switching threshold that would accomplish the task. He or she can estimate the accuracy of the model, relative to the IOM version of the model. The analysis provided in this experiment is for the logistic model, which is easy to analyze. To produce a similar analysis of another model say the yellow perch model, would be time consuming and the prediction accuracy might not be as good. This is an inherent problem in using an MRM. The ability to predict performance of any model depends on the nature of that model and, currently, it depends on a thorough profile of that model. For an IOM with a large parameter space, it is unlikely that the user will produce a thorough profile, so the user must develop some intuition regarding behavior of the model.

Another important result of the performance equation is that it may aid in the development of model switching policies. The current policy of setting a constant threshold value was developed for demonstration purposes. The performance equation, in conjunction with the profile data, describes the factors that influence the simulation's execution time, which can aid in identifying potential performance bottlenecks. The ability to identify and avoid these bottlenecks could improve and refine the ability of a modeler to control the simulation's execution time.

Consider the logistic MRM. The current switching policy waits for the invocation of the annual growth method before testing for a switch. However, stochastic effects can produce legitimate switching conditions before the annual step. The performance equation can estimate the cost of switching between annual growth updates which can form the basis of a new switching policy. Moreover, the equation can be incorporated into the STEMSS framework to make these estimates during realtime. The potential uses of the performance equation need further analysis.

## 5.4 General Discussion

Perhaps the most important result in this chapter is the evidence that STEMSS-MRM can improve execution speed. In particular, the graphs of execution time versus upper population bound (see Figures 47 and 48) show that, with an appropriate selection of the switching threshold, the user can improve execution time. This result reaffirms an underlying aspect of the central hypothesis of this dissertation.

The chapter also points out possible avenues for further research. In particular, the use of the performance equation in identifying bottlenecks is an important research direction. To use the equation effectively, the relationship between performance factors and model accuracy must be carefully analyzed. The analysis should lead to the development of heuristics for setting model switching policies.

In addition to characterizing the behavior of specific models, the incorporation of the performance equation into the STEMSS framework is an important research goal. By incorporating the performance equation into the framework, the modeler may be able to set switching policies based on a set of performance and accuracy goals rather than setting them based on an analysis of profile data. Moreover, the equation may be able to monitor the simulation's performance in realtime giving greater flexibility in controlling and possibly automating the simulation's switching behavior. Further research in the characteristics of MRM behavior must be conducted before these techniques can be implemented.

# CHAPTER VI

# CONCLUSIONS AND FUTURE WORK

## 6.1 A Review of the Hypothesis and Contributions

The central hypothesis of this dissertation is that multiple resolution modeling will allow modelers to represent populations at more than one level of resolution while keeping the results bounded by, and arbitrarily close to the behavior of the constituent single-resolution models. Multiple-resolution modeling should also improve the execution of the model relative to the execution time of the corresponding high-resolution model.

To validate this hypothesis in a population modeling context, it was necessary to develop a set of population MRMs. However, a review of the ecological modeling and computer science literature revealed two problems with this approach. First, the ecological modeling literature suggests that the MRM should be developed in the context of a software framework for molding and simulation, but no such framework exists. Second, the development of MRMs relies on a general structure for multiple-resolution modeling that extends the framework, but no such general structure exists. The task of this dissertation was the creation of a well-defined, general framework for population modeling and multiple-resolution modeling component that extends the framework.

The creation of the framework depended on the requirements of ecologists. To identify these requirements, it was necessary to understand how ecologists might use the framework. Therefore, the first step in the research presented in this dissertation was the identification of the requirements for developing the system. The second step was the identification of the requirements for a general multiple-resolution modeling system. The first two steps led to the development of STEMSS, which provides a framework for multiple-resolution population models. The final step in this dissertation was the validation of MRMs developed with STEMSS. The results of the validation study confirmed the central hypothesis for a specific set of models.

To identify the specific requirements for creating the multiple-resolution modeling framework and to validate the central hypothesis, Chapter I identified four research questions whose answers would direct the development and testing of the system. These questions were:

- How do ecologists use computer simulation?
- How does a model's resolution affect its accuracy and execution speed? What factors affect the selection of a particular resolution? What problems may arise in representing a population at two levels of resolution?
- What are the issues related to computer science in developing MRMs and using them in a simulation? In particular, how can consistency between constituent models, in an MRM, be ensured?
- How well do multiple-resolution population models perform?

### 6.1.1 Ecological Modeling and Computer Simulation

Until recently, ecologists have relied on mathematical techniques, such as differential equations and finite difference methods to model population dynamics. With the arrival of inexpensive computing, ecologists have extended their modeling repertoire to include system theoretic techniques such as individual-oriented modeling. These techniques provide significant benefits. For example, IOMs are assumed to be more accurate than mathematical models as the latter are based on average-case behavior and thereby lose information. On the other hand, IOMs are computationally expensive and are treated with suspicion because they implemented as large and generally *ad hoc* computer programs. This discussion lead to the following set of requirements for a population modeling framework:

- the framework must be based on a well-defined and generally-accepted modeling and simulation theory,
- the modeling and simulation system must support a general set of modeling structures that can be specialized to arbitrary model types (e.g., discrete-time, continuous-equation, discrete-event),
- the framework must provide a set of interface classes that refines the modeling structures to population modeling structures,
- the framework must also specify a technique for supporting code development from the model specification.

The purposes of these requirements are twofold. First, the framework must support the development of population models, from an ecological perspective, and second, the framework must support a multiple-resolution modeling extension.

The ecological modeling framework, STEMSS, was designed as a response to satisfy these requirements. STEMSS supports the development of ecological models by supplying two sets of modeling structures. One set is based on a generally accepted theory of modeling and simulation and the other extends the modeling structures to represent commonly used population modeling concepts. STEMSS also supplies a set of procedures for defining models. The procedures explain the model's structure and implementation concisely and completely.

STEMSS is, nominally, a population modeling framework. However, the underlying structure of STEMSS separates the support for ecological modeling from the general modeling and simulation packages. As a result, the ecological modeling support in STEMSS can be replaced with a similar structure from another domain. For example, a computational chemist can replace the population modeling structures with chemical modeling structures. The chemical modeling structures can represent a molecule as a single-entity (model class), as a set components, such as amino acids in a protein, (network class), or as a set of atoms. The STEMSS structure should provide a modeling interface for alternate domains.

### 6.1.2 Development of STEMSS-MRM

The developers of IOMs hoped the technique would eventually become the dominant paradigm for ecological models. However, as the development of individual-oriented modeling techniques advanced, the complexity of the IOMs increased, which increased the computational effort required to execute them. To address the problem with the computational requirements, ecologists have developed alternative representations, including mixed IOM-aggregate model representations. The alternative presented in this dissertation comes from the suggestion by DeAngelis and Rose, in which they describe the use of both models as a representation of a system. This suggestion is also a description of multiple-resolution modeling.

The specific advantage of an MRM, as described in this dissertation, is that the model can produce simulations that are arbitrarily close to the IOM, but can execute more quickly than the IOM. This approach may be especially useful for managing an environmental emergency, such as an oil spill. Under these conditions, those charged with cleaning up the disaster must make timely decisions about the placement of resources. To use STEMSS-MRM in this situation, the modeler must be able to predict, with confidence, the accuracy and execution time the MRM.

The problem with creating an MRM is the lack of a general description of the requirements for multiple-resolution modeling. The existing research into the problem, such as battlefield MRMs developed for the military, has identified fundamental issues for creating valid MRMs. Among these issues are the requirements for state consistency, as described by Davis and Hillestad (1993) and consistency in cross-resolution interactions as described in the multiple-resolution entity (MRE) concept (Reynolds et al., 1997). Other research groups have worked on different consistency problems. The following is a set of five consistency maintenance categories derived from a review of the existing literature:

- the first category of MRMs consists of models with multiple state resolutions,
- the second MRM category consists multiple temporal resolution models,
- the third category of MRM, those that impose behavioral consistency,
- the fourth category of multiple-resolution models consists of the models that impose interaction consistency,
- the fifth category of multiple-resolution models consists of models that impose the organizational consistency.

The description of multiple-resolution modeling presented in this dissertation used a well-defined theory of modeling, in conjunction with the consistency categories, to develop a set of structures for MRM. To maintain consistency between arbitrary models, the MRM must:

- include a means for translating the state of the constituent models,
- include a means for resolving state differences,
- include a means for resolving differences induced by temporal scale,
- include a means for resolving differences in input values,

- include a means for resolving differences between network and model states.

To incorporate these requirements into the STEMSS framework, three structures were added to STEMSS. The MRM class is used to couple the low-resolution and high-resolution models into a single representation, the Translator class is used to coordinate cross-resolution interactions and the ResolutionController class is used to implement model switches. The MRM class is used to impose state, behavioral and segmentation consistency between constituent models.

### 6.1.2.1 Testing of STEMSS-MRM

Three MRMs were developed to test the implementation of STEMSS-MRM extension to the STEMSS framework. These models include the logistic model, a predator-prey model and aversion of Rose's yellow perch model. The tests were designed to answer four questions. Does the MRM maintain state consistency for these models? Do simulations based on the MRM remain bounded between the high and low resolution models? Do the MRMs improve execution time? Does the MRM decrease the execution time of the model, relative to the IOM.

The first study is used to validate the STEMSS-MRM structures and model development techniques. Before testing the central hypothesis of this dissertation, using models developed with STEMSS-MRM, showing that the framework is correctly implemented is necessary. Therefore, the specific consistency maintenance routines were tested. The results of this test showed that the MRM was correctly implemented.

The second study uses the yellow perch suite to test the efficacy of STEMSS in developing realistic multiple-resolution population models. As the goal of the system is to aid in resolving real problems, the STEMSS system must support the construction of realistic ecological MRMs. The models used in this test were simplified for analysis purposes and for time constraints, but they are structurally consistent with the original models. Therefore, they provide a realistic illustration of a STEMSS-MRM model.

The third study examined the behavior of the MRM under stochastic conditions using the logistic MRM. These tests were used as examples of MRMs that support the central hypothesis of this dissertation. The results of the tests showed that the logistic MRM was bounded by its constituent models. Moreover, the test showed that the MRM's switching threshold could provide the user with a mechanism for controlling the accuracy of the model. This result, in conjunction with the results of the timing studies, suggested that STEMSS-MRM may be a useful tool for crisis management.

The fourth study examined the effect of model switching and model parameters on execution time. These studies use the same set of models and parameters as the accuracy study except that the model was compiled using TAU, which was used to measure the simulation's execution time. The test showed a linear relationship between execution time and switching threshold. This relationship shows that, over a range of values, an MRM can decrease the execution time of a model, relative to the MRMs high-resolution constituent model.

The third and fourth studies confirm the central hypothesis of this dissertation. The simulation produced by the MRM is bounded by its constituent models and the MRM improved the execution of the simulation, relative to the high-resolution model.

## 6.2 Future Work

### 6.2.1 Refinement of STEMSS

The most important piece of future work for the research in this dissertation is the refinement of STEMSS. The current version of STEMSS implements a hybrid model structure and a dynamic network structure. Though these structures can support continuous, discrete or hybrid models, their generality makes them cumbersome to use. Therefore, STEMSS needs to incorporate a wider variety of modeling options (e.g., discrete time and differential equation systems). Also, the current version does not support an abstraction of the Zmap. An appropriate generalization of the Zmap will aid in the development of MRMs.

Despite the modeling interface, producing ecological models with STEMSS is still difficult. This is especially true for MRMs. The ecological modeling package provides general structures common to many population models, but the relationship between an ecologist's view of a population and these structures has not been established. As the success of STEMSS relies on the ability of ecologists to construct complex models based on an ecological understanding of the problem, improving the STEMSS model development interface is important.

Improving the interface should take two directions. First, a thorough study of the way in which ecologists develop models must be performed. Second, the lessons learned in this study should be incorporated into a language, perhaps a visual language.

### 6.2.2 Spatial models

A final direction for the ecological aspects of STEMSS is the development of a spatial modeling component. Spatial modeling is an important direction in ecological modeling and STEMSS will have only limited utility if I do not include a spatial modeling component. Fortunately, the rudimentary spatial structures (grids, cellular automata and coupled-map lattices) have structures similar to the model and network structures.

### 6.2.3 Multiple-Resolution Modeling

The results of the experiments discussed in chapters V and VI imply that further investigation into multiple-resolution modeling is warranted. Some possible topics for further studies include the development and analysis of more complex models; the development and analysis of other model switching policies and techniques; the investigation of boundedness in other time series characteristics; and the impact of resegmentation on model accuracy.

This dissertation is not a proof of the efficacy of multiple resolution modeling. Rather, it supplies empirical evidence that MRMs may have a place in scientific modeling. To establish this point, we need to explore, both empirically and theoretically, the limitations of multiple resolution modeling.

# BIBLIOGRAPHY

Abbot, C. A., Berry, M. W., Comiskey, E. J., Gross, L. J. & Luh, H-K. (1997). Parallel Individual-Based Modeling of Everglades Deer Ecology. *IEEE Computational Science and Engineering*. 4. pp 60-72.

ATLSS (2002a). ATLSS Home Page. Retrieved Sept. 26, 2002. http://atlss.org/.

ATLSS (2002b). ATLSS Coding Introduction. Retrieved Sept. 26, 2002. http://www.tiem.utk.edu/~sylv/HTML/Work/Everglades/Code/index.html.

Barros, F. J. (1996). Dynamic Structure Discrete Event System Specification: A New Formalism for Dynamic Structure Modeling and Simulation. In Charnes, J. M., Morrice, D. J., Brunner, D. T. & Swain, J. J. (Eds.), *Proceedings of the 1995 Winter Simulation Conference, 781-785*. New York: ACM Press.

Barros, F. J. (1997). Modeling Formalisms for Dynamics Structure Systems. *ACM Transactions on Modeling and Computer Simulation* 7 (4), 501-515. New York: Association for Computing Machinery .

Barros, F. J., Zeigler, B. P. & Fishwick, P. A. (1998). Multimodels and Dynamic Structure Models: An Integration of DSDE/DEVS and OOPM. In Medeiros, D. J., Watson, E. F., Carson, J. S., & Manivannan, M. S. (Eds.). *Proceedings of the 1998 Winter Simulation Conference*, 413-419. New York: ACM Press.

Begon, M., Harper, J. L., Townsend, C. R. (1996). *Ecology: Individuals, Populations and Communities* (2nd Ed.). Boston: Blackwell Scientific Publications.

Booch, G. (1994). *Object-Oriented Analysis and Design*. Redwood City, CA: Benjamin Cummings Pub. Co.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley.

Caswell, H. & John A. M. (1992). From the Individual to the Population in Demographic Models. In D. L. DeAngelis & L. J. Gross (eds.), *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems*. (pp. 36-61). New York : Chapman & Hall.

Conallen J. (1999). Modeling Web application architectures with UML. *Communcation of the ACM*, 42(10), 63-70. New York: Association for Computing Machinery .

Costanza, R. & Maxwell, T. (1997). A Language for Modular Spatio-Temporal Simulation. *Ecological Modeling*, 103, 105-113.

Davis, P. K. (1993). An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection. In P. K. Davis & R. Hillestad (eds.), *Proceedings of Conference on Variable-Resolution Modeling*, Santa Monica : RAND Corp.

Davis, P. K. & Hillestad, R. (1993). Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management. In Evans, G. W.,Mollaghasemi, M., Russell, E. C., Biles, W. E. (Eds.), *Proceedings of the 1993 Winter Simulation Conference,* pp. 1003-1012. New York: ACM Press

DeAngelis, D. L., Gross, L. J, Huston, M. A., Wolff, W. F., Fleming, D. M., Comiskey, E. J. & Sylvester, S. M. (1998). Landscape Modeling for Everglades Ecosystem Restoration. *Ecosystems* 1, pp 64-75.

DeAngelis, D. L. & Rose, K. A. (1992). Which Individual-Based Approach is Most Appropriate for a Given Problem? In D. L. DeAngelis & L. J. Gross (eds.), *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems*. (pp. 67-87). New York : Chapman & Hall.

de Vasconcelos, M. J. P., Goncalves, A., Catry, F. X., Paul, J. U. & Barros, F. (2002). A Working Prototype of a Dynamic Geographical Information System. *Intl. J. Geographical Information Science*, 16(1). pp 69-91.

EPA (2003). Responding to Oil Spills. Retrieved Jan. 17, 2003. http://www.epa.gov/ oilspill/response.htm.

Foley, J. D., van Dam, A., Feiner, S. K. & Hughes, J. F. (1990). *Computer Graphics: Principles and Practice.* Reading, MA: Addison-Wesley

Freund, R. J. & Wilson, W. J. (1997). *Statistical Methods*. San Diego, CA: Academic Press.

Gathmann, F. O. (1998). Python as a Discrete Event Simulation Environment. In (Ed.). *Proceedings of the 7th International Python Conference*. Retrieved Feb 11, 2003. http://www. foretec. com/python/workshops/1998-11/proceedings/papers/gathmann/ gathmann.html

Gathmann, F. O. & Williams, D. D. (1998) Inter-site: a New Tool for Simulation of Spatially Realistic Popuation Dynamics. *Ecological Modeling* 113. pp 125-139.

Glass, K. A. (2002). *STEMSS User Guide*. Unpublished manuscript.

Glass, K. A. & Stevens, K. A. (1999) A Discrete Event Framework for Spatial and Hierarchical Ecological Models. In Hamza, M (Ed.), *IASTED International Conference Applied Modelling and Simulation*. (pp. 458-462), ACTA Press, Calgary, AB, Canada.

Haefner, J. W. (1992) Parallel Computers and Individual-Based Models: An Overview. In D. L. DeAngelis & L. J. Gross (Eds.), *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems*. (pp. 67-87). New York : Chapman & Hall.

Hiebler. D. (1995). The Swarm Simulation System and Individual-based Modeling. Retrieved Sept. 26, 2002. http://cam.cornell.edu/~hiebeler/swarm-paper.html

Huston, M., DeAngelis, D. & Post, W. (1988). New Computer Models Unify Ecological Theory. *Bioscience*, 38, pp 682-691.

Hyman J. B., McAninch, J. B. & DeAngelis, D. A. (1991). An Individual-Based Simulation Model of Herbivory in a Heterogeneous Landscape. In M. G. Turner & R. H. Gardner (Eds.), *Quantitative Methods in Landscape Ecology*. (pp. 443-475). New York : Springer-Verlag.

Judson, O. (1994). The Rise of the Individual-Based Model in Ecology. *Trends in Ecology and Evolution*. 9, pp 9-14.

Knuth, D. E. (1998). *The Art of Computer Programming (Vol. 2) Seminumerical Algorithms* (3rd Ed.). Reading, MA: Addison Wesley Longman.

Kobryn, C. (1999). UML 2001: A Standardization Odyssey. *Communcation of the ACM*, 42 (10), pp 29-37.

Leslie, P. H. (1945). On the use of matricies in certain population mathematics. *Biometrika* 33. 183-212. London: Cambridge University Press.

Levin, S. A. (1992). The Problem of Pattern and Scale in Ecology. *Ecology*, 73(6) 1943-1967.

Maxwell, T. (2000). A Meta-Model Approach to Modular Simulation. Retrieved Sept. 26, 2002. http://swan.cbl.umces.edu/SME3/MetaModels.html.

Maxwell, T. & Costanza, R. (1995). Distributed Modular Spatial Ecosystem Modeling. *Intl. J. in Computer Simulation*, pp 247-262.

McGlade, J. M. (1999). Individual-Based Models in Ecology. In J. McGlade (Ed), *Advanced Ecology Theory: Principles and Applications.* (pp. 1-22). London : Blackwell Science.

Murray, J. D. (1993). *Mathematical Biology* (2nd). Berlin: Springer.

Natrajan, A., Reynolds, P. F. & Srinavasan, S. (1997). MRE: A Flexible Approach to Multi-Resolution Modeling. In Ferscha, A., Ayani, R. & Tropper, C (Eds.). *Proceedings of the eleventh workshop on Parallel and distributed simulation*, (pp 156 - 163). New York: Association for Computing Machinery

Oestereich, B. (1999). *Developing Software with UML: Object-Oriented Analysis and Design in Practice.* Harlow, England : Addison-Wesley.

Olinick, M. (1976). *An Introduction to Mathematical Models in the Social and Life Sciences.* Reading, MA: Addison-Wesley Pub. Co.

Palmer, J. B. (1992). Hierarchical and Concurrent Individual-Based Modeling. In D. L. DeAngelis & L. J. Gross (Eds.), *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems.* (pp. 126-164). New York : Chapman & Hall.

Papadimitriou, C. H. & Lewis, H. R. (1981). *Elements of the Theory of Computation.* Englewood Cliffs, NJ: Prentice-Hall

Pascual, M. & Levin, S. A. (1999). From Individuals to Population Densities: Searching for the Intermediate Scale of Nontrivial Determinism. *Ecology*, 80 (7), 2225-2236. Tempe, AZ: Ecological Society of America.

Prähofer, H., Auernig, F. & Reisinger, G. (1993). An Environment for DEVS-Based Multiformalism Simulation in Common Lisp/CLOS. *Discrete Event Dynamic Systems: Theory and Applications.* 3, pages 119-149, Kluwer Academic Publ., Boston.

Reisinger, G. (1995). Simulation of Combined Discrete-Continuous Modular Hierarchical Models in Common Lisp/CLOS. Unpublished masters thesis. Johannes Kepler University, Linz, Austria

Reynolds, Natrajan & Srinavasan (1997). Consistency Maintenance in Multiresolution Simulation. *ACM Transactions on Modeling and Computer Simulation*, 7(3), pp368-392.

Rose, K. A., Christensen, S. W. & DeAngelis, D. L. (1993) Individual-based Modeling of Populations with High Mortality: A New Method Based on Following a Fixed Number of Model Individuals. *Ecological Modelling*, 68, pp 273-292.

Rose, K. A., Cowan, J. H., Houde, E. D. & Coutant, (1993) Individual-Based Modelling of Environmental Quality Effects on Early Life Stages of Fishes: A Case Study Using Striped Bass. *American Fisheries Society Symposium*, 14:125-145.

Rose, K. A., Rutherford, E. S., McDermot, D, Forney, J. L. & Mills, E. L. (1997). An Indivdiual-Based Model Analysis of Walleye and Yellow Perch Population Dynamics in Oneida Lake, New York. unpubished manuscript.

Rose, K. A., Tyler, J. A., SinghDermot, D. & Rutherford, E. S. (1996) Multispecies Modeling of Fish Populations. In Megrey, A. & Moksness, E. (Eds.), *Computers in Fisheries Research*. (pp. 194-222).

Scheffer, M., Baveco, J. M., DeAngelis, D. A., Rose, K. A. & van Nes, E. H. (1995). Super-individuals: A Simple Solution for Modelling Large Populations on an Individual Basis. *Ecological Modelling*, 80. pp 161-170.

SDG. (2003). Swarm Development Group Home Page. Retrieved Feb 19, 2003. http://www.swarm.org.

Shaw, M. & Garlan, D. (1996). *Software Architectures: Perspectives on an Emerging Discipline*. Upper Saddle River, NJ : Prentice-Hall

Sinko, J. W., Streifer, W. (1967). A New Model For Age-Size Structure of a Population. *Ecology*, 48(6). pp. 910-918.

SME (2002). Spatial Modeling Environment Home Page. Retrieved Sept. 26, 2002. http://swan.cbl.umces.edu/SME3/.

SPSS (2003) SPSS [Computer Software and Manual]. Chicago: SPSS Inc.

STELLA (1997) STELLA [Computer Software and Manual]. Hannover, NH: High Performance Systems Inc.

Stroustrup, B. (1997). *The C++ Programming Language*. Boston : Addison-Wesley.

TAU (2002) TAU[Computer Software and Manual]. Retrieved Nov. 11, 2002. http://www.cs.uoregon.edu/research/paracomp/tau/tautools/

Zeigler (1976). *Theory of Modelling and Simulation*. Malabar, FL : Robert E. Krieger Pub. Co.