Computer experiments for epidemiological studies : theory, methodology and models

Guo, Zaiyi

2011

Guo, Z. Y. (2011). Computer experiments for epidemiological studies : theory, methodology and models. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50491>

<https://doi.org/10.32657/10356/50491>

NANYANG TECHNOLOGICAL UNIVERSITY

**Computer Experiments for Epidemiological Studies:
Theory, Methodology and Models**

A thesis submitted to the Nanyang Technological University in fulfillment of the requirement for the
degree of Doctor of Philosophy

by

Guo Zaiyi

School of Computer Engineering

2011

Acknowledgements

I would like to express my greatest gratitude to Dr Tay Joc Cing, who is not only my supervisor, but also a genuine friend who encourages me to walk through the highs and lows along this journey. I sincerely thank him for having patience and confidence in me, giving valuable advices and unselfish sharings in both the research work and personal lives. I would also like to thank my supervisors Prof Stephen John Turner and Dr Huang Shell Ying for their kindness and invaluable support in helping me to complete the research work.

I am fortunate enough to work in the Centre for Computational Intelligence (C²i), where I have met many friends. My special thanks to Dr Ang Kai Keng and Dr Ho Nhu Binh, who showed me what makes a great listener and idea generator. My greatest fortune is that I met my significant other, Dr Huang Haoming, now my husband, in C²i. As a partner in both work and life, he supported me in his unique ways and I cannot be more thankful to all the things he did for me.

My sincere thanks also go to my teammates at ROSS Scientific: Ms Irene Teo, Mr Jimmy Ong, Mr Hardiyanto Wibisono, Mr Chan Wei Boon, Mr Jeff Chong, Mr Zhang Zonghui, and last but not the least, Dr Tay Joc Cing. The application of my research findings in the real work, which is also included in this dissertation, would not be successful without their creative ideas and hard works. I am also grateful to Prof Paul A. Tambyah for giving us the chance to conduct this unprecedented study in the National University Hospital, Singapore.

Lastly, I would like to thank my mother, for her unfailing support and generousness in giving me the freedom to pursue my ideal life.

Table of Contents

ACKNOWLEDGEMENTS.....	I
TABLE OF CONTENTS.....	II
LIST OF FIGURES.....	VI
LIST OF TABLES	IX
LIST OF CODE LISTINGS	X
ABBREVIATIONS.....	XI
ABSTRACT	XII
CHAPTER 1 INTRODUCTION.....	1
1.1. ADVANTAGES AND LIMITATIONS OF COMPUTER EXPERIMENTS	2
1.2. WHAT MAKES A COMPUTER EXPERIMENT?	4
1.2.1. <i>Randomized controlled experimental design</i>	4
1.2.2. <i>Comparison of the experimental workflow</i>	6
1.2.3. <i>Preliminary definitions</i>	10
1.3. NEW CHALLENGES OF COMPUTER EXPERIMENTS	12
1.3.1. <i>Challenge of building a sufficient representation of the real world</i>	12
1.3.2. <i>Challenges of implementing experiments on a null model</i>	16
1.3.3. <i>Caution on the use of control and randomization</i>	18
1.4. MOTIVATIONS AND SCOPE OF THIS WORK	21
1.5. ORGANIZATION OF THE THESIS.....	23
CHAPTER 2 BACKGROUND AND MOTIVATIONS.....	25
2.1. BACKGROUND	26
2.1.1. <i>The human immune system and HIV-1 pathogenesis</i>	26
2.1.2. <i>Nosocomial infections in a hospital ward</i>	31
2.2. REVIEW OF MODELING TECHNIQUES	36
2.2.1. <i>A comparative study on DE models and ABMs</i>	36

2.2.1.1.	Model scope	37
2.2.1.2.	Entity representation	39
2.2.1.3.	Behavior descriptions.....	41
2.2.2.	<i>Choice of modeling techniques for computer experiments</i>	44
2.3.	A PRELIMINARY EXPERIMENT WITH HIV PATHOGENESIS HYPOTHESES	47
2.3.1.	<i>A null model of immune system and HIV infections</i>	48
2.3.2.	<i>Hypothesis verification experiments</i>	52
2.3.3.	<i>Lessons learnt from the feasibility study</i>	55
CHAPTER 3	RESEARCH ROAD MAP	58
CHAPTER 4	MODELING MULTI-TIMESCALE EVENTS	61
4.1.	A SURVEY ON UPDATE SCHEMES.....	62
4.2.	AN ASYNCHRONOUS UPDATE SCHEME FOR MULTI-TIMESCALE SIMULATION	68
4.2.1.	<i>Event structure</i>	68
4.2.2.	<i>Scheduling operations</i>	70
4.2.3.	<i>Implicit and explicit event cancellation</i>	71
4.3.	CASE STUDY 1: PERFORMANCE COMPARISON WITH A B CELL LIFE CYCLE MODEL	72
4.4.	CASE STUDY 2: PATIENT STAYS AND HCW DAILY ROUTINES IN AN SICU WARD	80
4.5.	SUMMARY.....	86
CHAPTER 5	MODELING ENTITIES AND THEIR ENVIRONMENT	88
5.1.	MOTIVATIONS OF HYBRID ENTITY REPRESENTATION	89
5.2.	THE HYBRID MODELING APPROACH	91
5.2.1.	<i>Layered environment modeling</i>	91
5.2.2.	<i>Hybrid representation locally to agents</i>	98
5.3.	CASE STUDY 1: CHEMOTAXIS	102
5.4.	CASE STUDY 2: AIRBORNE TRANSMISSIONS AND FAULTY VENTILATION SYSTEMS.....	110
5.5.	SUMMARY.....	116
CHAPTER 6	MODELING AGENTS FOR EXTENSIBILITY	117
6.1.	AGENT BEHAVIOR NETWORK MODEL	118
6.1.1.	<i>Agent composition</i>	118

6.1.2.	<i>Agent behavior network (ABN)</i>	121
6.1.3.	<i>Model detail verification with ABNs</i>	123
6.2.	ITERATIVE STRATEGY FOR MODEL EXTENSION	123
6.3.	CASE STUDY: ENTITY GRANULARITY EXTENSION IN CHEMOTAXIS MODELS.....	127
6.3.1.	<i>Model I: Saturated Dependency on Gradient</i>	128
6.3.2.	<i>Model II: Receptor Kinetics</i>	129
6.3.3.	<i>Model III: Intracellular Signaling Pathway</i>	131
6.3.4.	<i>Experiment results</i>	133
6.4.	SUMMARY.....	137
CHAPTER 7 COMPUTER EXPERIMENT I: NOSOCOMIAL TRANSMISSIONS		139
7.1.	MODEL SCOPE AND METHODOLOGIES	140
7.2.	NULL MODEL DESCRIPTIONS	141
7.2.1.	<i>The SICU environment</i>	142
7.2.2.	<i>Behavior of healthcare workers, patients, and visitors</i>	143
7.2.3.	<i>Adherence to hand hygiene</i>	144
7.2.4.	<i>Disease natural history and transmission</i>	145
7.2.5.	<i>Admission and periodic screening</i>	147
7.2.6.	<i>Symptomatic colonization and treatment</i>	148
7.3.	EXPERIMENTAL RESULTS.....	149
7.3.1.	<i>Null model simulation results</i>	149
7.3.2.	<i>Full factorial experiment results</i>	151
7.4.	SUMMARY.....	159
CHAPTER 8 COMPUTER EXPERIMENT II: CHEMOTAXIS		161
8.1.	INTUITIVELY ASSUMED CHEMOTAXIS MODELS	162
8.2.	EXPERIMENTAL RESULTS AND DISCUSSIONS	164
8.3.	SUMMARY.....	170
CHAPTER 9 CONCLUSIONS.....		172
9.1.	SUMMARY.....	172
9.2.	SUGGESTIONS OF FUTURE WORK	175

REFERENCES 188

List of Figures

FIGURE 1-1 WORKFLOWS OF REAL WORLD EXPERIMENTS AND COMPUTER EXPERIMENTS.....	7
FIGURE 2-2 CENTRAL ROLE OF TH CELLS IN CELLULAR AND HUMORAL IMMUNE RESPONSES (FIGURE ADOPTED FROM [58])	28
FIGURE 2-3 TYPICAL 3-STAGE DISEASE PROGRESSION OF HIV INFECTED PATIENT (FIGURE ADOPTED FROM [98]).....	29
FIGURE 2-4 AGENT BEHAVIOURAL RULES FOR THE NULL HIV INFECTION MODEL	49
FIGURE 2-5 SCHEMATIC DIAGRAM OF THE NULL MODEL.....	50
FIGURE 2-6 MODELING SPACE WITH A 4x4 GRID STRUCTURE	50
FIGURE 2-7 SIMULATION RESULT OF THE NULL HIV INFECTION MODEL.....	52
FIGURE 2-8 SCHEMATICS OF FOUR HIV PATHOGENESIS HYPOTHESES	53
FIGURE 2-9 REPRESENTATIVE SIMULATION RESULTS OF HYPOTHESIS VERIFICATION.....	54
FIGURE 3-1 RESEARCH ROADMAP: GOALS, BUILDING BLOCKS, ISSUES, AND TASKS	59
FIGURE 4-1 TIME-STEP-DRIVEN UPDATE SCHEME: INFREQUENT UPDATES ARE TIMED BY COUNT DOWN VARIABLES	65
FIGURE 4-2 EVENT-DRIVEN UPDATE SCHEME: CONTINUOUS TIME AND VARIABLE TIME ADVANCES.....	67
FIGURE 4-3 EVENT STRUCTURE FOR EVENT-SCHEDULING UPDATE SCHEME.....	69
FIGURE 4-4 STATE DIAGRAM FOR A SIMPLIFIED B CELL LIFE CYCLE.....	72
FIGURE 4-5 B CELL LIFE CYCLE EVENT SCHEDULING	73
FIGURE 4-6 SCALES OF TIME INTERVAL PARAMETERS (INFORMATION COMPLIED FROM [45, 177, 178])	75
FIGURE 4-7 TEMPORAL POPULATION DYNAMICS (A) B CELL, EVENT-SCHEDULING APPROACH; (B) B CELL, UNIFORM TIME-STEP APPROACH; (C) ANTIBODY, EVENT-SCHEDULING APPROACH; (D) ANTIBODY, UNIFORM TIME-STEP APPROACH	77
FIGURE 4-8 COMPARISON OF TOTAL EXECUTION TIME (IN SECONDS).....	78
FIGURE 4-9 (A) EXECUTION TIME FOR EACH MONITOR INTERVAL; (B) PERCENTAGE OF EXECUTION TIME SAVED FOR EACH SAMPLING INTERVAL	78
FIGURE 4-10 SIMULATION OF PATIENT ACTIVITIES UPON ADMISSION (ARROW INDICATES SIMULATION TIME) .	81
FIGURE 4-11 EVENT SCHEDULING FOR PATIENT ADMISSION AND STAY	81

FIGURE 4-12 EVENT SCHEDULING FOR HCW ACTIVITIES	83
FIGURE 4-13 SIMULATION OF NURSE ACTIVITIES	84
FIGURE 5-1 FOUR-LAYER STRUCTURE OF THE SIMULATED ENVIRONMENT.....	92
FIGURE 5-2 MODELING THE EFFECT OF FLOW ON POPULATION QUANTITIES DISTRIBUTION	96
FIGURE 5-3 SAMPLING OF THE LOCAL MOLECULAR CONCENTRATION AND GRADIENT BY AN AGENT	97
FIGURE 5-4 ILLUSTRATION OF THE UNDER-AGAROSE ASSAY (FIGURE ADOPTED FROM [32])	104
FIGURE 5-5 SAMPLE RESULTS OF THE UNDER-AGAROSE ASSAY (FIGURE ADOPTED FROM [196]).....	105
FIGURE 5-6 VISUALIZATION OF SIMULATED CHEMOTACTIC MOVEMENT	105
FIGURE 5-7 SIMULATION RESULTS AT $T=5000$ WITH DIFFERENT k VALUES	106
FIGURE 5-8 CELL MIGRATION PARAMETERS: (A) $k = 10$ (B) $k = 25$	108
FIGURE 5-9 SNAPSHOTS OF THE CELL MIGRATION PATTERN, CELL HISTOGRAM, ATTRACTANT CONCENTRATION AND GRADIENT AT $t=1000$ AND $T=5000$ WITH $k = 25$	109
FIGURE 5-10 LONG-TERM RESULTS AT $t=20,000$ AND $t=100,000$ WITH $k = 25$	110
FIGURE 5-11 A HYPOTHETICAL WARD ENVIRONMENT.....	111
FIGURE 5-12 THE HYPOTHETICAL VENTILATION SYSTEM IN THE WARD ENVIRONMENT	112
FIGURE 5-13 CONNECTION BETWEEN 2 COMPARTMENTS VIA OUTLETS.....	112
FIGURE 5-14 SIMULATION RESULT – WITHOUT VENTILATION SYSTEM.....	114
FIGURE 5-15 SIMULATION RESULT – WITH WORKING VENTILATION SYSTEM.....	114
FIGURE 5-16 SIMULATION RESULT – WITH FAULTY VENTILATION SYSTEM	115
FIGURE 6-1 ABN REPRESENTATION OF THE CHEMOTACTIC MOVEMENT RULE.....	121
FIGURE 6-2 EXAMPLE OF A 3X2 AGENT BEHAVIOR NETWORK	121
FIGURE 6-3 ILLUSTRATION OF MODEL EXTENSION BY ADDITION.....	125
FIGURE 6-4 ILLUSTRATION OF MODEL EXTENSION BY REPLACEMENT	125
FIGURE 6-5 ITERATIVE STRATEGY FOR MODEL EXTENSION	126
FIGURE 6-6 ABNs FOR THREE MODELS OF CHEMOTAXIS (A) MODEL I: SATURATED DEPENDENCY ON GRADIENT (B) MODEL II: EXPANDED WITH RECEPTOR KINETICS (C) MODEL III: EXPANDED WITH SIMPLE INTRACELLULAR SIGNALING PATHWAY (SPECIFIC TO NEUTROPHIL)	128
FIGURE 6-7 MODEL I SIMULATION SNAPSHOT.....	134
FIGURE 6-8 MODEL I MIGRATION PATTERN MEASUREMENTS	134
FIGURE 6-9 MODEL II SIMULATION SNAPSHOT	135
FIGURE 6-10 MODEL II MIGRATION PATTERN MEASUREMENTS	136

FIGURE 6-11 MODEL III SIMULATION SNAPSHOT	136
FIGURE 6-12 MODEL III MIGRATION PATTERN MEASUREMENTS.....	137
FIGURE 7-1 THE SCALED FLOOR PLAN OF THE SICU WARD AND A PHOTO OF ACTIVITY OUTSIDE ROOM 10 (USED WITH PERMISSION FROM THE NATIONAL UNIVERSITY HOSPITAL)	143
FIGURE 7-2 SIMULATION OF VISITOR ACTIVITIES (ARROW INDICATES SIMULATION TIME).....	144
FIGURE 7-3 ILLUSTRATION OF SIMULATED HAND HYGIENE OPPORTUNITIES DURING A HCW'S VISIT TO A PATIENT	145
FIGURE 7-4. HYPOTHESIZED PROBABILITY DISTRIBUTION OF TOUCH EVENTS PER HCW VISIT TO A PATIENT	145
FIGURE 7-5 DISEASE NATURAL HISTORY FOR PATIENTS, STAFF AND VISITORS IN OUR MODEL	146
FIGURE 7-6 SCREENSHOT OF THE SIMULATION ON DAY 5 (MORNING WARD ROUND).....	150
FIGURE 7-7 PATIENT OCCUPANCY LEVEL FOR A SINGLE SIMULATION RUN OF 690 DAYS (DAY 31 TO DAY 720)	151
FIGURE 7-8 DISEASE SPREAD GRAPH SHOWS TRANSMISSION ROUTES AMONG INDIVIDUALS	154
FIGURE 8-1 TOP-VIEW OF THE CELL PATTERNS AT T=10000: MODEL I ON THE LEFT, (A)-(D): $k_I = 3, 10, 30, 100$; MODEL II ON THE RIGHT (A)-(D): $k_{II} = 0.3, 0.1, 0.03, 0.0$	165
FIGURE 8-2 MIGRATION PARAMETERS FOR MODEL I (LEFT) AND MODEL II (RIGHT)	166
FIGURE 8-3 SNAPSHOTS OF THE CELL MIGRATION PATTERN, CELL HISTOGRAM, ATTRACTANT CONCENTRATION AND GRADIENT AT T=1000, 2000, 3000 FOR MODEL I AND II	166
FIGURE 8-4 SIMULATION RESULTS OF MODEL III (T=5000)	167
FIGURE 8-5 MIGRATION PARAMETERS FOR MODEL III. $k_{III}=300$ (LEFT), $k_{III}=1000$ (RIGHT)	168
FIGURE 8-6 SNAPSHOTS OF THE CELL MIGRATION PATTERN, CELL HISTOGRAM, ATTRACTANT CONCENTRATION AND GRADIENT AT T=40, 60, 80 FOR MODEL III WITH $k_{III} = 1000$	168

List of Tables

TABLE 1-1 FORMAL DESCRIPTION OF THE USE OF CONTROL GROUPS	5
TABLE 2-2 COMPARISON OF EQUATION- AND BEHAVIOR-RULE-BASED INTERACTION DESCRIPTIONS.....	42
TABLE 2-3 EQUATIONS AND SCHEMATIC DIAGRAM OF A SIMPLE EQUATION-BASED MODEL [78]	42
TABLE 2-4 TH CELL AGENT RULES AND SCHEMATIC DIAGRAMS [118]	44
TABLE 2-5 NULL MODEL PARAMETER SETTING	51
TABLE 4-1 CLASSIFICATION OF UPDATE SCHEMES	64
TABLE 4-2 SAMPLE OF MANDATORY ACTIVITY LIST FOR A MORNING SHIFT STAFF NURSE (OBTAINED THROUGH INTERVIEW).....	83
TABLE 5-1 MICROSCOPIC RULES AND MACROSCOPIC EQUATIONS FOR RECEPTOR-LIGAND INTERACTIONS THAT ARE LOCAL TO AN AGENT’S ENVIRONMENT	101
TABLE 5-2 CELL MIGRATION PARAMETERS AT $t=5000$	106
TABLE 7-1 FULL FACTORIAL EXPERIMENTAL DESIGN (3 FACTORS AND 3 TO 4 LEVELS PER FACTOR)	141
TABLE 7-2 KEY MODEL PARAMETERS (PARTIAL LISTING): [V] SINGLE-VALUED [U] UNIFORM DISTRIBUTION [E] EXPONENTIAL DISTRIBUTION [B] BETA DISTRIBUTION – (MIN, LIKELY, MAX).....	141
TABLE 7-3 CHANGE IN PATIENT DISCHARGE COLONIZATION RATE DUE TO 20% INCREASES IN NURSE ADHERENCE RATES	152
TABLE 7-4 CHANGE IN PATIENT DISCHARGE COLONIZATION RATE DUE TO 20% INCREASES IN PHYSICIAN ADHERENCE RATES	153
TABLE 7-5 ANALYSIS OF EFFECT ON DISCHARGE COLONIZATION RATE DUE TO INCREASE STAFFING LEVEL FROM 1-TO-1 TO 1-TO-2	156
TABLE 7-6 ANALYSIS OF EFFECT ON DISCHARGE COLONIZATION RATE DUE TO INCREASE STAFFING LEVEL FROM 1-TO-2 TO 1-TO-3	157
TABLE 8-1 EXTENT OF THE CELL SPREAD: MEASURED IN DEVIATION IN X AND Y DIRECTIONS OF 10% OF THE FRONTAL CELLS.....	170

List of Code Listings

LISTING 2-1 PSEUDO CODE OF THE SIMULATION	51
LISTING 4-1 PSEUDOCODE FOR EVENT HANDLING	70
LISTING 4-2 PSEUDOCODE FOR SCHEDULER OPERATIONS	71
LISTING 5-1 AGENT RULE: RECEPTOR-LIGAND INTERACTION	100
LISTING 5-2 AGENT RULE: RECEPTOR-LIGAND INTERACTION (REWRITTEN FOR PARALLEL UPDATING)	102
LISTING 6-1 AGENT RULES OF MODEL I.....	129
LISTING 6-2 AGENT RULES OF MODEL II	131
LISTING 6-3 AGENT RULES OF MODEL III.....	132

Abbreviations

ABM	Agent-Based Model
ABN	Agent Behavior Network
APC	Antigen-Presenting-Cell
BCR	B Cell Receptor
BR	Behavioral Rule
C	Constant Parameter
CA	Cellular Automata
CI	Confidence Interval
CTL	Cytotoxic T cells (T killer cells), also denoted as TK
DC	Dendritic Cells
EI	Entity Instance
ET	Entity Type
HCW	Healthcare Worker
HH	Hand Hygiene
IL	Interleukin
LOS	Length of stay
MPH	Macrophage
MRSA	Methicillin Resistant Staphylococcus Aureus
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
DE	Differential Equation
SARS	Severe acute respiratory syndrome
SICU	Surgical Intensive Care Unit
SV	State Variable
TCR	T Cell Receptor
TH	T helper cell, also denoted as T _H
TK	Cytotoxic T cells (T killer cells), also denoted as CTL

Abstract

There are many types of computer-based models built for various purposes, for demonstration, exploration, prediction, and data analysis to name a few. In this work, we are interested in one particular type: to conduct epidemiological experiments with computer-based models. This is motivated by its potential in offering a less costly, safer, unquestionably ethical and possibly more powerful alternative to the real world scientific studies. There has been extensive research work on computational models that simulate real-life applications which can be used for exploring what-if scenarios in public health concerns, but they often lack a systematic approach in designing and conducting experiments that mirrors its real-life counterpart – that of a rigorous randomized, controlled experiment lauded as the gold standard by medical practitioners.

In this thesis, we envision the goal of conducting computer experiments for epidemiological applications as new discipline with its own theories, methodologies, and standard procedures. We advocate that the computer experiment should also comprise a null model as the control group, and one or more experimental models as the treatment groups, to mirror the methodology of a real world experimental study design. This is important so as to bridge both engineering and scientific goals of using such models. Through the literature review and a feasibility study of HIV pathogenesis, we identify Agent-Based Models (ABMs) as a more suitable modeling technique for supporting computer experiments, because of their capabilities in modeling highly heterogeneous entities and non-uniform interactions, while remaining highly extensible to support our process of incremental model improvement, although they also present disadvantages when compared to simpler and concise compartmental models prevalent in epidemiological research.

To support the construction of credible and extensible ABMS for the use by computer experiments, we propose a multi-scale agent-based modeling and simulation framework that formally addresses these modeling issues. The framework consists of following building blocks: (1) an *event-scheduling update scheme* that employs variable time advances and implicit / explicit cancellations and resolves the issue of modeling multi-timescale activities. Its update realism and

computational efficiency is demonstrated through the cases studies of B cell life cycles, patient admission and stays, and daily work routine of a staff nurse. (2) A *hybrid entity representation* that resolves the issue of entity relative scales by allowing the coexistence of population-level and individual level entities in one model. We demonstrate with a model of chemotaxis that two entity types (i.e., molecules and cells) whose quantities differ by the order of 10^3 can be efficiently modeled with the proposed hybrid entity representation. (3) A *layered environment model* that resolves the issue of building topologically accurate environment with four functional layers: an Agent Holder Layer, a set of Population Quantity Layer(s), a Connection Layer, and a set of Flow Field Layer(s). The capability of this model is fully demonstrated with a ventilation system case study: two compartments - the open space and the ventilation ducts are modeled, with inter-compartment functions such as pathogen absorption, filtration, and leakage. (4) An *iterative strategy for model extension* based on an *agent behavior network model* (ABN) that addresses the issue of appropriate entity granularity and model extensibility. With ABN representation, the models can be extended incrementally through a combination of addition and replacement operations. The iterative strategy for model extension conducts checks and applies incremental changes for each iteration, until the refined model is sufficiently detailed and accurate to serve as the null model. We demonstrate this approach by three chemotaxis models with gradually added granularities.

Finally, we apply the proposed multi-scale agent-based modeling and simulation framework to two real world public health applications. The first experiment assesses the effect of healthcare workers' adherence rates and staffing levels on the nosocomial transmission of MRSA in a surgical intensive care unit. We construct a null model that represents the existing state of the SICU, and validate it with hospital stakeholders and against the clinical observations, such as patient occupancy level and patient discharge colonization rates. We then apply the experimental manipulations by setting hypothetical values for hand hygiene adherence rates and staffing levels. Experiment results suggest that nurse adherence rate is the most impactful factor (explains at least 80% of the variance) on the patients discharge colonization rates. The effect of staffing level is not consistent across the various experimental settings, and a detailed analysis on the individualized transmission routes (*patient* $\rightarrow$ *staff* $\rightarrow$ *patient*, and *outside ward* $\rightarrow$ *staff* $\rightarrow$ *patient*) suggests a correlation between the adherence drop and low staffing level. If this correlation is low, then the constraint of staffing level is not a clinically significant factor in causing increased nosocomial infections to patients. The second

computer experiment mirrors a thought experiment with four chemotaxis experiment models, each corresponds a competing hypothesis. Each model puts forward an intuitive assumption, or hypothesis, of how chemoattractant concentrations and gradients are quantitatively related to the locomotion of cells. Experiment results show that the model with the details of receptor kinetics produces results that correlate with the reality most closely in terms of the cell movement and spread patterns. For the other three models, the simulated cell movement patterns always concentrate to a small, rounded region, contradicting the laboratory observations where the cells should spread out with time. Following the procedure of a thought experiment, assumptions that lead to contradictory results imply that the original assumption is flawed.

For future work, we wish to extend the relatively less-explored case studies, such as extending the ventilation system model for airborne disease transmissions to include realistic hospital environment, and expanding the HIV pathogenesis hypotheses experiments with realistic cell movement patterns backed by the chemotaxis models. Since a large number of simulation runs and experiment configurations are needed for robust computational results, we also wish to explore the utilization of distributed computational infrastructure to speed up the runs. We are also interested in extending computer experiments to other areas such as ecology, economics, and social sciences.

Chapter 1

Introduction

Experiments play vital roles in scientific research in that they provide empirical evidences to *support* or *disprove* hypotheses and theories [1-4]. In contrary to pure observational studies, experiments typically involve artificial manipulation or regulation of environmental conditions [2]. Some familiar examples from history illustrate the power of the well-designed experiment. Galileo Galilei used the “ball and ramp experiment” [5, 6] to *disprove* the Aristotelian Theory of Motion which thought that heavier objects fell faster than light objects [7, 8]. Louis Pasteur used S-shaped flasks to eliminate possible contamination, and this experiment *disproved* the Theory of Spontaneous Generation, which believed that living organisms like worms, frogs, and salamanders could arise from mud, dust, and unpreserved food [9-11]. James Lind *supported* the hypothesis that citrus fruits could cure scurvy using the first ever clinical trial experiment where scorbutic sailors were divided into 6 groups which differed with each other only in the dietary supplement of acidic quality [12-14]. John B. Watson and Rosalie Rayner’s “Little Albert Experiment”, which was carried out on a 9-month infant named Albert, showed empirical evidence to *support* the theory of classical conditioning [15, 16].

In the examples we have discussed above, experiments are conducted on *real physical subjects*, such as balls, water, gas, electrical apparatus, cell cultures, animals, and human beings. We will refer to these experiments as *real world experiments*. The advances in computational technologies have been facilitating the real world experiments at several levels, from data calculation and analysis, to experiment automation [17]. We are particularly interested in the application of *computer experiment* as a less costly, safer, and potentially more powerful alternative to the real world experiments. Like real world experiments, we would expect computer experiments to be able to generate results that add support or disapprove hypotheses or theories.

For real world experiments, standard practices have been proposed over the years of research for more rigorous and scientific study design. For example, the experiment has to eliminate the confounding

factors, or control their effect by setting up control groups and applying randomization techniques [18-21]. However, to our knowledge, little research has been done to formalize the theories, techniques, or standard practice that a *computer experiment* needs to follow.

The goal of our research is to formalize computer experiments as a discipline with its own theories, methodologies, and standard practices. To clarify the challenges of developing rigorous and credible computer experiments, we research the concept of computer experiments in terms of advantage, limitations, compositions, and challenges, with real world experiments as a reference. We present our findings in this introductory chapter.

1.1. Advantages and limitations of computer experiments

Computer experiments are not implemented on real world objects; instead, they rely on a computational representation, or a model, as the substitution of the real world. With this fact, we would expect computer experiments to be beneficial in the following ways:

1. *Reduced economic and logistical costs.* Computer experiments avoid expenditures that are otherwise needed in the real world experiments, such as machinery, equipment, chemicals, laboratory setups, regulatory controls, in addition to manpower for following through the study design process of preparation, execution, observation, recording, and analysis of the entire project. One cost associated with computer experiments is the computer and its associated software environment, which may be expanded to computational clusters if the demand for computational resources is high. As most software for research these days is available for free, the main infrastructure setup concerns the efficient and distributed use of computing nodes and for data sharing. The other cost for the computer experiments are manpower and logistics; comprising modelers (typically computer scientists) and domain experts who jointly do literature and field research, collect data through sampling, followed by building a simulation model equipped with accurate details and then to analyze, interpret and communicate them. While these costs are still substantial, they are *relatively* less costly when compared to real-life experimental costs that are incurred to answer these same questions or achieve the same results.

2. *Reduced work hazards.* As the computer experiments are performed in a virtual, simulated world, the researchers or experimental subjects are not exposed to real-life hazards such as radiation exposure, machinery failures or susceptibility to diseases and infections. Thus, computer experiments are obviously more safely conducted than their real world counterparts.
3. *Faster results.* Real world experiments have different time spans. Short experiments can be completed within a 45-minute laboratory class in secondary schools, such as verifying Ohm's law. Conversely, some complex experiments can last years to decades, due to the slow progress rate of the phenomena to be observed. For example, Gregor Mendel's experiments on plant hybridization with pea plants, which led him to summarize the Mendel's Laws of Inheritance, lasted 7 years from 1856 to 1863 [22, 23]. It is possible to advance the simulation time faster than the real world time by several orders of magnitude, resulting in reduced waiting time. However, the gain of speed is at the cost of loss in accuracy due to the simplification and abstraction assumptions made in the model. Modelers need to balance between the model details and simulation speed with respect to a specific modeling goal.
4. *Full control over system manipulation.* Not every kind of artificial control or manipulation is achievable with real world experiments. For example, it is not practical to artificially reduce HIV mutation rates in a human patient so as to observe its clinical or subclinical outcome. In addition, real world experiments are faced with numerous confounding effects that make the experimental subject sub ideal. For example, patients may vary in their age, weight, hygiene habits, health conditions, disease history, family members' health conditions, dietary preferences, profession, stress levels, people he/she interacts with, places of commute and so on. For experiments such as clinical trials, experimenters are limited to find similar patients only based on a subset of the aspects that they deem matter, for example, the current health conditions. Many other properties, though they are not selected as relevant factors in the experiment, they are nevertheless present in reality and may become the source of confounding effects. On the other hand, with computer experiments, the properties that are present in the model are fully controllable, such as setting the interaction rates, modeling with certain statistical distributions, or specifying the behavior using if-then rules. Manipulations such as setting the HIV mutation rate to an unrealistically low or high level are achievable with computer experiments (and

ethically so as well).

Despite of these advantages, one should be fully aware of the tradeoffs of a computer experiment that comes from its dependency on a humanly conjured model. As a model is at best an abstraction of reality, the process of abstraction could be biased due to limitation in the modeler's knowledge. The outcome of a computer experiment does not necessarily match the true outcome if that experiment were implemented on the real life subjects. Therefore, computer experiments, in spite of their various advantages, are not meant to replace the real world experiment entirely. Real world experiments are still needed, at the very least to validate the predicted results of the simulated model or justify a null model (more on this later). If based on reasonably realistic computational simulation, models can also suggest real-world experiments that are likely to provide new insights.

1.2. What makes a computer experiment?

By “conducting a computer experiment”, we mean to run one or more computer simulations that closely mimics the observable events of an experiment that would take place in the real world system. For example, although even simple Monte Carlo simulations can mimic the chance events of coin tossing or more complex distributions, it is unclear at all how such simulations should be set up to say, evaluate the effect of nurse hand hygiene adherence rate on bacterial infections. In this Section, we aim to establish what we believe a computer experiment workflow should entail by comparing it to the workflow of real-world experiments. *Computer simulation models* and *computer experiments* are two fundamental concepts in the workflow. In order to describe their structures and relationships, we formally define a few terms which will be used frequently in this thesis.

1.2.1. Randomized controlled experimental design

The key features of a rigorous experiment are its use of control and treatment groups, controlling the confounding factors, and varying only the factors under study [24]. This methodology differentiates scientific experiments from other scientifically unsound methods such as those under the scope of alternative medicine [25]. For a real-life experiment, without the differentiation of control groups and treatment groups, and proper randomization to eliminate confounding factors, it would not be possible to tell whether a difference in the outcome, such as the effectiveness of a treatment, is caused

by the factor under study (e.g., treatment), or some other factors. Similarly, if a computer model simply simulates a real-life or hypothetical scenario, it is at best a demonstrative or predicative case study, not a rigorous experiment for decision making. To understand the rationale of setting up control and treatment groups, we formally express the procedure in Table 1-1.

Table 1-1 Formal description of the use of control groups

<p>Let $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$ be the set of factors that are kept the same across the control and treatment groups, and a_{n+1} be the factor that is only present in the treatment group but is absent from the control group.</p> <p>Define a function E as the experiment process that maps a set of input factors an experiment outcome. Let B_1 and B_2 denote the experiment outcomes of the control group and the treatment group, respectively. Then we have</p> $E(\mathbf{A}) = B_1, \quad E(\mathbf{A}, a_{n+1}) = B_2$ <p>Thus the difference between B_1 and B_2 has to be attributed to the presence of a_{n+1} and the interaction between a_{n+1} and $\mathbf{A}$ in the experiment.</p> <p>More generically, if there are m groups and each differ from each other in the factor a_{n+1}. Let $a_{n+1,i}$ be the setting of factor a_{n+1} for the i^{th} group, and B_i be the experiment outcome of the i^{th} group. Then we have</p> $E(\mathbf{A}, a_{n+1,1}) = B_1,$ $E(\mathbf{A}, a_{n+1,2}) = B_2,$ $\dots$ $E(\mathbf{A}, a_{n+1,i}) = B_i,$ $\dots$ $E(\mathbf{A}, a_{n+1,m}) = B_m,$ <p>The differences between $B_1, B_2, \dots, B_i, \dots, B_m$ have to be attributed to the differences in $a_{n+1,1}, a_{n+1,2}, \dots, a_{n+1,i}, \dots, a_{n+1,m}$ and their interactions with $\mathbf{A}$ in the experiment.</p>
--

Intuitively, the effects of the set of common factors $\mathbf{A}$ (or simply the common $\mathbf{A}$) are cancelled off when the groups are *compared to each other*. This is especially useful for research fields where the confounding factors are numerous and the artificial controls over these factors are impractical.

Ideally, the confounding factors can be considered taken care of if they can be lumped into the common $\mathbf{A}$. For example, for clinical trials where the purpose of the study is to conclude the effectiveness of a treatment, a case-control design [18] is typically:

Patient + no treatment (e.g., placebo) $\rightarrow$ symptoms persist

Patient + treatment (e.g., medicine) $\rightarrow$ symptoms are removed or reduced

With real world experiments, there are practical issues in devising a common $\mathbf{A}$, applying the difference $a_{n+l,i}$, and concluding the significance of the differences among B_i . For example, it is difficult to find two identical patients, or two patient groups that are strictly the same in all aspects in their lives (that is, the so-called common $\mathbf{A}$ is not exactly the same across the groups). It may be unethical to leave the patient untreated and let their disease develop (some $a_{n+l,i}$ may face ethical charges). It is likely that many subjects (e.g., patients) are needed in the experiment in order to conclude the statistical or clinical significance in $a_{n+l,i}$ causing B_i , which can be costly. Research methodologies have discussed extensively how these problems can be solved or reduced in practice, through randomization techniques such as double-blind trials [19-21].

Given the scientific foundation of the randomized controlled experiment, we believe that a rigorous computer experiment should mirror this procedure as a necessary but insufficient condition for producing credible experimental results. Computer experiments are different from real world experiments in many ways by construction. This implies that they face different types of challenges too. One example is that randomization is a big practical issue for real world experiments but is technically trivial with computer experiments (we will discuss this topic further in Section 1.3.3). In the next Section, we discuss the similarities and the differences in the workflows of a computer experiment and its real-life counterpart.

1.2.2. Comparison of the experimental workflow

For the ease of description, we will describe the experiment workflow using the example of *chemotaxis*, the natural phenomenon where the biological cells move in the direction of the concentration gradient of certain molecules (known as chemoattractants) [26-30]. Chemotaxis is an important biological process underlying many biological functions such as the tracking of bacterial infections by neutrophils, the organization of the embryo in metazoan, and sourcing of nutrients by

prokaryotes [30, 31]. This phenomenon has been researched extensively; processes include the design of *in vitro* experiment (wet lab experiment) [32], preparation of materials and established experiment protocols [28, 32, 33], establishment of hypothesis and theories on the cell movement patterns on the population level [26, 27, 30], as well as results on the signaling mechanisms at the cellular levels [29-31, 34].

random walk pattern in a chemoattractant gradient (the prior knowledge) [32], we may intuitively propose a *hypothetical guessing* such as “cell movement speed is a function of or is dependent on the chemoattractant gradient”. More formally, researchers may propose a *theory* or a *model* that advocates complex mechanisms expressed with some mathematical equations or behavioral rules. Here, we use the word theory and model interchangeably since theories are expressed with models and models describe theories. For example, Keller and Segel developed a model whose premise was that the cells responded to fluctuations in estimates made of the concentration of the chemoattractant [26], and this model was expressed using Partial Differential Equations (PDEs) that quantitatively relate the flux of the cells per unit time in the direction of increasing chemoattractant to the spatial distribution of the chemoattractant [26, 35, 36]. The model can be used to make concrete, testable predictions, such as to predict the cell spread pattern after certain time with Keller and Segel’s model.

3. *Design the experiment:* The experiment design aims to reflect the scenario stated in the hypothesis, so that its outcome can be used to support or disprove the hypothesis. With real world chemotaxis experiments such as Under-Agarose assay [28, 32, 33], the cells are initially placed in a roundish cell well within an artificially constructed chemoattractant, so that the cell spread pattern at the end of the experiment can be properly measured and compared to the theoretical prediction. Ideally, the experiment should contain a control group, which is free from experimental manipulations, as a reference for the comparison. The experiment design also has to specify the steps to prepare the materials and how the factors under study are systematically varied (manipulated). Similarly, a computer experiment on chemotaxis needs to specify the setup of the virtual environment (such as the quantities and the initial distribution of the virtual cells and chemotaxis), and the ways to vary the factors under study. As shown in Figure 1-1, the design of the experiment is limited by what are available for manipulation in the system that it relies on. For example, with real world experiments, certain types of cells are used because they can be colored with dyes at the end of the experiment so that their distribution pattern can be visually observed. The design of the computer experiment design is limited by the scope and the level of details of the computational simulation model, a simplified representation of the real world system. We refer to such a model as a *null computational simulation model*, or simply a *null model*, to indicate that it is free from experimental

manipulations. It is analogous to a control group in the experiment terminologies. The fact that a null computational simulation model is itself fully controllable by the modeler implies that the model construction can be revised to have wider scope or greater level of detail to suit the needs of the experiment. Thus, we show an *interdependent* relationship between the design of the experiment and the null computational simulation model for computer experiment in Figure 1-1.

4. *Conduct the experiments:* While real world experiments are carried out on real subjects or entities following a case-control study design, or at best a randomized controlled trial, computer experiments are conducted on a *null computational simulation model*, or *null model*, that serves as an analog of the control group. Before the null model can be used for experiment purposes, its details and simulation results should be verified and validated against its real world counterpart, possibly with the help of the domain experts. Next, to obtain the analog of the treatment group, we implement treatment-specific features on the null model to create a *modified computational simulation model*, which we refer to as an *experiment model*.
5. *Collect the data:* Experimental data can be collected during and at the end of the experiments, using various techniques that are applicable to the experiment settings. For example, a real world chemotaxis experiment may have several sets of identical under-agarose assays running in parallel, each is terminated at a different time point (e.g., after 1 hour, after 2 hours, and after 3 hours) so that the cells can be dyed to show the migration patterns at the specific time points. For computer experiments, they can instrument the simulation and extract data to log files during the simulation, hence recording the states of the system over time.
6. *Analyze and conclude:* After the experimental data are collected, they are analyzed and compared to the predicted outcome of the hypothesis. Intuitively, if they are agreeable, then the experiment adds support to the theory; otherwise, the theory is disproved. As with many real life experiments, the conclusions may remain statistically insignificant if the sample size is too small, and a larger sample size is impractical or too costly. With computer experiments, one can repeat the experiment many times until the sufficient statistical confidence (e.g., 95% confidence interval) is obtained to confirm the result.

The comparison of real-life experiment and computer experiment workflows shows that they share much in common; the major difference is the use of a *computer simulation model* as the substitute of the real-life objects (Steps 3 and 4). In the following Section, we define the terms and components that make up such a computer simulation model.

1.2.3. Preliminary definitions

In the previous Section, we have shown that computer experiments are conducted on *computer simulation models*, which is a fundamental difference from real-life experiments and the root of which new challenges arise. In this Section, we formally define computer simulation model and the related concepts.

Definition 1-1

A *computational simulation model* is a tuple $(\mathbf{SV}(t), \mathbf{BR}, \mathbf{C})$, where $\mathbf{SV}(t) = \{SV_1(t), SV_2(t), \dots, SV_n(t)\}$ is a set of *state variables* (or *SVs*) that can change over the simulation time; $\mathbf{BR} = \{BR_1, BR_2, \dots, BR_m\}$ is a set of *behavioral rules* (or *BRs*) that operate on $\mathbf{SV}$ by defining how and when the state changes occur; and $\mathbf{C} = \{C_1, C_2, \dots, C_p\}$ is a set of *constant parameters* (or *CS*).

Definition 1-2

Simulation time t_{sim} , or simply t , refers to the virtual time within a simulation. For example, the ward model is simulated for “a period of 2 years”. This is in contrast to the *computational time* or the *running time* t_{run} which is the actual time it takes to run the simulation on a computer, for example, it takes “half an hour” to complete the simulation. It is desired that $t_{run} \ll t_{sim}$, that is, the simulation runs much faster than their real-world counterpart so as to make predictions.

Definition 1-3

A *simulation run* of a computational simulation model is the process of updating its $\mathbf{SV}$ over simulation time by applying $\mathbf{BR}$ at the correct timing in the correct order. The simulation run produces the sequence of *SVs* over the simulation time t : $\mathbf{SV}(t_0), \dots, \mathbf{SV}(t_i), \dots, \mathbf{SV}(t_T)$, where t_0 and t_T are the start and end time of the simulation, respectively.

With respect to Definition 1-1, we would consider a queuing model [37] as a computational

simulation model if it explicitly models customers in the queues and the servers that serve the customers. However, if the model is meant for pure analytical computations without the intention of executing any simulations over time, such as curve fitting or data regression, we would refer to it generically as a *computational model*, but not a *computational simulation model*.

Often, the *SVs*, *BRs* and *Cs* of a computational simulation model can be naturally aggregated into logical objects which we shall refer to as *entity instances*. We exemplify this notion with the same example of the queuing model. Each customer and each server can be seen as an entity instance: A customer entity instance possesses behaviors such as arriving, queuing, and departing. A server entity instance possesses behaviors such as serving the customer. Many entity instances of customers and servers may exist in the model; however, they can be seen as only two *entity types*: the customer *entity type*, and the server *entity type*. Entity instances belong to a same entity type if they share the same state variables, behavioral rules, and constant parameters. We formally define these concepts as follows:

Definition 1-4

An *entity instance*, or *EI*, is a tuple $(SV(t, EI), BR(EI)_{local}, BR(EI)_{interact}, C(EI))$, where $SV(t, EI) \subset SV(t)$ is a subset of the *SVs* of the computational simulation model (or simply the model) that belongs to that EI; $BR(EI)_{local} \subset BR$ is a subset of the model's *BRs* that only operate on that EI's *SVs*; $BR(EI)_{interact} \subset BR$ is a subset of the model's *BRs* that operate on the *SVs* of both that EI and other EIs; and $C(EI) \subset C$ is a subset of the model's *Cs* that are used by the EI. An EI represents a logical object of interest in the given system.

Definition 1-5

An *entity type*, or *ET*, is a tuple $(SV(ET), BR(ET)_{local}, BR(ET)_{interact}, C(ET))$, which are templates of *SVs*, *BRs* (local and interact), and *Cs* of those EIs that are derived from this ET. An ET represents how modeler classifies EIs into logical types based on the way the EIs share the same definitions of *SVs*, *BRs*, and *Cs*.

It is more convenient to view a simulation model as a collection of EIs and their interactions, since they can be mapped onto the real life counterparts they attempt to represent. This will be the view

we shall take for the subsequent discussions.

With the definition of a *computer simulation model*, we define a computer experiment as follows:

Definition 1-6

A *Computer experiment* is an experiment conducted using a computational simulation model, in contrast to the real world experiments which are conducted on the real world subjects. An experiment here has the usual semantics as defined in the Oxford dictionary: a scientific procedure undertaken to make a discovery, test a hypothesis, or demonstrate a known fact [2].

With *computer simulation model* and *computer experiments* formally defined, we discuss in detail the challenges of performing computer experiments in the next Section.

1.3. New challenges of computer experiments

Computer experiments face different types of challenges than their real-life counterparts, arising from the use of a *computer simulation model* as the substitute of the real-world objects and the basis of experimental manipulation. In this Section, we elaborate on three distinct challenges: to build a sufficient representation of the real world; to implement the experiment on a null model, and the use of control and randomization.

1.3.1. Challenge of building a sufficient representation of the real world

We have loosely described the notions of *null computational simulation model* and *experiment model* in Section 1.2.3. We formally define them below:

Definition 1-7

A *null computational simulation model*, or simply a *null model*, is a computational simulation model intended as a control for a computer experiment. Its content ($SV(t)$, BR , C) reflects the modeler's judgment, with respect to the purpose of the study, on the collections of SV s, BR s, and C s that sufficiently represent the real world system for experimental manipulations. Its simulation result predicts the outcome of a system that is free from experimental manipulations,

i.e., the outcome of a control group.

The null computational simulation model should be validated in both its modeling details and the simulation result, preferably with the help of domain experts. Iterations of revisions may be required if the model's scope or level of details are not appropriate for the purpose of study. Through model validation, we gain confidence that the null model can serve as the basis for the subsequent experimental manipulations

Definition 1-8

An *experiment model* is a computational simulation model that is modified from the *null model* with treatment group specific changes. Formally, it is a tuple ($D_{SV}(SV(t))$, $D_{BR}(BR)$, $D_C(C)$), where D_{SV} , D_{BR} , D_C are conversion functions that modify the null model's SVs , BRs , and Cs to form that of the experiment model, respectively.

The conversion functions D_{SV} , D_{BR} , D_C may contain changes such as addition, removal, and replacement. For example, an experiment that tests the unusually high or low HIV mutation rates would directly replace values of mutation rates, which can be a constant value or constant parameters of some random distributions, with new values. An experiment that tests the syncytia formation during HIV pathogenesis need to add SVs , BRs , and Cs that describe how syncytia are formed from T helper cells, and how it causes cell deaths.

The construction of a null model is practically nontrivial as it requires the modeler to fully study and understand the real world system, make abstraction on entities and interactions, build the model on computer, and validate its details and simulated results. Models in general, whether they are conceptual models, physical models, mathematical models, or computational models, are merely abstractions of the reality, they are *inevitably unrealistic* in various ways, due to the omission of details and simplification assumptions [38]. If a model approaches reality, its would require infinite numbers of SVs , BRs , Cs :

$$|SV| \rightarrow \infty, |BR| \rightarrow \infty, |C| \rightarrow \infty, \text{ if } \text{Model} \rightarrow \text{Reality}$$

Where $|\bullet|$ is the cardinality of the set.

Simplification is necessary for *SVs*, *BRs*, and *Cs* to be tractable in order for the model to be built, analyzed, or simulated efficiently with limited resources in both time and space. It is common to see simple models with assumptions are not strictly realistic, but they become popular and accepted over time for their capability to make useful predictions and when events unfold to confirm their predictions. For example, classical economics models assume *perfect information*, *profit maximization*, and *rational choices*, which are not entirely true in many real life settings [39]. Milton Friedman argued that the usefulness of an economics model should be judged by its fruitfulness as an engine of prediction, not by its descriptive realism [40]. This is in line with the Occam's razor [41] which recommends the simpler theories given that they give same explanatory power as those more complicated theories. In practice, many models continue to be useful despite of their simplicity: they are accepted and used for situations where the assumptions largely apply; meanwhile they are criticized and motivate development of improved models with more explanation power and accurate predictions. For example, contemporary economic theories, such as behavioral economics [42, 43] are motivated to address bounded rationality.

Our key challenge with the null model construction is that it needs to be a *sufficient representation* of the real world *for the purpose of computer experiment*, despite of the fact that it will remain imperfect, unrealistic, and contradictory to the reality in various ways. This implies that it need possess certain level of *model realism* that at least allows for results to be validated by a domain expert. Take the example of a hospital ward simulation model, where the researches are interested to know whether the meal porters' adherence rates to hand hygiene polices have any impact on the nosocomial transmission of certain contact-based disease. If the model simply assumes 2 human entity types – patients and staff, without sub-classifying them into different roles (such as medical officers, physiotherapists, nurses, cleaners, meal porters), then it would be impossible for such a model to answer the question, because the elements concerned by the question (i.e., the meal porter) simply do not exist in the model. If say, the model does sub-classify the human entities according to their roles so that the meal porters are existent in the model, however *for simplicity*, assume that meal porters' physical contacts with patients or other staff are negligible because such contacts are few and “unimportant” as compared to the patient-nurse contacts or patient-physician contacts, then the model cannot answer the question either. If the result shows that the meal porter's hand hygiene adherence has no impact on the disease transmission, this is not a discovery but an artifact due to the unrealistic

assumption which artificially removes the meal porters' contacts.

Though the example above may seem trivial, it is a challenging task when one considers that there are possibly an infinite number of details one can add to the model, and to decide which could be simplified and which should not, given the constraints of resources without compromising the need for necessary and sufficient realism (Occam's razor [41]). A computer experiment cannot allow investigations or make discoveries about entities and their relationships if they are not directly encoded apriori in the model or are not derivable during simulation. Without sufficient and acceptably correct details necessary for validating a hypothesis, simulation results would be invalid. In other words, a null model as a sufficient representation of the reality should have appropriate *model scope* and *the level of details*. We define the term *model scope* below:

Definition 1-9

The *model scope* of a computational simulation model is the set of *entity types (ETs)* present in the model.

As defined in Definition 1-5, an entity type is a template that specifies what the entity is (e.g., it is HIV), what behavior rules it has (e.g., it replicates inside infected cells; it infects CD4+ cells), and the parameters that quantify it states and behaviors. For ease of description, we will use only the entity's name to represent the entity type. For example, the scope of an HIV pathogenesis model may contain only HIV virions and T helper cells, hence its model scope is {HIV, TH}. Another model may contain HIV, T helper cells, B cells, antibodies, and T killer cells; hence its model scope is {HIV, TH, B, AB, TK}. The latter model has a wider model scope than the former one. When approaching reality, the cardinality of the model scope will approach infinity.

The level of details of a model is characterized by the *entity granularities* of the entity instances in the model, which we define as follows:

Definition 1-10

The *entity granularity* of an entity instance is defined by its set of state variables, $SV(t, EI)$.

Intuitively, the more state variables are used to describe an entity instance, the more detailed is the entity model. A computational simulation model may have possibly large number of entity instances, and each could have a different granularity level, resulting in heterogeneous entity granularity levels within a single model.

Deciding on an appropriate model scope and entity granularities for a null model is challenging in that such decisions are influenced by the modeler's domain knowledge and judgment, which are subject to continuous revisions and corrections as the knowledge advances. Therefore, it is desirable to have a methodology for building computational simulation models that allows for iterative revisions on model scope and entity granularity. This will be a main topic in this thesis.

1.3.2. Challenges of implementing experiments on a null model

Both real world experiments and computer experiments need to artificially manipulate the factors under investigation. Experimenters may have to compromise if the desired manipulations are not practically feasible due to physical constraints. For example, animal experiments need to adhere to the "3Rs" of animal research, namely, reduction of animal numbers, refinement of methods, and replacement of animals by viable non-animal alternatives [3]. Besides, the experiments may require approval from multiple organizations (e.g., Institutional Animal Care and Use Committee or IACUC).

Computer experiments, on the other hand, may look much easier since theoretically, anything in this computer program, whether it is a variable value or statistical distribution, an if-else rule, or a mathematical equation, can be changed at the experimenter's will. As defined in Definition 1-8, an *experiment model* is itself a computational simulation model, derived by modifying the *null model* by applying treatment group specific (or experiment specific) changes to the *SVs*, *BRs*, and *Cs*. Though making such programming changes are usually technically easier than making manipulations on real world subjects, there exist a different kind of challenge that is rooted in the construction of an *appropriate* null computational simulation model that underlies the experiment models. We illustrate this problem with an immune system model. Such a model may include cells populations such as TH cells¹, TK cells¹, B cells², Macrophages³, and molecular populations such as antibodies

¹ TH cells: or T Helper cells, are lymphocytes that detect the antigens on the antigen-presenting-cells (APCs) and secrete molecules to activate TK and B cells

and a subset of interleukins. One could think that this model is sufficiently detailed as a null model since it is capable of reproducing the dynamics of primary and secondary immune responses⁴, the basic behavior of an immune system [44, 45]. However, if the computer experiment concerns the balance of TH1/TH2 in regulating the cellular immune responses (dominated by TK cells) the humoral immune responses (dominated by B cells) [46, 47], the null model misses many details which, as an ad-hoc solution, will have to be added to the experiment model. For example, the experiment model has to extend the null model by modeling the TH cell's life cycle of proliferation and differentiation into TH1 and TH2 subtypes, the respective cytokines secreted by each TH subtype, and the interaction mechanisms of these cytokines in activating B cells and TK cells.

The example above raises the question where the line between a null model and an experiment model should be drawn: namely, which set of details should go into the null model, and which are the experiment specific details that only need to be implemented in the experiment model. Let $(SV_o(t), BR_o, C_o)$ be the null model, $(SV_E(t), BR_E, C_E)$ be the experiment model, and their difference be $(\Delta SV(t), \Delta BR, \Delta C)$, namely, a tuple of change set of SV s, BR s, and C s. If a SV is the same in both the null and experiment model, it will not appear in the change set $\Delta SV(t)$; same apply for BR s and C s. We argue that the change sets should be much smaller than the null model, namely, $|\Delta SV(t)| \ll |SV_o(t)|$, $|\Delta BR| \ll |BR_o|$, $|\Delta C| \ll |C_o|$. That is, we would expect the experiment model to be a *slightly* modified version of the null model, rather than a drastically different model by itself. In practical terms, it would mean that the null model needs to include sufficient details which may not appear to be necessary if seen alone, but will be necessary in the face of experimental manipulations. This is analogous to a real world experiment, where the experimenters attempt to modify only a small set of properties of the real system while keeping other variables fixed or constant. With dramatic changes in a complex individual-based model that lacks a globally prescribed dynamic, it is often difficult to deduce at face value which part of the change accounts for observed changes in the

¹ TK cells: or T Killer Cells, Cytotoxic T lymphocytes (CTLs): are lymphocytes that, upon activation, kill the cells that are infected with virus

² B cells: are lymphocytes that differentiate into antibody-secreting plasma cells upon activation

³ Macrophages: are leukocytes that can engulf pathogens in the tissue. They also present the processed antigens on their cell surface, serving as antigen-presenting-cells (APCs).

⁴ The immune response occurring on the first exposure to an antigen is known as the primary response. The subsequent exposures will trigger a much faster and more efficient immune response known as the secondary response.

experimental outcomes.

Therefore, the construction of the null model must take into account the requirements of the computer experiment when deciding its scope and entity granularity. As demonstrated in the example of immune system model above, a null model could appear to be a sufficient representation if its purpose is limited to conceptualization, demonstration of capabilities, or making straightforward predictions, but without the need of experimental manipulation. In such cases, much of the details may be deemed unnecessary and omitted, following the Occam's Razor [41, 48]. It would not be a surprise if the null model is found to be inadequately detailed for the experiment hence requires iterative revisions. This reinforces the goal described in Section 1.3.1: we shall focus on developing a modeling methodology and framework that supports iterative model revision and extensions; this would allow us to build null models that are both a sufficient representation of the real world system, and a platform appropriate for experimental manipulations.

1.3.3. Caution on the use of control and randomization

When chosen well, control groups are often used in real world experiments to eliminate the effect of confounding factors. That is, by selecting members in the control group and the treatment group so that they are non-differentiated in all characteristics aside from the factor(s) that are being studied. In practice, randomization techniques have to be used to minimize selection-related biases [19, 20, 24, 49]. For example, in randomized clinical trials (RCT), it is required that each patient be *randomly assigned to a group*, and all the patients in the group receive (or not receive) an intervention [50]. Double-blind experiments require that neither the patients nor the researchers know who belongs to the control group or the treatment group (and will only know it after the data collection), so as to minimize prejudices and unintentional physical cues on the results [18-20, 51].

Table 1-1 shows the importance of having the set of common factors **A** (or simply the common **A**) across the control and treatment groups. Randomized selection of members for control can be understood as set of practices that strive for the common **A**. In many study designs (prospective, observational or retrospective [21]), it is very costly to engender identical subjects (such as “identical patients”) for the control and treatment groups, since human beings are inevitably different from each other in a myriad of ways. Though these properties cannot be set to identical values, researcher can

assume that they belong to independent and identical statistical distributions. For example, the ages of the patients under study can be seen as random samples from the age distribution of the human population that is hospitalized. By randomly assigning patients to control and treatment groups, it can be expected in theory that the age distribution of the two groups would follow the same distribution (the common **A**). The practice attempts to minimize the selection bias [18, 52]. One example of such bias would be the control group receives the younger patients and the treatment group receives the older ones, hence violating the assumption of common **A**. The notion of common **A** also includes application of the same treatment procedures, and the absence of prejudices from the researchers.

In contrast to the real world systems where heterogeneities are to be controlled, computational models for experimentation require differences to be made representative of the population. As such, it is simpler to implement homogeneity (where simulated entities have identical properties and behaviors) than heterogeneity (simulated entities with different properties and behaviors that are representative of the population) with computers. This leads to the following corollaries:

1. Creating a common **A** is technically trivial with computer experiments because unless explicitly set, all the properties are identical by default (e.g., same values, same statistical distributions, same mathematical equations, and same interaction rules).
2. As the Occam's Razor tends toward simpler models, a computational simulation model risks being unrealistically deterministic if the modeler does not sufficiently address the heterogeneities and random factors presented in the real world system.

The first corollary can be seen as an advantage of the computer experiment. The setup of control and treatment groups is much more straightforward than its real world counterpart. A simple scenario could be, the null model is used as the “control group”, and the experiment model (with the experiment-specific changes, which is most likely the factor(s) under investigation) is used as the “treatment group”. One may argue that since every factor is technically “controllable” by the modeler, there is not a need to set up a control group. However, there are numerous tunable factors in a model and it is not the interest of a study to vary *every* tunable parameter to test their results. Thus in a computer experiment, we use null model to set up a baseline scenario that resembles the

default state of reality as close as possible. The majority of the factors in this null model will remain the same throughout the experiment, that is, they constitute the “control group” for the computer experiment.

The second corollary is closely related to the issue of model realism. The real world contains measurements that are constant, periodic, prescribed, dynamic, stochastic and chaotic. Computational models can simulate this entire range of phenomena including stochastic behaviors via the use of pseudo random number generators [53]. As with all stochastic simulations, the model has to be run many times to reveal the statistical behavior (the averages and deviations) of the system over time or space. Take a ward model as an example. The patients’ admission events need not occur at fixed intervals; instead, they are separated by geometric periods of time. In our study of the NUS SICU ward [54], the patients’ length of stay (LOS) greatly varies from 1 to 8 days, with 4 days as the average. The visitation sequence taken by a physician visiting patients in the ward also varies, thus, the model may randomly choose a patient as the starting point for each visit. The durations of activities are also not fixed. For example, the roll call of nurses ranges from 10 minutes to 45 minutes, with 30 minutes as the average [54]. One could argue that certain random elements have to be present in the model because they may influence the effect of disease transmission by altering the duration and order of the physical contacts between human beings. Through the exact same reasoning, one could argue that other random elements can be excluded because they are not seen (to the modeler’s knowledge) to have impact on the study. For example, a modeler may omit the age property of the patients if she judges that age is a less relevant factor in the study. Without an age property in the model, there would be no need to set a statistical distribution of age values. It would imply that the human beings in the model would be *homogeneous* in terms of ages since there is no explicit attempt to make them different in this respect.

In summary, though both the real world and computer experiments use the notion of randomization, they have related yet different meanings. Real world experiments use randomization in forming control and treatment groups so as to minimize confounding effects. Computer experiments use randomization to simulate a subset of heterogeneous properties and behaviors present in the real world system that could potentially influence the experimental outcomes.

1.4. Motivations and scope of this work

Our work is motivated by the great potential of computer experiments as a less costly, safer, and possibly more powerful alternative of the real world experiment. We are particularly interested in applying such computer experiments to predict epidemiological outcomes. As described in Section 1.3, the primary challenge underlying any computer experiments is the construction of a null computational simulation model that can serve the purposes of presenting the real world system (with the validation by domain experts), predicting the experimental outcome of a control group, and providing the basis of experimental manipulations. Computer experiments for epidemiological studies are especially challenging due to the numerous factors that potentially need to be modeled.

Epidemiological studies cover a wide range of topics and concern many professions and disciplines. On the macro level (human hosts and populations), the epidemiologists study the causal relationship between exposures of the hosts and their disease development, trace the transmission routes, and conduct bio-statistical analysis; the hospital decision makers may have to try out different precaution measurements and implement operational changes in the face of disease outbreaks. On the micro level (intra-host level), microbiologists and pharmacologists try to understand the pathogenesis of the bacteria or the virus, and develop medications and vaccines for the disease. These disciplines involve a lot of *in vitro* and *in vivo* experiments and are often expensive, costly to execute, involve a myriad factors that are difficult to separate or control, and may expose the researchers or experimental subjects (such as patients) to health hazards. We believe that computer experiments are beneficial to epidemiological studies for the advantages discussed in Section 1.1, notably: reducing the cost and risks, completing experiments in a shorter time, and giving the experimenter full control over control variables. Though they are not intended as replacements of the real world experiments, their results are useful in suggesting directions for real world experiments.

Deciding an appropriate model scope and entity granularities is challenging and usually driven by the purpose of the study. Subjects in the epidemiological studies, being the cells and molecules on the micro level, or the human hosts on the macro level, are complex entities with incredibly large and diverse set of properties and behaviors, individually and collectively. Therefore, we set our primary goal as to develop a modeling framework for building computational simulation models that are suitable for computer experiments. The framework here refers loosely to our theory of what

constitutes a scalable and extensible software architecture for building models for computer experiments. We will describe the components of this framework in detail in Chapter 3. The specific sub-goals are:

1. To distinguish the advantages and limitations of existing modeling techniques through literature review and preliminary feasibility studies. We study in depth two popular modeling techniques for the computational simulation model: the Differential Equation (DE) based approach, and the Agent-Based approach. With first-hand insights, we elucidate the inherent limitations of these modeling techniques.
2. To propose the architecture and the building blocks for a modeling framework that addresses the challenges and limitations revealed by our preliminary studies: This includes methods to manage the multi-scale nature of the system in both the temporal and spatial dimensions without needing to sacrifice model realism, and to extend entity granularities (which collectively form the model's level of detail) iteratively and systematically when required. We construct simple *demonstrative* simulation models (which are not full null simulation model themselves) to explain the working principles and capabilities of the proposed building blocks.
3. To apply the proposed modeling framework to computer experiments on real world studies in immunology and epidemiology.

Equipped with the modeling framework, we are then in a better position to construct null models that can be used as virtual control groups for conducting the computer experiments. We apply the full computer experiment workflow presented in Section 1.2.2 to a real application that investigates the nosocomial infections (i.e., the infections that take place in the hospital environment) caused by the bacteria Methicillin Resistant *Staphylococcus Aureus* (or MRSA). The experiment attempts to answer the question that concerns the hospital decision makers: to what extent does hand hygiene compliance by the health care workers and staffing level influence the nosocomial transmission of MRSA?

We further elaborate the use of computer experiments with multiple hypotheses using the application of chemotaxis, the microbiological phenomenon where the cells move towards the higher concentration of chemoattractant. This experiment is an analogy of a *thought experiment* where the

experimenter proposes plausible explanations (hypothesis), makes inferences based on the explanations mentally (without physically conducting the experiment), and rejects those that lead to contradictory results [55, 56]. Traditional thought experiments are relatively easy hence mental inferences are probably sufficient, such as the one described by Galileo in *Discorsi e dimostrazioni matematiche* (1628) which rejected the common thought that the heavier object falls faster than the light ones¹ [57]. If the interaction dynamics in the system is complicated and mental inference is difficult, computer experiments can then be used to replace the mental inference process, which is the goal we wish to achieve with the application of chemotaxis.

1.5. Organization of the thesis

In Chapter 1, we have introduced the computer experiments in general in terms of the advantages, limitations, compositions, and challenges. We set our work scope to be developing a modeling and simulation framework for computer experiments for epidemiological applications.

In Chapter 2, we review the literature to understand the experimental requirements of epidemiological applications, and analyze the strength and limitations of agent-based models when compared to conventional mathematical modeling approaches. We validate our findings with a small-sized model of the human immune system with HIV infection, which is used to verify the HIV pathogenesis hypotheses. We present insights on the practical problems of ABMs through this limited model.

In Chapter 3, we present our research roadmap and the architectural overview of our proposed modeling framework. Its building blocks are to be further studied and developed in Chapters 4 - 6.

In Chapter 4, we propose an event-scheduling update scheme for simulating multi-timescale activities. This proposed update scheme is shown to be both realistic and computationally more efficient than its time-step driven counterpart. Two representative studies are presented to validate our results, namely; modeling the B cell life cycle, and the admission and discharge of patients as well as nurse

¹ Galileo asked: if two objects with different masses were tied together, would the united body fall faster than the heavier one (since their net mass would be greater than each of them), or fall slower than the heavier one (since the lighter object should fall slower and hence retard the speed of the heavier object)? Both are valid inferences from the hypothesis that heavier object falls faster; yet they contradict with each other.

work routines in a ward.

In Chapter 5, we propose a layered compartment model and a hybrid entity representation in order to address the requirement of appropriate entity relative scales and topologically accurate environment models. The application of these methodologies is demonstrated with two case studies; chemotaxis, and ventilation assisted nosocomial airborne disease transmission.

In Chapter 6, we propose a formal description for agent using agent behavior network which facilitates model validation and incremental model revision. We then propose the iterative strategies for model extension. We demonstrate the approach by systematically revising and extending ABMs through three chemotaxis models with increased entity granularity.

In Chapter 7 and Chapter 8, we apply the proposed modeling and simulation framework to two full scale computer experiments. The first application makes predictions that address hospital stakeholders' what-if concerns, which can potentially be used for practical decision making. The second application assesses the causal intuitions of the quantitative relationship between chemoattractant concentrations and gradients, and the cell locomotion.

In Chapter 9, we conclude our work and propose the possible future works.

Chapter 2

Background and Motivations

To understand the background and motivation of building models for computer experiments, we have conducted literature review in many relevant fields. Our focus has been in two areas: the domain knowledge, and the modeling techniques. For the domain knowledge, the aim is to comprehend the topics and terminologies in the domain in order to understand their modeling requirements and to communicate with domain experts. We have studied the microbiological applications of immune system [44, 45, 47, 58-60], HIV pathogenesis [61-66], and chemotaxis [29-31, 34, 35, 67-69]. We have also studied epidemiologic applications, in particular, nosocomial MRSA infections [70, 71] and the intervention methods [72-75]. For the modeling techniques, the aim is to understand the existing achievements and the limitations of various modeling techniques. We have studied the conventional Differential Equation (DE) based mathematical models [76-79], and also the more recent Agent-Based Models (ABMs) [80-84]. Given the modeling requirements imposed by the domain, we further studied the existing modeling strategies that address the multi-scale issues [85, 86], the explicit modeling of the environment and space [87, 88], scalability of models [89], and the adoption of distributed and parallel processing [90-93]. Due to the length constraint, in this Chapter we only present our reviews of literature that are directly relevant to the discussions on computer experiments.

The computational simulation models for computer experiments are distinct from those that are for simpler purposes such as conceptualization, demonstration or education. The former potentially requires a wider scope and a more detailed entity granularity in order to be useable for experimentation. Creating models for computer experiment requires thorough domain knowledge on how the real world system works and how the experiment goals affect the modeling requirements. In this Chapter, we aim to understand the requirements and complexities that the epidemiological applications impose on the modeling process, so that informed decisions on the modeling technology can be made. We present the domain background of two example applications: for immune system dynamics with HIV infections, and for nosocomial MRSA transmissions within hospital wards. With the understanding of the epidemiological applications and their modeling requirements, we

proceed to review the strength and limitations of the existing modeling techniques, in particular, the Differential Equation (DE) based approach, and the Agent-Based approaches. To validate our findings, we design and conduct a preliminary computer experiment to investigate the sufficiency of HIV pathogenesis hypotheses. This preliminary experiment reveals important insights on the practical modeling issues with ABMs. The findings in this chapter motivate our work of proposing a modeling framework for computer experiments, which will be presented in Chapter 3-6.

2.1. Background

In this Section, we briefly present the domain knowledge of two example epidemiological studies: one on the human immune system and HIV-1 pathogenesis on the intra-host level, and one on the nosocomial infections in a hospital ward on the inter-host level. The purpose of this Section is two-fold:

1. To enable the discussion on the specific requirements these applications impose on the modeling process; by understanding these requirements, we will be make informed decisions on the modeling techniques.
2. To provide the reader the minimal knowledge to understand the modeling technique review in Section 2.2, and computer experiments presented in Section 2.3 and Chapter 7.

Given the tremendous amount of detail the domain knowledge entails, we have made simplifications and omissions which we deem appropriate for a more concise exposition. Interested readers may refer to relevant text books and literatures mentioned in this Section for a more comprehensive understanding of the domains.

2.1.1. The human immune system and HIV-1 pathogenesis

Acquired Immune Deficiency Syndrome (or AIDS) is characterized by a combination of opportunistic infections and a markedly reduced circulating T Helper (or TH) cell count [63, 94]. The disease is known to be caused by the Human Immunodeficiency Virus (or HIV) which directly infects the cells bearing CD4+ molecules (including the TH cells). Various hypotheses have been proposed, attempting to identify the pathogenetic mechanism within the HIV-TH infection process that can account for the inevitable progression to AIDS. We will present four representative hypotheses in

this Section starting with basic background knowledge on the composition and microbiological function of the human immune system. Interested readers may refer to immunology texts [45, 95] or literatures [62, 63, 65, 96, 97] for a more comprehensive introduction to this field of research.

Human immune system involves many kinds of cells (mainly the white blood cells, or leukocytes) and molecules. The cells move between several immune organs, blood vessels, and body tissues, guided by the changes in chemokine concentrations [45]. There are two main types of immune responses. The *innate immune responses* involve cells that attack pathogens in a non-specific manner, known as mast cells or phagocytes. They are innate in that humans are born with such immunity, and non-specific in that they do not adapt to newly encountered pathogens. The other type, the *adaptive immune response*, involves mainly the lymphocytes including T cells and B cells. This immune mechanism adapts to new pathogens and confers long-lasting or protective immunity to the host, which forms the basis for vaccinations. T cells and B cells start in an inactive state, known as the naïve state. Upon recognition of antigens, they undergo activation and proliferation (cell divisions) that last 3-4 days and differentiate to the end state, known as the effector state. Some of the cells differentiate into a memory state, which are long-lived cells that can be activated again in subsequent encounters with the antigens.

The T Helper (TH) cells, a subtype of T cells, have a central regulatory role in adaptive immunity, as shown in Figure 2-2. They interact with antigen-presenting-cells (APCs) such as macrophages (MPHs) and dendritic cells (DCs), which ingest the pathogen and present the antigen fragments on the cell surface in the form of a MHC-peptide complex. MPHs then secrete interleukin-1 (IL-1) molecules that activate TH cells. When physically in contact with APCs, TH cell surface receptors (TCRs) detect and bind to the antigen presented on the surface of APCs by shape matching. Activated TH cells upon antigen recognition secrete a variety of stimulatory molecules, such as IL-2, to trigger proliferation of cytotoxic T lymphocytes and B cells (B) so as to maximize the phagocytic activity of MPHs. TH cells can mount an immediate attack on infecting and infected cells that present unusual surface antigens, a process known as *cellular immune response*. On another front, activated B cells divide and differentiate into plasma cells, which produce antibodies (also known as immunoglobulins, a type of protein) which bind to matched free antigens, neutralizing the antigen's virulence and serve as a marker for MPHs. MPHs bind to the antigen-antibody complex and

perform phagocytosis. This form of immune defense that involves B cells secreting antibodies is otherwise known as a *humoral immune response*.

Figure 2-2 Central role of TH cells in cellular and humoral immune responses
(figure adopted from [58])

HIV is known to infect CD4⁺ cells (cells bearing CD4 molecules, including TH cells, dendritic cells, and MPHs). The killing mechanisms include: direct viral killing of infected cells; increased rates of apoptosis in infected cells; and killing by TK cells which recognize the infected CD4⁺ cells [98]. HIV is a lentivirus (a member of the retrovirus family) that is replicated in a host cell via the enzyme reverse transcriptase to produce DNA from its RNA genome [99]. Because reverse transcription lacks the usual proofreading of DNA replication, the replication process is error-prone and hence HIV has high mutation rates, which is thought to be the reason why some HIV strains are able to evade immune attacks and persist in the host [62]. Due to the central role of TH cells in stimulating and regulating an immune response, their abnormal loss from HIV infections is a convincing characterization of an immune system's failure at the onset of AIDS [94]. The clinical progression of HIV infection towards AIDS typically follows three phases [62, 76], as shown in Figure 2-3. The patient exhibits acute symptoms during the early phase of infection, which lasts a few weeks. With the onset of HIV-specific antibodies and TK cells, the amount of virions sharply declines by a factor of 100 or more. The viral load then remains at a relatively low but constant level, while the TH cell count slowly decreases for a period up to 20 years. When the TH cell count falls below 200/ μ l, AIDS is then considered to have developed [76].

Figure 2-3 Typical 3-stage disease progression of HIV infected patient (figure adopted from [98])

Besides attributing TH cell losses to direct infection by HIV, other mechanisms have been proposed to explain the slow but inevitable loss of TH cells over the years of infection. These include: rapid viral genetic mutations [62], syncytium formation [100, 101], filling of CD4 receptor sites [96], HIV-induced cell apoptosis [66], impaired production of new TH cells [94] and the existence of viral reservoirs [64]. We briefly introduce 4 of them below.

- *Direct Effect on CD4+ Cell:* During the period 1983 to 1984, Montagnier and Gallo advocated that HIV had a direct cytopathic effect on CD4+ cells (such as TH cells) [65]. They believed that HIV could infect and destroy the TH cell, which in turn causes the immune system to lose its immune function when the TH cell population level becomes too low.
- *Rapid Viral Mutation:* HIV replication is error-prone during reverse transcription which results in mutant strains [62]. Hence the immune system needs to detect each type of mutant strain before it is able to activate the specific antibodies. It is postulated some of the viral strains could persist by evading from detection by the immune systems.
- *Syncytium Formation:* The mature HIV envelope glycoprotein (gp120/gp41) expressed at the surface of infected cells drives the cell-to-cell fusion with adjacent uninfected TH cells which results in the formation of multinucleated syncytia [100]. After several rounds of fusion, syncytium attains volumes equivalent to several dozens or hundreds of individual cells. Cell apoptosis is then triggered, which then contributes to the global loss of T cells.
- *Filling of CD4 Receptor Sites:* HIV is known to attack CD4 receptors in at least two ways [96]. First, HIV virions can attach to CD4 receptor sites via its gp160 spikes. Second, HIV is capable

of releasing or freeing its exterior gp120 envelope glycoprotein, thereby generating a molecule that can actively bind to CD4-bearing cells. Because CD4 receptors assist T Cell Receptors (or TCR) in binding to the MHC-II-peptide complex, it is speculated that TH cells lose their immune function as a result of CD4 receptor site filling even they are not infected.

These hypotheses provide plausible explanations of the onset of AIDS, and their described mechanisms have been supported by evidences from wet lab observations. Here, we are interested in how sufficient are these mechanisms to cause AIDS. This will be the objective for preliminary computer experiments to be presented in Section 2.3.

Based on the descriptions above, if one would develop a computational simulation model for human immune system or HIV pathogenesis, following requirements must be observed:

1. *Heterogeneous entity types*: the system contains a wide range of heterogenous entities, such as biological cells, molecules, and viral particles;
2. *Complex entity states and behaviors*: Entities such as biological cells have distinct and complex lifecycles. For example, the lymphocytes undergo state changes upon activation and for each state, their behavior will change;
3. *Non-linear entity interactions*: interaction mechanisms among the entities are complex and non-linear, such as secretion, division, activation, suppression, binding, and killing;
4. *Non-uniform interactions in a topologically complex environment*: the interactions do not take place uniformly in a well-stirred environment; the probability distribution for interactions is dramatically altered by the non-random movements of cells through the immune organs, guided by the concentrations of chemokines circulating in the blood stream;
5. *Multi-scale in spatial and temporal domains*: the system is multi-scale in nature, in both temporal and physical dimensions. In the temporal dimension, the cell activation and division last 3-4 days; the acute phase of HIV infection lasts several weeks, and the latent phase lasts several years. In the physical dimension, entities vary greatly in their physical size and relative quantities: antibodies are about 15 nm [102], HIV is about 0.1 μ m [103], while a T cell is around 5-10 μ m [104]. In terms of relative quantities, CD4 epitopic density on circulating lymphocytes are measured to have a mean level of 46,000 epitopes per cell [105];

6. *Demands for great level of details:* molecular details of the immune responses and viral pathogenesis are important parts of the hypotheses. This implies that the model needs to be detailed enough to account for these molecular details, at least for those involved in the hypotheses.

In the next Section, we describe the domain background of another epidemiological application on the macro (human host) level, in particular, the nosocomial transmissions in a hospital ward environment. As we shall see, the six modeling requirements we listed above will re-occur for the macro level application. Hence we will use them as the benchmark for the discussion of the modeling techniques in Section 2.2.

2.1.2. Nosocomial infections in a hospital ward

This application focuses on how the operational factors, in particular, the staff hand hygiene compliance and the staffing level, can alter the epidemiological dynamics of nosocomial Methicillin Resistant Staphylococcus Aeurus (MRSA) transmissions. Nosocomial infections due to MRSA have become endemic in many countries in the last two decades [70, 106-111]. When they are imported into a hospital environment, such multi-drug resistant infections result in increased morbidities and mortalities, prolonged patient stays and higher treatment costs. In Europe, 5 million cases (amounting to 25 million extra hospital days) occur with a total cost of 13-24 billion Euros and 50,000 deaths per year. In the US, incidence rates reached 4.5% in 2002 or 9.3 infections per 1000 patient days. This resulted in 1.7 million affected patients and 99,000 deaths [112]. And in the UK, the National Audit Office figures show 300,000 patients pick up infections in hospitals each year and 5,000 die as a result. The cost to the National Health Service is estimated at £1 billion per year. Thus, nosocomial MRSA has become an unprecedented international public health concern.

Various intervention measures have been proposed and implemented in attempts to reduce nosocomial MRSA. They include patient screening and surveillance [72, 113-115], isolation and decolonization of the MRSA+ patients [74, 75, 116, 117], restrictions and guidelines on anti-microbial use [118], and programs to encourage and audit hand hygiene compliance of healthcare workers [73, 119-122]. Practitioners such as infection control departments of major hospitals have made tremendous efforts to implement the intervention measures according to the recommendations from the literature and

guidelines set by the World Health Organization (or WHO) [73]. Unfortunately, the implementation of the intervention measures is usually imperfect or suboptimal, due to physical constraints or human factors. To hospital decision makers, there are economic considerations when choosing among different interventions, so that even though some are marketed to be effective in laboratory conditions, information concerning their practical effectiveness under actual ward conditions and, how pre-existing routes of nosocomial transmissions may ultimately impact its efficacy forms the basis for a broadened analysis on costs, benefits and risk.. Therefore, having the ability to make accurate predictions about the effectiveness of such interventions when exercised in under ward conditions can significantly improve public health policies.

For example, much research has advocated that higher adherence rates for hand hygiene and increased cohorting can reduce nosocomial MRSA transmissions [73, 75, 119, 120, 123]. WHO enforces the former with a 5-moment standard for directing Health Care Workers (HCW) to apply Hand Hygiene, which is (1) Before touching patients, (2) before a cleaning/aseptic procedure, (3) after body fluid exposure risk, (4) after touching patients, and (5) after touching patient's surroundings [73]. In reality however, HH adherence rates are generally low, with mean baseline rates ranging from 5% to 89%, representing an overall average of 38.7% [73]. Some reports also show that non-adherence differs among physicians in ICUs compared with internal medicine wards, especially when intensity of patient care is high [73, 124]. Practical reasons often quoted for skipping HH include: insufficient time, wearing gloves, forgetfulness, patient's urgent need or irritation to the skin [73].

Besides HH and contact precautions, patient and staff cohorting are also interventions intended to reduce the likelihood of cross-transmissions by reducing patient and staff contact. For example, MRSA+ patients should be isolated from the rest of the patients through the use of isolation rooms, or in beds within a segregated area if the former were not available [74, 75]. HCWs should be assigned to patients in such a way that they focus on a particular set of patients only instead of tending to multiple patients. In practice, the effectiveness of these strategies is often undermined by the nature of the ward activities, especially when faced with shortage of staff [125].

It is not hard to appreciate the logic of intervention measures such as isolation precautions once evidence exists to demonstrate pathogenic transmission occurs by air or by human contact. However,

to model how the imperfect implementation of these intervention measures impact the resulting disease transmission dynamics, demands a sufficiently precise model of the day-to-day operations in the ward environment in question. While it could be convenient to assume a well-mixed population of patients, visitors, and HCWs, and then model the transmissions to take place stochastically between any pair of human hosts at certain rates that are dependent on the HH adherence rates, but in reality, the actual patterns and frequencies of human contact are greatly altered by factors such as HCWs' roles, the assignments of HCWs to patients, the nature of the activities that HCWs' perform on patients, the ward layout, and not least, even the positioning of the physical objects (such as computers, curtains, furniture, and equipments). These operational factors are specific to each hospital, and even to each ward.

To understand the effects of these operational factors, we conducted a field study at the Surgical Intensive Care Unit (SICU) at the National University Hospital in Singapore¹. The field study was led by Dr. Tay Joc Cing through ROSS Scientific Pte Ltd, with three shifts a day and over a period of two weeks, which included manual and automatic tracking of human movements within the ward, manual observation of the daily routines, and direct interviews with different HCW roles. The results were detailed in the technical report submitted to the hospital, and below is a brief summary.

The SICU comprises 12 single-bedded isolation rooms. Patients are assigned to an isolation room upon admission (which effectively implements patient cohorting), regardless of their MRSA screening test results. All patients are observed to be non-ambulant throughout their hospital stay. Visitors are allowed to visit the patients during two visitation hours; 12:00 Noon to 2:00 PM and 5:00 PM to 8:00 PM. There is a visitor lounge outside the ward where visitors can wait and enter the ward from there. There are 11 distinct HCW designations and 6 shift types; the former includes physicians (such as consultants, medical officers, registrars, physiotherapists and pharmacists), nurses (such as nurse managers, nurse clinicians, staff nurses and assistant nurses), and other supporting staff

¹ This work is a collectively effort of the team at ROSS Scientific, including: Joc Cing Tay, Wei Boon Chan, Jimmy Ong, Irene Teo, Hardiyanto Wibisono, and myself. The project would not have been possible without the support by Prof Paul A. Tambyah at National University Hospital, Singapore. The modeling and experiment results are detailed in the paper Joc Cing Tay, Zaiyi Guo, Wei Boon Chan, Hardiyanto Wibisono, Paul A. Tambyah, "The Effect of Staffing Levels on Hand Hygiene Adherence: New Insights from an Evidence-based Behavioral Model of MRSA Transmission within an SICU," (in Preparation).

(housekeepers and meal porters). This corresponds to a total of 18 to 20 HCWs present in the ward at any point of time during a shift. Each individual HCW visits the ward or patient rooms according to their assigned shift and work routines. In addition to stipulated staff duties, they also perform coordinated activities such as roll call, shift change, ward rounds and coverage of duties. Staff cohorting is practiced through staff nurses where each staff nurse is assigned to one isolation room except when there is insufficient staff, then one staff nurse may be assigned to manage 2 or 3 rooms (or equivalently, patients). For other staff roles such as assistant nurses and physicians, they take care of all the patients (e.g., visit all the patient rooms in arbitrary sequence during ward rounds), so there are no dedicated patient assignments for these staff roles. Existing interventions at the SICU include single-stage screening by nasal swabs, contact precautions, labelling of rooms with MRSA+ patients, hand hygiene through chlorhexidine hand pumps as well as treatment of blood-stream infections due to MRSA. When visiting a patient's room, HCWs are required to perform HH according to the 5-moment guidelines by WHO [73]. There is a customized rule in this SICU that HCWs must perform HH (through alcohol hand rubs) and wear contact precautions (comprising masks, gowns, and gloves) before entering the patient's room. Visitors are encouraged to perform HH or contact precautions before entering the patient rooms.

The field study gives us invaluable insights on the level of details that the healthcare practitioners would require from the computational model and experiments. We summarize them below:

1. *Heterogeneous entity types*: even an approximation of the real-world environment involves vast heterogeneities – pathogens such as bacteria; human hosts such as patients, visitors, staff nurses, assistant nurses, medical officers, registrars, housekeepers; and physical objects such as desks, chairs, beds, doors, equipments, computers.
2. *Complex entity states and behaviors*: each human host behave according to her role and act with goals; these determine her daily activities, schedules, movement routes.
3. *Non-linear entity interactions*: interactions among entities are complex and non-linear; their occurrence may be pre-planned but the details are subject to stochasticity. For example, the ward round activity of nurses and physicians is pre-planned for every day at a fixed time; however, the actual time the ward round starts may differ from day to day, and the actual physical contacts between HCWs and patients are non-deterministic.

4. *Non-uniform interactions in a topologically complex environment:* The ward is compartmentalized into rooms for different purposes. The interactions between entities do not take place uniformly in the ward. The density and the nature of interactions are driven by the roles and goals of the interacting human hosts. For example, many HCW-patient interactions take place in the isolation rooms at patients' beds, but a lot less in the central area and corridors.
5. *Multi-scale in spatial and temporal domains:* the multi-scale nature of the system can be seen in both temporal and physical dimensions. In the temporal dimension, human movements are at the scale of seconds to minutes, and their interactions vary from minutes to hours. The natural history of diseases evolves over days, so is the time needed for screening tests. To evaluate the effectiveness of intervention measures, human studies can last from weeks to months, or even longer. In the physical dimension, entities vary greatly in their physical size and relative quantities. The physical size of pathogens (e.g., MRSA bacteria) is about $0.6\ \mu\text{m}$ [126]. They are also numerous and distribute unevenly at various sites on the human body [127].
6. *Demands for great level of details:* hospital decision makers are concerned with a lot of practical and operational details. For example, during a patient-tending activity, the number and the type of HCW-patient contacts, and hence the resulting number of moments where HH should be applied are dependent on the nature of the activity itself. This would mean that if the model is not detailed, it would be impossible to regenerate the correct HH compliance (or non-compliance) behavior, and hence would not be able to correctly answer the question on how HH compliance might impact nosocomial transmission. Therefore models are often iteratively revised with more and more details obtained from feedbacks from the domain experts, up to a point when the amount of detail is sufficient to represent aspects of the hypothesis, a philosophy known as Occam's Razor [41].

It can be seen that the application of nosocomial infections shares the same set of modeling requirements imposed by the application of immune system with HIV infections, presented in Section 2.1. In the next Section, we review the modeling techniques with respect to their capabilities in handling these challenges.

2.2. Review of modeling techniques

In this Section, we review the two dominant approaches that have been frequently adopted for computational simulation models: the Differential Equation (or DE) based models, and Agent-Based Models (or ABMs). The purpose of the comparison is to study their suitability in building incrementally revised computational simulation models for computer experiments.

2.2.1. A comparative study on DE models and ABMs

DE Models are based on a set of differential equations (Ordinary Differential Equations or ODEs, and Partial Differential Equations or PDEs). Each equation describes the change of a variable, such as the size of a population or a sub-population, with respect to time and space. In theory, one could solve the equations analytically. For example, one could analyze the stability properties of the system represented by the equations. For complex models, translating the equations into a computer simulation is also straightforward: the simulation is advanced by a small dt each time, and the changes of a variable (e.g., the population size) are computed and applied back to the variable. This numerical approach has been widely used and is the basic technique introduced in biological system modeling textbooks [128].

The distinctive difference between ABMs and DE Models is that the former explicitly models the individual entities and their interaction events. ABMs-based computer simulations carry out individual-level interactions in order to reveal system-level dynamics. They are usually computationally intensive due to the large number of individuals and events to be modeled and simulated but with advances in microprocessor technologies, they have become more practical [129, 130].

We propose three model characteristics as the basis of comparing DE Models and ABMs: *Model Scope*, *Entity Representation*, and *Behavior Descriptions*. They use the notions of *entity types*, *entity instances*, and *behavioral rules*, which have been defined in Definitions 1-9, 1-4, and 1-5. For ease of discussion, we will use HIV pathogenesis application and their models as examples for illustration.

2.2.1.1. Model scope

As defined in Definition 1-9, the *model scope* of a computational simulation model is the set of *entity types* (ETs) present in the model. Naturally, the more entity types a model has, the more behavioral rules it has to specify their interactions. HIV pathogenesis involves a wide range of entity types and behavioral rules (see Section 2.1), but typically, only a subset is modeled. With DE Models [131-133], we observe the tendency to restrict the Model Scope to a minimum (since in general, the more equations, the more difficult it is to analyze the model). Typically they restrict the scope to {TH, HIV}, i.e., the T helper cells and HIV virions. They omit the other essential entities of the immune system, such as TK cells, B cells, MPHs, and antibodies. In reality, TH cells themselves do not eliminate pathogens but secrete interleukins to activate other cells (such as TK cells, B cells, macrophages) that in turn mount direct humoral attacks on the HIV virions and cellular attacks on HIV-infected cells. The absence of entities other than TH cells necessarily implies that the humoral or cellular immune responses are not modeled, hence intentionally or unintentionally reducing the actual capabilities of the immune system to a Target-Cell-Limited one. Target-Cell-Limited theory [59, 131] advocates that the sharp decline of viral load during the initial acute phase is not necessarily due to the existence of immune response; the decline can possibly be attributed to the non-availability of target cells (e.g., TH cells). Without enough victims to infect, the number of new virions that can be produced are reduced, resulting in the observed decline of the overall viral load.

$$\frac{dV}{dt} = pI - cV \quad \text{Eq. 2-1}$$

where

- I : Infected T cell population size
- V : HIV virion population size
- p : Virion production rate (per infected T cell)
- c : Virion clearance rate (per virion)

Target-Cell-Limited models typically describe the change of the HIV population compartment with an equation similar to Eq. 2-1. The loss of HIV (defined with the term $-cV$) is only due to the natural loss of virions whose rate is proportion to the current population size of HIV, but it is not related to the existence of immune responses (e.g., the population sizes of B cells or TK cells, or their activation levels). How HIV replenishes its population size is through the production of virions from the infected TH cells (the term pI). The fewer infected TH cells available, the slower the new virions are produced. With low levels of infected TH cells and high levels of viral load (which is the case

during the acute phase), pI is small and cV is large, hence a decline in viral load (negative dV/dt) would not be surprising.

Phillips' work [131] attempted to verify the Target-Cell-Limited theory that viral decline can be independently obtained without an immune response. With this specific purpose, it was necessary to restrict the Model Scope to TH cells and HIV virions *only* so as to ensure HIV were not eliminated due to immune responses. Though the simulation result confirmed the qualitative pattern of the peak in the viral load and subsequent declines, it was not quantitatively verified with the observations. In a later study, Stafford et al [134] quantitatively verified the Target-Cell-Limited theory using patient data. The parameter values that were determined through regression analysis consistently predicted a viral load that was higher than the actual measurement after about 100 days of infection. The authors attributed this difference between prediction and reality to the existence of an immune response, leading to an extension of their Model Scope to include 1) TK cells which increased the death rate of infected TH cells; and 2) soluble factors secreted by TK cells that inhibited the production of virion particles from infected TH cells. These studies suggest that those DE models which conveniently restrict the Model Scope by omitting essential entities in the immune response risk generating biased results (e.g., overestimation or underestimation of rate parameters) as compared to reality. This process shows that models need to be incrementally improved to account for discrepancies, such as the widening of model scope.

We observe that ABMs literature process much wider model scope than simply $\{TH, HIV\}$. Grilo et al. [82, 135] implemented an ABM on a Cellular Automata (or CA). Their model included HIV virions, TH cells, TK cells, MPHs, B cells, antibodies and a series of molecules such as IL-1, IL-2, IL-5 and γ -IFN. A similar coverage of cells and chemical molecules by Sieburg et al. [136] was based on a strict CA structure, and by Jacob et al. [137] which modeled cell movements in a continuous three dimensional space. This is in contrast to the DE models, where TK cells, B cells and antibodies are usually omitted and chemical molecules are not explicitly identified or included.

The reason that ABMs tend to have wider scopes can be explained by the cognitive process of designing an agent's behavior (typically using IF-THEN rules). These rules only require descriptions of the biological function of individual entities (such as those of a cell). On the contrary,

constructing mathematical equations requires a further translation of these rules into Population Level relationships such as birth-death flows [128]. The facility of directly mapping from biological entity functions to individual agent rules, we believe, results in models with greater detail and wider scope [138, 139].

2.2.1.2. *Entity representation*

An entity instance is a collection of state variables, behavioral rules, and constant parameters, which is a subset of that of the model (see Definition 1-4). We define *entity representation* based on the notion of entity instance:

Definition 2-11

Entity representation refers to how entity instances are implemented in the model.

We identify three types of entity representations from literature:

1. *Parameter Level*: Entities here are not explicitly represented; only their effects are present in the form of parameters within the dynamical description. For example, the presence of a drug can be modeled as an efficacy parameter that modifies the rate of the infection. For a Parameter Level entity instance, its state variables are constants (i.e., they do not change over simulation time t), and only one entity instance of its type needs to be present in the model.
2. *Population Level or Subpopulation Level*: Entities here are treated as a group, or a population. Population can be subdivided into Subpopulations. What matters in the model is the sizes of the population (or subpopulations) and their changes over space and time. The state variables of an entity instance at the Population Level are the population size, or the subpopulation sizes. Take the Target-Cell-Limited model as an example. The TH cell entity instance is represented by two subpopulation sizes, or state variables: the healthy subpopulation size, and the infected subpopulation size. The HIV entity instance is represented by one state variable, that is, the HIV population size. For Population Level representation, only one entity instance of its type needs to be present in the model.
3. *Individual Level*: Entities here are modeled as distinct and stateful individuals. For example, a TH cell can be modeled as an individual described with state variables such as its activation

status, the specific T-Cell Receptor and its genetic sequence, and whether it has been infected by HIV. For Individual Level representation, multiple entity instances may be present in the model of its type.

For entities modeled at the *Parameter Level*, their existence is reflected as parameters and can be either a constant or dependent on other factors in the model. For example, drug models [78, 140] use a parameter called *efficacy* to model the existence of a drug. In the case of Reverse Transcriptase Inhibitor, the efficacy parameter specifies the proportion of infections that are blocked; while with a Protease Inhibitor, the efficacy parameter specifies the proportion of virions produced from infected cells that are noninfectious.

DE or Compartment Models [59, 76, 78, 79, 131-134, 141-143] typically specify the main entities of interest, particularly TH cell and HIV virions at the *Population Level*. The Target Cell Limited Model by Phillips [131] contains one population of HIV and three subpopulations of TH cell types. They are: healthy, productively infected (which produce HIV virions upon death), and latently infected (which do not produce HIV virions) cells. A therapy model, such as the one proposed by Perelson et al. [78, 143] divides HIV into two subpopulations: infectious and noninfectious. The noninfectious subpopulation corresponds to those inhibited by the drug known as a Protease Inhibitor. The number of subpopulations (and thus, the number of equations) increases with the number of heterogeneous states an entity could have. For example, Essunger and Perelson [144] investigated the virulence of HIV towards memory TH cells. To achieve this, the model divided the TH cell population into subpopulations of naïve, activated, and memory TH cells, and each of these had a corresponding infected subpopulation. The infected memory TH cells were further divided into transiently (revertible to a healthy state) and permanently infected, resulting in a total of 7 TH cell subpopulations and one HIV population. In another example, Wein et al. [140] models the drug effect with 10 subpopulations, out of which 4 are subpopulations of virions: infections & susceptible to drug, infectious & resistant to drug, noninfectious and susceptible to drug, and noninfectious & resistant to drug. The increased number of subpopulations gives rise to more complex coupling between the equations to represent interactions among subpopulations.

Entities modeled at the *Individual Level* are typically found in ABMs. Instead of having to partition

a population into many subpopulations, the implementation of different states is more intuitive here: an agent, based on its type, can transit among a set of possible states over time. For example, in the model by Sieburg et al. [136], there are 8 distinct states for a TH cell, arising from all the possible combinations using 2 state variables: the activation status (with 4 possible values: naïve, secretory, proliferating, memory), and healthy status (with 2 possible values: healthy, or infected). Other entities in the model are modeled with one state variable each, hence the number of states is equal to the number of different values that state variable can take. The model also specifies 4 states for a TK cell (naïve, secretory, proliferating, memory), 5 states for a B cell (naïve, antigen-presenting, activated, antibody-producing, and memory), and 3 states for a macrophage (non-binding, antigen-presenting, and stimulated). On top of this, 3 types of interleukins (IL-1, IL-2, IL-5) and γ -Interferon are also modeled. This is a lot more complicated and detailed than DE models as previously described. The more detailed entity modeling in ABM can be explained by the easiness of adding additional states. In the example above, only 2 state variables are needed for a TH cell entity instance to represent 8 distinct states; to achieve the same, a DE model needs to specify 8 subpopulation sizes (i.e., 8 state variables). This is a desired property when a model needs to be incrementally revised with more details, possibly with more entity states.

2.2.1.3. Behavior descriptions

Entity instances have behavioral rules which update their state variables over simulation time (see Definition 1-4). We define *Behavior Description* based on this notion:

Definition 2-12

Behavior Descriptions refer to the actual implementation of behavioral rules, namely, the logic that updates the entity state variables.

The behavior descriptions directly depend on the model's choice of entity representation. DE Models and ABMs adopt considerably different approaches for behavior descriptions. For DE Models [59, 76, 78, 79, 131-134, 141-143], since they use the population level of entity representation, they use a set of coupled ordinary or partial differential equations that quantitatively define the changes to the population sizes (i.e., the state variables). We refer this as the equation-based approach. For ABMs [82, 135-137], since the entities are represented at the Individual Level and

each has its own set of state variables, it is natural to describe the behaviors with set of IF-THEN rules, which we refer to as agent-rule-based approach. The IF part of the rule describes the conditions under which the rule will be fired, and the THEN part describes the changes that need to be applied to the state variables. The differences between the equation-based approach and the agent-rule-based approach are compared in Table 2-2. Next we elaborate their differences with two examples.

Table 2-2 Comparison of equation- and behavior-rule-based interaction descriptions

	Equation-based	Agent-rule-based
Defined by	Liner combination of products of population sizes weighted by interaction rates	Agent rules: IF some conditions are met, THEN take certain action(s)
Concerned with	System Level Observables	Individual States
Level of view	Macroscopic	Microscopic
Representation	<i>Implicit</i> representation Behaviors are embedded as parameters in the equations	<i>Explicit</i> representation: Behaviors are modeled as actual events happening in the simulation

Table 2-3 Equations and schematic diagram of a simple equation-based model [78]

$\frac{dT}{dt} = \lambda - dT - kVT \quad (1)$ $\frac{dI}{dt} = kVT - \delta I \quad (2)$ $\frac{dV}{dt} = pI - cV \quad (3)$	<p>Where:</p> <p>T : Healthy T cell population size I : Infected T cell population size V : HIV virion population size λ : T cell production rate d : Per cell death rate of healthy T cell k : Infection rate (per virion-cell encounter) δ : Per cell death rate of infected T cell p : Virion production rate (per infected T cell) c : Virion clearance rate (per virion)</p>
---	---

Table 2-3 shows the equations from a simple ODE model by Perelson [78]. This model involves three subpopulation sizes, or state variables: healthy TH cells (T), infected TH cells (I) and HIV virions (V). Each equation relates the rate of change for one population size to a linear combination of a few factors. For example, the rate of change of healthy TH cell subpopulation:

- Increases due to production: λ ,

- Decreases due to natural death: dT . The number of dying healthy TH cells per unit time is proportional to the number of currently living healthy TH cells (T) and,
- Decreases due to infection by virions: kVT . The number of infected cells per unit time, which will then join the infected TH cell population (I), is proportional to the current living healthy TH cells (T), and the current number of virions (V).

The schematic diagram in Table 2-3 reveals the coupling between equations. Each population is represented by a rectangular box, with solid arrows representing the incoming flows (factors that increase the population size) and outgoing flows (factors that decrease the population size), and dotted arrows representing the dependencies between variables. The viral infection behavior is *implicitly* embedded in the term kVT , which is the rate at which the healthy TH cells (T) convert to infected TH cells (I), shown as a flow from T to I in the schematic diagram. It can be seen that the microscopic process of how virions infect TH cells (the microscopic details) is not explicitly modeled since the equation-based models are only concerned with the *macroscopic observables*. By aggregating the microscopic behaviors into macroscopic interaction rates, they are further assumed to be constant, often so that the system of ODEs can be solved.

We next exemplify how an ABM uses agent-rules to describe interactions. Each individual, or agent, has a set of rules which governs their own update behaviors and the interaction behaviors with other agents. Figure 2-5 gives examples of three rules from our preliminary work on modeling HIV pathogenesis [145]. The three rules describe antigen matching, infection and death due to cytotoxic effect for TH cells. From the corresponding schematic diagram, incoming arrows denote the conditions of the rule. If the antecedent is satisfied, then the rule is fired (shown as outgoing arrows). Arrows that point back to the agent indicates that the action part of the rule changes the state of the agent. Agent rules can be specified from a description of a single cell's (or a molecule's) biological function and the modeler need not explicitly enforce any Population Level relationships, such as the flow rates between subpopulations, commonly seen in the DE models. While rule specification is done at a microscopic level here, macroscopic level properties, such as changes in population sizes, are emergent properties as a result of the large collection of local, individual-level interactions.

Table 2-4 TH cell agent rules and schematic diagrams [118]

<p>IF the receptor matches antigen THEN reproduce send out Activation Signal</p> <p>IF reception of Infection Signal (from HIV) THEN change state to Infected send out Successful Infection Signal</p> <p>IF reception of Death Signal (from TK) THEN Die</p>	<pre> graph LR MA[Match Antigen] --> TH1[TH] TH1 --> R[Reproduce] TH1 --> AS[Activation Signal] IS[Infection Signal] --> TH2[TH] TH2 --> I[Infected] TH2 --> SI[Successful Infection] DS[Death Signal] --> ITH[Infected TH] ITH --> D[Die] </pre>
--	---

2.2.2. Choice of modeling techniques for computer experiments

With a detailed comparison between the DE Models and ABMs, we further discuss their suitability for computer-based experimentations in epidemiological applications. We will focus on the modeling requirements highlighted in Section 2.1 and 2.1.2, which are:

1. The system contains a wide range of heterogeneous entity types;
2. Entities have complex states and behaviors;
3. The interactions among the entities are non-linear;
4. Interactions are non-uniform, taking place in a topologically complex environment;
5. The system is multi-scale in temporal and physical dimensions
6. In order to implement the experiment manipulations, the model needs to contain sufficient level of details.

Entity heterogeneity can be naturally handled with ABMs. Each entity in an ABM could have many and even infinity number of states, represented by a set of categorical or real-valued state variables. For example, a TH cell's activation status can have following categorical values: quiescent, activated, productively infected, latently infected, or memory. The amount of surface and intracellular molecules of a TH cell can be represented as quantity values, which is an example of real-valued state variables. For DE models, to account for different entity states, they need to divide the population into more subpopulations (one per state) and each needs one differential equation to describe its change over time or space. Adding a new dimension to the states, for example, drug effects on the HIV virions, would easily double the number of required equations [77, 143]. As a result, DE models generally assume homogeneity of entity types so as to limit the number of computable states

while compromising on the realism of their predictions. ABMs, on the other hand, can afford many entity types and entity states by specifying the required state variables at individual level, without significantly affecting computational tractability.

In epidemiological studies, the entities such as biological cells and human beings, usually interact based on physical contacts within spatial proximities. Entities involved in epidemiological studies are complex, stateful objects whose movement behaviors are often guided, for example, biological cells moving towards higher concentration of certain molecules, or even purposeful, for example, when nurses go to certain ward areas to accomplish their pre-assigned duties. As a result, entities and their interactions are not uniformly distributed in space. This spatial proximity information is typically absent in the ODE Models, which implies the assumption that entity interactions are uniformly occurring at the same frequency at all places. We believe that this assumption is oversimplified and is not sufficient for epidemiological studies. To consider the spatial non-uniformity using equation-based description, PDE models can be used: they specify the dynamics with respect to x , y and z -axes in space in addition to the time dimension. This results in an increased number of coupled equations, making the model computationally even more expensive. ABMs can handle the non-uniformity of entity distributions and interactions by explicitly assigning spatial information (such as the coordinates) to each individual entity as a state variable, and contact-based interactions can be modeled via behavioral rules that are triggered by the spatial proximity. The non-uniform distribution of entities naturally emerges from the individual's movement rules, saving modelers the trouble of explicitly defining the distribution formula on the system level.

Modeling the multi-scale nature of the epidemiological applications is a challenge for both DE models and ABMs. Since DE models are only concerned with the system level changes in population sizes, entity movement on a smaller scale in physical dimensions and the temporal events taking place at that scale are lost. The method of simulating DE model is the same as solving DEs numerically, namely, by dividing the space and time into infinitely smaller steps. However, this pure mathematical manipulation does not enhance any microscopic, or smaller scale insights to the DE models. ABMs, on the other hand, are built on individual entities. The system-level dynamics observed on a larger scale are emergent by simulating numerous smaller scale events over time.

Even though ABMs are able to mirror the multi-scale nature of the application through this bottom-up approach, they still face many practical implementation issues. For example, it would be computationally expensive (and even infeasible) to actually simulate the *numerous* small-scale events (such as the molecular interactions at the timescale of seconds to minutes) in order to produce the larger-scale dynamics (such as population size changes over years). We will elaborate further on this challenge in the preliminary feasibility study in Section 2.3, and propose possible solutions in Chapter 4 and 5 in an Agent-Based modeling framework for computer experiments.

Entities in ABMs are modeled at the *Individual Level*, which are more detailed than DE models by construction. This makes ABMs good candidates for experimental manipulations. However, the modeler still faces the challenge of deciding the appropriate entity granularity (see Definition 1-10) and model scope (see Definition 1-9) to reach a balance between the experimental requirements and the computational tractability. A high entity granularity or wider model scope implies more state variables which generally implies a more accurate and complete representation of its real life counterpart. However, it is also associated with an increased computational cost, since it requires more memory for storage, and more agent rules that operate on it. For example, the following sets for state variables are possible for a TH cell in an ABM, in order of increased entity granularity:

1. A single Boolean state variable `activated`, which is either `true` or `false`. An activation rule would simply update this state variable from `false` to `true`.
2. A single categorical state variable `activation_status`, which can take values such as `naïve`, `activated`, `effector`, and `memory`. An activation rule would need to change the state from `naïve` to `activated`, and plan how long this state will persist until it is changed to `effector` or `memory`.
3. A set of real-valued variables that track the surface and intracellular molecular levels, such as TCR, CD4, CD28, CTLA4, IL-2, and IL-2 receptors [45], which characterize the activation status of a TH cell,. An activation rule would necessarily detect that the binding of these molecules to their ligands on the contacted APCs (e.g., B7.1 and B7.2 molecules) is strong enough for activation, which is practically a series of upgrades and downgrades of the surface and intracellular molecular expression levels.

For computational tractability, modelers usually keep the entity representation at a possibly less granular level until there is an experimental requirement to extend it to more granular representations. Thus we face the issue of *entity extensibility*, which we define as follows:

Definition 2-13

In an ABM, we refer *model extensibility* refers to the property that a model can be revised with wider scope and higher entity granularities without the need to entirely reconstruct the model.

The extension of entity granularity involves not only the change in the state variables, but also the associated agent behavior rules that operate on them. This may result in an uneven granular ABM, where certain types of entities are more granular than others. Since this is a common task for computer experiments, it should be addressed formally. We will propose a systematic approach for iterative model extension in Chapter 6.

Based on the literatures review and discussions above, we believe that ABMs provide a suitable modeling technique for computer experiments in epidemiological applications. However, the ABM applications in the literature [82, 137, 146-148] are mainly for demonstrative or educational purposes, and we observe that their modeling processes vary greatly (such as the choice of model scope, entity granularity, and space representation) and lack a consistent and systematic methodology. To thoroughly understand the practical issues faced by ABMs and their implications on computer experiments, we conduct our own preliminary feasibility study with the application of HIV pathogenesis, which is presented in the next Section.

2.3. A preliminary experiment with HIV pathogenesis hypotheses

In this Section, we present an application of the agent-based modeling approach to elucidate its efficacy for computer experiments. The scientific question underlying this experiment is: given the various pathogenesis mechanisms that have been advocated to be responsible for inevitable TH cell loss in HIV infection (see Section 2.1), we would like to know which mechanism, or a set of mechanisms, are sufficient in causing AIDS. This computer experiment mimics a negative control experiment, and the steps are:

1. To construct a *null computational simulation model*, or the *null model*, that captures the basic model entities and their interaction rules of an immune system, and the way that HIV infects TH cells. It represents of the real world system and serves as the basis for further experimental manipulations. The model is tuned such that it produces a negative result (namely, the AIDS does not develop), hence it serves as a negative control.
2. To extend the *null model* by implementing the hypothesis-specific entities and interaction rules. Differences in the resultant system dynamics (in particular, the AIDS develops) can thus be attributed to the hypothesis-specific modifications.

The purpose of this preliminary study is two-fold:

1. To test the feasibility of using ABMs as the computational simulation models for computer experiments. We want to use ABMs to test if the *emergence* [149-151] of high-level, HIV pathogenesis dynamics can arise out of the simulation of many individualized, microscopic interactions. In the process, we can determine if the null model can be altered to achieve the observed results by manipulating values of parameters, to indicate the level of granularity that is needed.
2. To identify the practical limitations and challenges of ABMs in terms of building credible and extensible null models that are required by the epidemiological applications.

2.3.1. A null model of immune system and HIV infections

The scope of the null model includes following entities: HIV virions, B cells (B) Antibodies (AB), T helper cells (TH), cytotoxic T cells (or CTL; also known as T killer cells, or TK).

Figure 2-4 Agent behavioural rules for the null HIV infection model

Figure 2-4 depicts the individual agent rules for each type of entity in the model. We devise a simple notation: the rectangular boxes represent individual agents, and the incoming edge describes the conditions of the rule. If the condition is met, then the rule is fired (that is, the decision part of the rule is carried out), and its effect is shown with the outgoing arrows. The outgoing arrows pointing back to the agent indicate that the agent's state changes as a result of rule firing. For example, an infection event takes place in the following order. The first rule of HIV (Figure 2-4, rule [1]) says that an HIV virion continuously (at every simulation time step) sends out an "Infection Signal" to its local environment (to be elaborated later), which does not require any specific conditions to be met. The second rule of TH (Figure 2-4, rule [5]) says that if a TH cell receives an "Infection Signal" from local environment, it changes its own state to "infected", and sends out the "Successful Infection" signal. Upon receiving this signal, the HIV virion is in a situation that satisfies the condition of the third rule (Figure 2-4, rule [3]), so that the virion starts to reproduce itself. Other events, such as TH cell activation, antibody secretion, and infected cell killing can be interpreted in a similar way. Antigen matching is implemented as follows. Each HIV virion has a specific physical shape, which is computationally represented by binary string based state variable. A TH cell or B cell successfully

detects the antigen only when its receptor (also implemented via a binary string based state variable) matches the physical shape of the HIV. The diagrams are further summarized and simplified in Figure 2-5.

Figure 2-5 Schematic diagram of the null model

To model the spatial proximity of the agents, the environment is partitioned into a 4x4 grid structure (see Figure 2-6). Each agent resides in one of the grid cells at a time. The signals (e.g., fatal signal, death signal, activation signal, infection signal) are sent and received within the same grid cell. Hence, only agents that are close to each other (within the same grid cell) can interact with each other. There is no additional spatial structure within a grid cell. As shown in Figure 2-6, the interaction topology of the entities (agents) within a grid cell is a complete graph. At each time step of the simulation, an agent can either change to a neighboring grid cell, or remain in the same grid cell, mimicking the effect of cell movements.

Figure 2-6 Modeling space with a 4x4 grid structure

The pseudo code in Listing 2-1 illustrates how the simulation is run. The simulation is controlled by a main loop: for each iteration, simulation time is advanced by 1 (arbitrary time unit) and the rules are fired for each agent in the system, and the visualizations are updated to reflect the current status of the simulation. The simulation can be stopped at any time through a control panel, once the trend of the

population dynamics is unchanging.

```

Advance time by 1
  For each agent in the system
    Go through the rules in its signal reception queue to decide which
    rules to fire;
    Change own state, if applicable
    If 'send signal'
      Get all the agents in the same swarm and add the signal to their signal
      reception queue
    If 'reproduce':
      Create a clone of self and insert into the same swarm
    If 'die':
      Change state to 'dead' and remove self from the swarm
    If still alive:
      Move to neighbouring swarm with a probability p
  Repeat for all agents
  Update the spatial visualization
  Update the population dynamics visualization
Loop

```

Listing 2-1 Pseudo code of the simulation

Table 2-5 Null Model Parameter Setting

Entity Type	TH	HIV	B	AB	CTL
Initial Count	100	50	120	5	30
Production Rate	5	1*	4	10**	3

* reproduction rate only upon successful infection

** production rate per activated B cell

In this preliminary study, we experiment with a small number of agents, configured to be proportional to the entire population size as follows: a healthy individual has on average TH cells measured at 1,000/ μ l, and when it falls below 200/ μ l, AIDS is considered to have developed [65]. In the simulation model, the initial population size of TH cell is set to 100, and when the cell count drops to less than 20, the immune system will be considered weak and susceptible to opportunistic pathogenesis. The parameter values are summarized in Figure 2-5.

Though the model size (namely, the number of entities) is drastically smaller than reality, the simulations of this null model consistently produce the qualitative dynamics that the HIV level is

contained and the AIDS does not develop. This is in spite of the various stochastic factors present in the model, such as the random movements and encounters among entities, the rate that the activated B cells produce the antibodies, and the varied lifespans of each entity. To observe the causal relations between entity interactions and the resultant population-level dynamics, we show the resultant population dynamics of a single simulation run in Figure 2-7. HIV is introduced into the system at $t=0$ with a count of 50. Following the injection of HIV virions, TH count drops, B cell count and antibody count increase. This is a result of the local interactions of HIV infections and B cell activation (by the living TH cells). The HIV count then quickly drops with the fast production of specific antibodies. Similar patterns are observed for subsequent appearance of HIV ($t=35, 53, 70, 90$): HIV does not have a chance persistent and progress to AIDS in the face of effective immune responses.

Figure 2-7 Simulation result of the null HIV infection model

2.3.2. Hypothesis verification experiments

We will implement four experiments to compare against the null hypothesis; namely, Direct Effect on CD4+ cell [65], Rapid Viral Mutation [62], Syncytium Formation [100] and Filling of CD4+ Receptor sites [96] (these have been briefly introduced them in Section 2.1). These hypotheses are chosen because of extensive research supporting each and predominantly wide acceptance by HIV-1 researchers.

Figure 2-8 shows the implementation of the four hypotheses, each of which is an modification of the null model:

1. *Direct Effect on CD4+ Cell* (Figure 2-8a): This hypothesis is essentially the same as the Null Model in terms of agent rule specification. Two parameters are tuned in the experiment to represent different pathogenic strengths: 1) HIV replication rate and 2) the initial HIV virion count.

2. *Rapid Viral Mutation* (Figure 2-8b): The mutation mechanism is added by altering the HIV's shape state variable at randomly chosen time points, computationally implemented as toggling a series of binary bits in the binary string.
3. *Syncytium Formation* (Figure 2-8c): This experiment model adds syncytium formation events, which take place when an infected TH cell gets close to an uninfected TH cell, or an uninfected TH cell gets close to an existing syncytium and is conjoined to it.
4. *Filling of CD4 Receptor Sites* (Figure 2-8d): This experiment model models the site filling of a TH cell by reducing the activation signals from TH cells to B cells. The reduction is implemented on a probabilistic basis, that is, only N% of the signals are received compared to the null model.

Figure 2-8 Schematics of four HIV pathogenesis hypotheses

Figure 2-9 shows the representative single-run population dynamics for each hypothesis.

1. *Direct Effect on CD4+ Cell* (Figure 2-9a): This hypothesis consistently fails to produce AIDS-like population dynamics in various parameter settings, in particular, the HIV population is contained and the exponential viral growth at the AIDS stage is not observed.
2. *Rapid Viral Mutation* (Figure 2-9b): The result resembles a relatively long chronic stage where the HIV virion count remains low for an extended period, and the final stage where the HIV count increases exponentially while the T_H cell count drops below 20, indicating the onset of

AIDS. In another simulation run (results not shown here), the antibody production rate is increased to 3 times that of the Null Model. Despite the high level of antibodies production, HIV still manages to overcome it eventually, further giving support that this hypothesis is a possible etiology for AIDS.

3. *Syncytium Formation* (Figure 2-9c): Syncytium starts to form from the early stages of the infection. Even though the population of syncytium in the system is not large, the dynamics are changed substantially. AIDS-like population dynamics is observed.
4. *Filling of CD4 Receptor Sites* (Figure 2-9d): When N% is set to 70%, the result does not differ significantly from the null model as it does not lead to AIDS eventually. However, when N% is set to 30% (result not shown), the simulations consistently show relative high peaks in virion count, delay in antibody production in response to virion count rises, and occasionally results in HIV outbreak.
- 5.

(a) Direct Effect on CD4+ Cell

(b) Rapid Viral Mutation

(c) Syncytium Formation

(d) Filling of CD4 Receptor Sites (N%=70%)

Figure 2-9 Representative simulation results of hypothesis verification

Based on a negative control study design, if the treatment group (which implements the hypothesis-specific manipulations) shows a positive result (namely, AIDS develops), the positive result can be attributed to the factors that are unique to that group (namely, the hypothesis-specific manipulations). Applying this logic, the simulation results of the computer experiments presented

above would support the speculation that the hypotheses of Rapid Viral Mutation and Syncytium Formation play more vital roles than the other two in causing AIDS.

Through this feasibility study, we have demonstrated process of using ABMs for constructing a null computational simulation model, and conducting a negative control experiment to answer the scientific question on the sufficiency of four HIV pathogenesis hypotheses. This study has also revealed important practical problems of ABMs, which we will discuss in the next Section.

2.3.3. Lessons learnt from the feasibility study

The feasibility study has revealed several pitfalls in modeling that if overlooked, would sacrifice model realism and reduce their usefulness in computer experiments. We elaborate them in detail in this Section. Notably these problems are results derived from our feasibility study and other models similar to ours, where the focus was on mapping the domain knowledge into model details, but with limited thought given to the modeling issues. They are not necessarily generic to all ABMs and there have been research attempts for improvements in these aspects [152-155]. It is our goal in this thesis to coherently address these problems in a unified modeling framework, so as to support credible and extensible ABMs for conducting computer experiments.

1. *Meaningless, arbitrary time units:* The time units (or time steps) in preliminary study do not have temporal interpretations; that is, they do not correspond to real life terms such as hours or days. The same problem appeared for even some well-known ABMs in the literature [44, 82], where the simulation consists of a sequence of time steps (where agent states are updated, or they interact, at each time step). Without a temporal interpretation for each time unit, it is impossible to make quantitative conclusions, such as whether the onset of AIDS occurs within a few weeks, or in tens of years.
2. *Costly implementation to span multiple timescales with constant time-steps.* Assigning temporal meaning to time units has important implications for justifying the update frequencies of the agent interactions. For example, if each time step is assumed to represent 1 day, it would be too infrequent for activities such as cell movements, antigen matching and binding (which should occur at the timescale of seconds to minutes). With the same time unit definition of 1

day, it would be too soon for AIDS to occur in 100 days (Figure 2-9c). An update scheme based on a constant time step is not suitable for multi-timescale nature of the ABM applications.

3. *Improper model size.* In the context of ABMs, *model size* is characterized by the product of two factors: the number of agents present in the model, n , and the rate of interactions (number of interactions per time unit), m , and the required length of the simulation, t . The model size is an indicative measure of the computational complexity of running the simulation, and is bound by $O(mnt)$. For computational tractability, we have restricted the model size to a few hundred cells and molecules (by reducing n), and manually stopped the simulation once the trend of the population dynamics was clear (by reducing t). While such a model size would hardly be representative of the entire immune system, larger-sized models are more computationally expensive. Much research effort have been devoted to make ABMs scalable, namely, to ensure reasonable computational tractability when the model size is increased. Examples include partitioning an agent-based simulation for distributed processing [91, 92, 156-159] and group frequently communicating individuals based on spatial [83, 92] or temporal proximities [91].
4. *Improper entity relative scales.* In the context of ABMs, we use the term *entity relative scale* to refer to two things: the number of agents in relation to others, and the physical sizes of the agents in relation to others. In the preliminary feasibility model, the entity relative scales are not realistic. For example, the number of antibodies (protein molecules) produced during an infection should be several orders higher than the number of cells. An appropriate entity relative scale should account for the multi-scale nature of the model entities by choosing the right representation of entities and their associated interaction description methods, so that the correct relative number and sizes of these entities can be acknowledged with tractable computational resources.
5. *Simplistic environment modeling.* Though the grid-based space model used in the feasibility study provides a platform to simulate the spatial proximity of agents, it is simplistic and may not suffice for many real world applications. This issue has been discussed in various works [81, 146, 160, 161] that in general, the uneven distribution of agents and their interactions have significant effects on the simulation outcomes. Take the human immune system as an example,

it is not a square or cubic container, but a topological connection of several compartments, such as blood vessels, lymph nodes, lymphatic ducts, spleen and tissues. Agents behave differently in terms of movement speed and activation states within different compartments, guided by the compartment's molecular concentrations and flow gradients. An agent does not only interact with other agents; it interacts with the environment too. A realistic environment model needs to account for the topological compartments and the continuous space and flows experienced by the agents.

We have conducted literature reviews to understand specific requirements on modeling techniques in epidemiological applications, and through a comparative study, identified ABMs as the choice of the modeling technique. The preliminary feasibility study for the HIV pathogenesis application is our first attempt to apply the workflow of a computer experiment. Most importantly, it reveals the practical modeling issues of ABMs that may sacrifice model realism and reduce their usefulness in computer experiments. These are insights we had not directly obtained from literatures that focused on the translation of domain details and simulation results [44, 82]. In the next Chapter, we outline the research roadmap that works towards an agent-based modeling framework for computer-based scientific experimentation.

Chapter 3

Research Road Map

Through the literature research, we have clarified the specific requirements for computer experiments by studying two epidemiological applications: the human immune system with HIV-1 pathogenesis, and the nosocomial infections in a hospital ward. In summary, the model used by the computer experiments has to handle a wide range of heterogeneous entity types and their non-uniform interactions. The model also has to address the multi-scale nature of the system in both temporal and physical dimensions, and provide sufficient level of details with easy extensibility. We have identified Agent-Based Models (ABMs) a more suitable modeling technique over the traditional Differential Equation (DE) based models in the face of these requirements. However, through a preliminary computer experiments on HIV pathogenesis hypotheses, we have shown that without a proper modeling methodology, ABMs still face practical issues that compromise the model realism and credibility, in particular: arbitrary time units that have no real life semantics, inefficient handling of events at multiple timescales, improper model sizes and entity relative scales, and topologically inaccurate environment settings. With this background, we are now in a better position to identify the building blocks of an agent-based modeling and simulation framework that supports computer experiments.

Figure 3-1 depicts the roadmap of our work. The ultimate goal is to support computer experiment with a null computational simulation model and one or more experiment models. The former is a representation of the real system and acts as the control group; the latter implement experiment specific changes based on the null model to form the treatment groups. We classify the practical issues learnt from the preliminary study into two categories of desired model properties:

- *Model Realism*: recognizing that models can never be the exact duplicate of the reality, here we use model realism to mean that a model sufficiently approximates the reality so that it produces quantitatively meaningful and verifiable (or falsifiable) simulation results. The practical issues to be addressed under the need of model realism are: to assign realistic temporal semantics to

simulation steps or events, to support the multi-timescale nature of the application, to ensure appropriate model size and entity relative scale, and to model the virtual environment with accurate topologies.

- *Model Extensibility*: this means that models need be easily extensible to incorporate a wider scope and a more detailed level of entity granularities. This is because the initial guess on the model scope and entity granularities is usually imperfect, in that it lacks details for experiment manipulations, or produces simulation results that deviate from the reality. The practical issues to be addressed under the need of model extensibility are: to systematically improve entity granularities, and to iteratively revise models by applying incremental changes.

Figure 3-1 Research roadmap: goals, building blocks, issues, and tasks

The white rectangles in Figure 3-1 show the specific tasks we set out to accomplish in order to build the desired multi-scale agent-based modeling and simulation framework; each solves one or more practical issues. These are:

- *An event-scheduling update scheme* which targets at the semantics and multi-scale issues in time modeling through non-uniform time advances based on events. This will be discussed in Chapter 4.

- *Hybrid entity representation* that targets at bringing down the computational complexity of ABMs yet maintaining the appropriate entity relative scales with a mixed entities representation at both individual and population level. *Layered environment modeling* aims at building a topologically realistic environment model and also providing the supporting structure for the hybrid entity representation. These will be discussed in Chapter 5.
- *Agent behavior network model* targets at a methodology for formal descriptions of agent states and behaviors, based on which the entity granularity can be increased incrementally. *Iterative strategies for model extension* are to provide a systematic approach for model revision in terms of model scope expansion and entity granularity extension. These will be discussed in Chapter 6.

With the framework that supports the construction of credible and extensible ABMs, we will be ready to take on full scale computer experiments. In Chapter 7, we present a computer experiment that addresses the hospital stake holders' concerns on how the staff hand hygiene practices and staffing levels influence the nosocomial disease transmissions. The null model replicates the hospital ward setting as accurately as possible; the experiment models test various hand hygiene adherence and staffing level settings based on the null model to predict whether the nosocomial infection dynamics will be significantly changed. Decision makers can easily relate to the simulation results and use them as references for policy making or devising further studies. Another application in Chapter 8 is to demonstrate the capability of running sophisticated thought experiments on computers. The experiment examines various hypothetical causal intuitions on the quantitative relationships between chemoattractant and cell locomotion in the phenomena of chemotaxis, and rejects those predict contradictory results.

Chapter 4

Modeling Multi-Timescale Events

In this chapter, we focus on the modeling of time in ABMs, through *update schemes*. The simulation run of a computational simulation model is the process of the updating its state variables (*SVs*) by applying behavioral rules (*BRs*) at the correct timing in the correct order. How exactly the updates take place lies in the *update scheme* that the model adopts. As models' *SVs* and *BRs* can often be aggregated into logical entity instances, we define *update scheme* based on the notion of *entity instances*:

Definition 4-1

The *update scheme* of an ABM defines the *frequencies* at which the *model entity instances* update their states, and the *temporal order* of these updates in relation to each other.

Computer experiments require the update scheme to reflect real-life semantics and scales. This means that the time steps should be assigned real-life meanings, events should take place at the correct timescale, and the model scales well from smaller-timescale micro events to larger-timescale emergent system dynamics. We identify two orthogonal requirements for update schemes in the context of computer experiments: *temporal realism* and *computational efficiency*.

Definition 4-2

We consider a simulation to be *computationally efficient* if it optimizes the use of computational resources. One such resource is CPU cycles incurred during unnecessary state update executions.

Definition 4-3

We consider a simulation to have *temporal realism* if its update scheme correctly reflects the *timescales, frequencies, and the temporal order* of different agent activities, without introducing any programming artifacts.

Axtell [146] has illustrated programming artifacts such as spurious synchronization and non-intentional correlations among agent behaviors. Such effects arise from the way the agent state updates are implemented and are not originally intended by the model.

In the preliminary experiment study presented in Section 2.3, the ABMs used a *uniform time-step based update scheme* which is commonly found in the literature [82, 96, 136, 146, 162]. This update scheme has clear limitations in both *temporal realism* and *computational efficiency*. Activities in ABMs can take place at very different *timescales*, possibly ranging from seconds to years. Making updates at uniform time steps would be *temporally unrealistic* if the update frequency is set too low, where fast-changing activities are missed. It would not be *computationally efficient* if the update frequency is set too high for the sake of accommodating the smallest time-scale activities, which potentially wastes execution time in trivial update routines of larger time-scale activities.

In this Chapter, we will survey the different types of update schemes in terms of their realism and computational efficiency. Then we propose an event-scheduling based update scheme which eliminates the use of time-step as the time modeling method under a credible ABM. We enhance the event-scheduling/time-advance algorithm [163], which has been conventionally used in discrete-event simulation, to efficiently handle agent updates as frequent as to be considered continuous (e.g., antibody productions by plasma cells) and as infrequent as taking several months (e.g., the lifespan of a B cell). We demonstrate the kind of model realism and computational efficiency that the proposed update scheme can achieve through two case studies: a model of the B cell life cycle, and a model of the daily work routines of the nurses in an Intensive Care Unit.

4.1. A survey on update schemes

Most research works have studied the issue of update scheme with relatively simple life-like models, such as Cellular Automata [162, 164, 165], Random Boolean Networks [166, 167], and Coupled Map Lattice [168, 169]. These models can be viewed as special cases of ABMs: they comprise multiple individuals (*cells* in Cellular Automata, *nodes* in Random Boolean Network, and *local maps* in Coupled Map Lattice), whose local interactions drive the system dynamics. The realism issue of the

update schemes has been explored in the context of *synchrony* versus *asynchrony*, as well as in the comparisons between the sub-versions of asynchronous schemes [161, 168, 170-172]. *Synchronous update* means that all the agents are updated simultaneously by computing the changes of all entities before applying any changes back to the agents. *Asynchronous update* means that changes for each agent is computed and immediately applied, and that change will be seen by those agents who get updated subsequently. These studies showed that asynchronous updating produced considerably different behaviors from the synchronous version. For example, Huberman and Glanceb [172] showed the effect of synchrony with a Spatial Prisoner's Dilemma experiment: when all players updated their strategies synchronously, the population at 217 generations (updates) contained both cooperators and defectors; but with asynchronous update of strategies, there were only defectors left in the population. This result shows that simulation results can be easily altered with the implementation artifacts like update schemes. Hence justification of the choice of update scheme is essential for ensuring model realism.

As opposed to the frequently discussed topic of synchrony and asynchrony, another important issue with the update scheme has been less discussed in the literature: the capability of handling *heterogeneity in update frequencies*. Simple and abstract models such as Cellular Automata, Random Boolean Network, and Coupled Map Lattice do not explicitly state the time semantic of each update or its analogy to reality, that is, how much time each update actually represents. Some discussions on real applications of ABMs help us to appreciate the sources of heterogeneous timescales and the importance of assigning time semantics to the update schemes. Take the immunological agent activities as an example. Cell surface molecule interactions caused by direct cell-to-cell contact occur at a time scale of seconds to minutes: during the first 60 seconds of T cell – APC contact, the T cell makes its initial activation ‘decision’ to either sustain the interaction or disengage [148]. On the other hand, Cell division takes a few hours. Activated T cells are estimated to divide two to four times every 24 hours, and this process lasts for 3 to 5 days [45]. The lifespan of cells also vary greatly: lymphocytes (T cells and B cells) in a naïve or memory state are usually long-lived (in the order of years); armed T cells and plasma cells (the antibody-producing B cell) are typically short-lived (in the order of days), but there is also a subset of plasma cells that are long-lived (in the order of months to years), which sustains the pathogen-specific antibody levels in bloodstream [45]. The substantial lifespan differences of the target cells of a virus is believed to

cause different phases in disease progression, for example, the acute, latent and AIDS phases of HIV infection [143, 173].

For the ease of comparison and discussion, we propose a classification of the update scheme based on the following criteria:

- *Time Synchrony*: whether the update scheme is *synchronous* or *asynchronous*, and
- *Time Advance*: whether the update scheme is *time-step-driven*, where the simulation time advances *uniformly* by a constant time-step; or *event-driven*, where the simulation time advances by a *variable* interval towards next event.

Table 4-1 Classification of Update Schemes

	Time-step-driven (Uniform time advance)	Event-driven (Variable time advance)
Synchronous	A	-
Asynchronous	B	C

This categorization results in 3 classes of update schemes, as shown in Table 4-1. Synchronous update scheme is necessarily a time-step-driven approach (class A) because it relies on a uniform time-step dt by definition. For asynchronous event updating, either time-step-driven (class B) or event-driven approaches (class C) may be used.

The defining characteristic of a synchronous update scheme (class A) lies in its two-phase update: for each round of updating or a *sweep*, every agent is visited twice in two phases (the terminology follows those by Cornforth, Green, and Newth [174]): the *interaction* phase, where each agent computes the new state and holds it in a temporary store; and the *update* phase, where each agent commits the state change by copying the values from the temporary store to its state variables. The new state of any agent is not available to other agents for revaluation of their states in the current sweep. As a result, the agent visitation order in each time-step does not affect the simulation result. The realism of synchronous updating has been challenged [166, 172, 174]. A common argument against synchronous updating is its requirement of a “global clock”, which is believed to be absent in natural systems unless further evidences prove otherwise [166]. It is worth noting that in ABMs, though global synchrony is usually not justifiable, local synchrony does exist when agents undergo

state changes together as a result of interactions. For example, when an infection event occurs, the cell agent changes its state to *infected*, and the virion agent changes its state to *in the host*. While global synchrony may be artificial and designed, local synchrony is natural and spontaneous.

As opposed to synchronous updating, asynchronous updating (class B and C) of agent states immediately commits the state change after the state evaluation; this means that the agents updated later are affected by those updated earlier. For the asynchronous, time-step driven (class B) update scheme, the choice of the within-sweep agent visitation order has a significant impact on the simulation result. A fixed visitation order often introduces non-intentional correlations among agent behaviors. Schönfish and De Roos [175] examined a spectrum of visitation orders for a two dimensional CA model and demonstrated that the more rigid the order, the more structured patterns appeared as a result. Generally, unless the real world counterpart does exhibit regular visitation orders (e.g., in a power distribution network the effect of any fault will propagate through the system in a fixed order rather than in random order [174]), it is believed that the visitation order should be randomized for each sweep so as to avoid programming artifacts such as spurious synchronization and undesired agent correlations [146].

Figure 4-1 Time-step-driven update scheme: infrequent updates are timed by count down variables

One important problem faced by the time-step-driven update schemes (class A and B) is that the finest achievable time resolution is its time-step dt . This implies that for less frequent updates which require updating intervals to be larger than dt , they can only occur at multiples of dt in the simulation; and if an update is supposed to occur between t and $t + dt$, it has to be delayed until $t + dt$. Setting a large time-step dt causes the smaller timescale updates to be missed, which sacrifices model reality. In the work by Kleinstein and Seiden [44], dt is inferred to be 8 hours, which is too coarse for the

faster-occurring events such as production of antibodies. One natural solution would be setting dt to align with the smallest timescale activities. As such, the larger timescale activities have to wait for many time-steps for their next occurrence, possibly by keeping a timer variable. In each sweep, every agent is visited, during which its state is examined to decide the need of any activity execution at that time point. A larger-scale activity reduced its timer by dt in the sweep; if the timer drops below zero, it implies the long interval has passed and the larger-timescale activity can be executed. Such timer decrement operations are *trivial updates*, as opposed to *nontrivial updates* that actually invoke behavioral rules to change agents' states. Figure 4-1 illustrates the time-step-driven update scheme with agent updates at multi-timescales. The time is discretized, with dt set to the smallest updating interval required by Agent 1. The other three agents have activities that require less frequent updating; their correct timings are represented by the arrows. Most of their update executions are simply to decrement the timer, namely, trivial updates (shown as white dots). The actual timings of the execution of activities (shown black dots) are aligned with time-steps, which lag the desired timing of their executions (arrows). This leads to two observations:

- Trivial update executions exist which waste computational resources (issue of *computational efficiency*), and
- The activities may not occur at the right time point but have to be delayed to align with the time-steps (issue of *realism*).

For asynchronous update schemes without a uniform time-step (class C), it eliminates the problem of a within-sweep agent visitation order, as the absence of a time-step renders the notion of sweeps irrelevant. The simulation can be viewed as a sequence of updating events, executed according to their chronological order, and interspaced by appropriate time intervals. Explicitly specifying time gives several advantages. It models continuity in time, since each interval change can take arbitrary positive real values. This eliminates the problem of delay due to discretization of time as with the time-step-driven asynchronous updating approach. Also, as each agent is allowed to autonomously decide the time interval to its next update, this eliminates the need for a central control of the updating order because the timestamps associated with the next update defines the updating order. This decentralized ordering control is in line with the theme of agent autonomy in ABMs.

Figure 4-2 Event-driven update scheme: continuous time and variable time advances

A general implementation of this scheme is the event-scheduling/time-advance algorithm [163]. We will use the term *event-scheduling update scheme* based on this implementation for the following discussions. This algorithm was designed for discrete-event system simulations, taking advantage of the fact that the system state only changes as discrete events occur. Its applications have been focused on engineering domains, such as simulating supply chains, manufacturing, and queuing systems. A classical example is to determine the number of tellers required to serve customers [176]. In this algorithm, the scheduler is responsible for ordering events chronologically. Figure 4-2 depicts a similar scenario as in Figure 4-1, with a scheduler's view of the overall execution of the activities. Time is advanced to a point where update is required; therefore the update intervals are not uniform. As such, it naturally handles updates with different timescales because the time granularity can be increased to an arbitrary level where necessary.

The event-scheduling based update schemes are potentially more efficient with multi-timescale applications. Assume a pool of N agents, among which the smallest update interval is dt . With a time-step-driven update scheme, each entity needs to be examined to check for applicable behavioral rules and state changes. The number of updates per time step is therefore N , since every agent is visited and updated, whether it simply decrements the timer (a trivial update), or it indeed updates one or more states (a nontrivial update). Let the total simulation time be T , and $n = T / dt$ be the number of time advances required for the simulation. The time complexity of the simulation is thus $Nn = NT/dt$. With event-scheduling based update schemes, only nontrivial updates are executed. Let p to denote the proportion of updates that are nontrivial among all the update executions. Throughout the simulation, the number of nontrivial, hence actually executed updates is pNn . However, there is a computational overhead in maintaining the order of the scheduled events. In our implementation (which will be presented in Section 4.2), the time complexity for re-ordering the events is $O(\log N)$.

Two re-ordering operations occur for each event: at the time it is scheduled, and at the time it is executed or cancelled. Therefore the time complexity of a simulation with event-scheduling update schemes is $pNn(1 + 2O(\log N))$. Compared to the time-step-driven update scheme, the difference is $2pO(\log N) + p - 1$. When p approaches 1, that is, the timescale is not diverse hence the nontrivial updates are few, the event-scheduling scheme does not improve performance but worsens off due to the overhead of event re-ordering operations. The modeler will start to see the benefits of event-scheduling with a small p . This happens with the case of multi-timescale requirement: dt may go very small hence p quickly decreases.

Though conceptually straightforward, the event-scheduling update scheme needs to be adapted to suit ABMs. Heterogeneous timescale activities not only occur for different types of agents; they occur on the multiple “concurrent processes” within each agent, too. For example, a biological cell can be both infected and replicating. A patient can be both treated by the medicines while being contacted by colonized hands of HCWs and developing new colonization sites. An event has two phases: *scheduled*, where it is registered in the scheduler, and *invoked*, where the event is actually executed. An event scheduled by one process may be voided before it is invoked due to state change of the agent caused by another process. This problem is specific to the ABM context and is not tackled by the general event scheduling/time advance algorithm. The event-schedule update scheme we propose in next Section will be tailored for such needs.

4.2. An asynchronous update scheme for multi-timescale simulation

In this section we propose an event-scheduling update scheme designed for multi-timescale ABMs. While the principal idea agrees with the event-scheduling/time-advance algorithm [163], we adapt it with *implicit* and *explicit cancellations* to fit the context of ABM applications. We will describe the two key components for an event-scheduling update scheme: an *event structure* used by the agents which require updates, and the *scheduler* which orders and invokes the update tasks.

4.2.1. Event structure

Figure 4-3 shows the class description of an *event structure*. The *Event* class consists of an *Event Name*, which serves as the callback for the scheduler. It contains a *Parameter List* which is relevant

to its own operations, and the parameter values are supplied at the time the event is scheduled. The event is guarded by a set of *Conditions*. If the event is invoked by the scheduler but fails the condition checking, the event is automatically voided. This leads to the notion of *implicit cancellation* and *explicit cancellation*, which we define below:

Definition 4-4

Implicit cancellation of an event happens when the execution of an event is stopped because it fails to pass the conditions of the event.

Definition 4-5

Explicit cancellation of an event happens when the event is explicitly taken off from the scheduler before it gets executed.

For example, a cell division event is scheduled with the condition that the cell should be in an activated state. Other events taking place before this event may change the cell state to effector (a terminal state where division is not possible). By the time the division event is invoked, the condition checking fails hence no more cell division occurs. If the event succeeds in condition checking, it proceeds with the operations specified in the *Operation List*, and schedules the events in the *Next Event List* with the scheduler. These operations will possibly change the agent's state.

Figure 4-3 Event structure for event-scheduling update scheme

The structure of the *Next Event* is designed based on the following rationale. There are times when more than one event is valid for scheduling at that state, but only one is actually scheduled. For example, an activated B cell may differentiate into either a plasma or memory cell, and because we do not have sufficient knowledge to decide which differentiation a cell will take [45], we may use

data-derived probability values to approximate the likelihood of each differentiation. Therefore, we define the *Next Event* in the form of a stochastic event: it consists of a list of *valid* events (with required parameters): $e_1, e_2, \dots, e_i, \dots, e_n$. Let the probability of e_i happening be p_i , with the time lapse t_i . The time lapse is a random variable with certain probability distribution of values. Upon scheduling, only one event is chosen through roulette-wheel selection, and one value of the time lapse is obtained from the distribution. Listing 4-1 summarizes the processes described above.

```

Invoke (params):
    if (check(conditions) = true) then
        for each operation  $op_i$  in Operation List:
            Execute( $op_i$ )
        endfor
        for each NextEvent  $ne_i$  in Next Event List:
            Event  $e_i = ne_i.selectRandom()$ 
            Time  $t_i = e_i.getTimeLapsed()$ 
            Add ( $e_i, e_i.params$ ) to Scheduler, with waiting time  $t_i$ 
        endfor
    else
        //Do nothing. Event is cancelled (implicit cancellation).
    Endif

```

Listing 4-1 Pseudocode for event handling

4.2.2. Scheduling operations

The operations of the scheduler are summarized in Figure 4-2. They closely follow the generic event-scheduling/time-advance algorithm, used in discrete event simulations [163].

The scheduler keeps the events in chronological order in a data structure known as a *Future Event List* (or FEL). After initialization, the scheduler runs as long as there are events left in the FEL (as the invoking of events may cause more events to be added to the scheduler). A maximal simulation time can be set as an optional termination condition. The new event is scheduled to be executed at waiting time t from the current time. There is a special operation *Cancel All*, which is intended for agents that are dying or permanently leaving. For example, when marked virions undergo phagocytosis or when an individual dies from viral infection, such agents are removed from the environment. All its scheduled events will have to be canceled. This is an example of the *explicit cancellation* defined in Definition 4-5.

```

Initialize:
    Set current time = 0
    Add initial set of events to Future Event List (FEL)

Run:
    While (Future Event List is not empty)
        Get the first event e from FEL
        Set current time = get timestamp of e
        e.invoke (e.params)
    End While

Add (Event e):
    Waiting time t = get waiting time from e
    Add e to FEL with timestamp: current time + t

Cancel All (List l):
    For every event ei in l
        Remove ei from FEL
    End

```

Listing 4-2 Pseudocode for scheduler operations

4.2.3. Implicit and explicit event cancellation

The implementation of event-scheduling update scheme improvises the generic discrete event scheduler by including the *implicit* and *explicit cancellation* operations, which we describe in this Section. As shown in Section 4.2.1, interleaving of agent state changes caused by concurrent activity processes on the same agent may void events that are scheduled but are yet to be invoked. In such cases, the voided events have to be canceled. We employ two ways of canceling a scheduled event, which have defined as *implicit* and *explicit event cancellation* (see Definition 4-4 and Definition 4-5). We explain their rationales in this Section.

Implicit cancellation caters for computational efficiency by eliminating the need to track interdependent events. To find those events that are scheduled and dependent on the current state change would require a mechanism to track interdependent events. This requires deep domain knowledge, and can incur significant computational overheads. By guarding every event with a set of conditions, implicit cancellation happens when the condition checking fails at the time the event is invoked.

Explicit cancellation is required in situations such as agent's death (e.g., a biological cell dies naturally or due to infection) or extinction from the environment (e.g., a patient is discharged). This is straightforward as all events scheduled by the disappearing agent are voided. With potentially great amount of agents since creation, it is important to release the resource occupied by the dead agents in time. It eliminates null pointers problems with C++ implementation¹, or unreleased computational resources with Java implementation²; hence explicitly cancellation is compulsory for both performance and smooth running of the simulation.

4.3. Case study 1: performance comparison with a B cell life cycle model

In this Section, we use a model of the B cell life cycle to illustrate the use of event-scheduling based update scheme and to compare its performance against the uniform time-step approach. B cells are the dominant players in the humoral immune response (part of the adaptive immune response) [45]. Their major role (aided by TH cells) is to differentiate into antibody-producing cells, called plasma cells. Antibodies are simply serum molecules which are the soluble versions of the antigen-specific receptors expressed on the B cell surface. Antibodies can bind, neutralize and mark the antigens for phagocytosis [45].

Figure 4-4 State diagram for a simplified B cell life cycle

We base the model on the B cell state diagram shown in Figure 4-4, which is a much simplified picture of the actual biological phenomenon. A naïve B cell, once matured in the bone marrow,

¹ In C++, if the memory dedicated to an agent is released upon its death, all pointers to this agent become invalid. If an event scheduled by this agent is not immediately cancelled but left untouched in the scheduler as in implicit cancellation, by the time it is invoked it will cause a segmentation error, as the agent it points to no longer exists.

² In Java, if the agent's death does not explicitly cancel the scheduled events, the references to the agent will persist and hence the dead agent cannot be collected by the garbage collector.

enters the bloodstream. The naïve B cell is resting, or inactive, with condensed chromatin of the nucleus and little transcriptional activity [45]. At one stage of the circulation, it enters the lymphoid tissues (examples are lymph nodes, spleen, Payer's patches, tonsils, adenoids, and appendix) where they have a chance to closely interact with antigens and TH cells. If the interactions cause matching and activation, the naïve B cell is trapped in the lymphoid tissue, become activated, and start to divide. If not activated, it passes through the lymphoid tissue, eventually entering the blood circulation, and the whole process is repeated. Activated B cells in lymphoid tissues start to divide through a process known as clonal expansion which lasts for several days [45]. At the end of clonal expansion, they differentiate into either plasma cells (short-lived, in the order from days to months) or memory cells (long-lived, in the order of years). Plasma cells produce large amounts of antibodies at a rate of 2000 antibody molecules per second [177], and eventually die. Memory cells are mostly resting, with a small percentage dividing infrequently [45]. The memory B cell counts are maintained with little alteration for the lifetime of the organism. Here we assume the memory cell is as long-lived as the host, unless its fate is altered by reactivation due to another antigen encounter. Upon reactivation, the memory cell reenters the activated state and divides, which produces more offspring cells that either become plasma cells or memory cells.

Figure 4-5 B cell life cycle event scheduling

Figure 4-5 depicts how the event-scheduling update scheme sees the temporal relationship of the

various B cell life cycle events. The externally triggered events (black dots) and self-scheduled events (white dots) are labeled in the order that they get scheduled onto the scheduler. The chronological order of their actual occurrence is indicated by their relative positions on the timeline. The coexistence of multiple timelines indicates the concurrency of the multiple processes on the same B cell. For succinctness, we will only explain the modeling of B cell generation, death in naïve state, and clonal expansion in this Section. Interested readers may refer to Appendix B for the completion description on the application of the event-scheduling update scheme to the B cell life cycle model.

The discussion below refers to the B cell instance *A* in Figure 4-5. A B cell is produced by the bone marrow with an initial naïve state. This is modeled as the execution of an event `create_from_bone_marrow`, which sets the cell state to `naïve` (label 1). Since a resting naïve B cell does not have much activity until its receptors bind to an antigen, the model only needs to schedule one event for it: `naïve_death` (label 2) with waiting times t_1 . That is, after the scheduling, the scheduler waits for t_1 units of time, and then invokes the `naïve_death` event. The duration parameter t_1 represents the lifespan of the a naïve B cell, which is in the order of weeks to years [45]. If the naïve B cell does not encounter and match any antigens, it naturally dies after t_1 elapses. The `naïve_death` event is conditional on the cell being in the `naïve` state. If the cell changes the state (e.g., becomes activated) in its lifespan, the `naïve_death` event will be implicitly cancelled.

After encountering and matching an antigen (e.g., viron), the B cell goes through the self-scheduled events of endocytosis and antigen presenting (two such occurrences are shown in Figure 4-5: labels 3 – 5, and 6 – 8). It becomes `activated` after an externally triggered event where it encounters some effector TH cell (label *d*). The modeling details of these events are left to Appendix B. The scheduler invokes the event `activation` (label 9) which switches the B cell state to `activated` (and effectively voids the `naïve_death` event). At the end of this event, it schedules the next cell division event (`divide`, label 10), which starts the process of clonal expansion. Clonal expansion usually takes 2-3 days (t_5); then the cell differentiates into either a plasma or memory cell. With the event-scheduling update scheme, the two possible fates are modeled probabilistically as two eligible *next events* of the event `activation` after durations t_5 : `convert_plasma` with probability p_1 , and

`convert_memory` with probability p_2 . The event `divide` is recurring, that is, at the end of each `divide` event, it schedules another `divide` event (label 10, 13-15) with the interval t_4 , which is estimated to be 6-8 hours [45]. The `divide` event is guarded by the condition that the cell state remains in `activated` state. Once either `convert_plasma` or `convert_memory` happens after t_5 , the subsequent `divide` event (label 15) will fail the condition checking hence is implicitly cancelled. This effectively ends the process of clonal expansion.

If the `convert_plasma` event is chosen after t_5 , the cell enters the `plasma` state and starts another recurring event `produce_antibody` (label 16, which further triggers events 18-24). Antibody production is a much more rapid process; this is reflected by its interval t_6 which is set to the order of seconds. The process lasts until the occurrence of the event `plasma_death` (label 17), which implicitly cancels the last `produce_antibody` event (label 24).

Figure 4-6 Scales of time interval parameters (information complied from [45, 177, 178])

Figure 4-6 summarizes the different scales of the time parameters used in the B life cycle model. Different activities of one agent can take place concurrently (such as endocytosis and collisions) and opportunistically. Events are interrelated through scheduling and the common agents that they modify. The B cell has both long lasting idle states (naïve and memory) with little activities, and short-lasting busy states (activated and plasma) with intensive activities.

To compare the computational performances, we also develop a uniform time-step based model for the same B life cycle phenomena. For model realism, the time-step is set to accommodate the smallest timescale activity, which is the rate at which antibody production by plasma cells. The

larger timescale activities are executed at multiples of the time-step. To achieve this, we use timers, implemented using count-down variables, for timing the larger timescale updates. For every time-step dt , the timers that satisfy its own count-down conditions are decremented by dt . If the count-down causes the timer to drop below zero, the operations, which correspond to the larger timescale activities, are executed. For recurring activities, such as cell division, the execution operation also top up the timer for the next occurrence of the activity. This approach has the drawback that the updates can only take place at discrete, evenly-spaced time points, as illustrated in Figure 4-1.

The experiment setup, parameter settings and justifications, and the computational environment are detailed in Appendix B. We highlight some of the experiment setup below:

- The environment is initially populated with 3000 naïve B cells (equivalent to the B cell count in 10 μ l of blood). Initially there are no antibodies.
- From day 1 to day 5, the environment remains free of antigens.
- From day 6 to day 10, a series of antigen binding events are scheduled at a time interval of 0.001 days, which mimics the intrusion of antigens.
- From day 11 to day 13, a total of five B cell activation events are scheduled at a time interval of 0.3 day. This mimics the fact that only a small subpopulation of B cells are capable of antigen binding, hence are activated.
- The population dynamics is sampled at a time interval of 0.2 days.
- The simulation is run for 40 days (simulation time unit)
- The simulation is repeated for 100 times for both update schemes.

Figure 4-7 plots the population dynamics (averaged over 100 simulation runs) of the four B cell subpopulations (naïve, activated, memory, and plasma) and antibodies for both event-scheduling and uniform time-step update schemes. The majority of the naïve B cells remain inactive for the whole simulation. The peak is a result of the clonal expansion of the 5 activated B cells. As expected, activated B cell population appears first, followed by plasma cells and memory cells. The antibody count sharply increases at the time the cell population reaches its peak, reflecting the existence of the large amount of plasma cells. The plasma B cells die off fast (in several days), but the antibodies they produce last longer, whose count drops relatively slowly with a half-life of 22.5 days (Figure 4-7

(c) and (d)). This is agreeable with the half-life of IgG antibody, which is 21-24 days documented in the literature [179]. Eventually only the memory and naïve cells remain in the population.

Figure 4-7 Temporal Population Dynamics (a) B cell, event-scheduling approach; (b) B cell, uniform time-step approach; (c) Antibody, event-scheduling approach; (d) Antibody, uniform time-step approach

Between the two update schemes, their temporal patterns closely resemble each other, confirming that though the update scheme implementations differ, they model essentially an identical set of behaviors, and produce almost identical results. Both simulations resemble the real-life experience where the white blood cell counts increase during immune responses, which can be physically seen by symptoms such as swollen lymph nodes [45]. However, the execution times are drastically different, as shown in Figure 4-8. The event-scheduling based approach is about 40 times faster than the uniform-step based approach, finishing within 24.7 seconds on average.

Figure 4-8 Comparison of total execution time (in seconds)

Figure 4-9 details the savings in the computational time. For $t < \text{day } 10$ and $t > \text{day } 20$, the event-scheduling update scheme spends essentially zero execution time. This is expected because the naive and memory cells do not schedule regular events for execution. Though a series of antigen binding events take place from $t = \text{day } 5$ to day 10, they are computationally relatively inexpensive, hence their effects can be hardly observed. On the other hand, the uniform time-step update scheme spends a considerable amount of execution time for these relatively inactive periods, and the amount of time spent is proportional to the population size. As shown in Figure 4-9(b), the event-scheduling scheme saves execution time by almost 100% during the inactive periods ($t < \text{day } 10$ and $t > \text{day } 20$).

Figure 4-9 (a) Execution time for each monitor interval; (b) Percentage of execution time saved for each sampling interval

From $t = \text{day } 10$ to $\text{day } 20$, both schemes have an increased execution time as a result of increased population size and more intense update activities. During this period, the smallest update interval in the event-scheduling scheme is the same as the time-step in the uniform time-step scheme, which is the update due to antibody production and loss. We therefore expect both schemes to have the same performance in updating the plasma subpopulation. Even so, due to the fact that the B cell population contains other less active subpopulations (activated, memory, naïve cells), the event-scheduling scheme is still superior due to the time saved in those subpopulations, which is estimated to be at least 80% as shown in Figure 4-9(b).

The activities in the activated and plasma subpopulations are more intense. The update routine merely counts down for the naïve cells, while for activated and plasma cells, more substantial activities, such as new cell creations and antibody productions, are taking place. Correspondingly, intensive activities take longer to execute. As can be seen from Figure 4-7(b) and Figure 4-9(a), when the population size is at its peak, the B cell count exceeds the initial value by more than 50%, but for the uniform time-step update scheme, the execution time increases by more than 100%, which implies that the execution time of update tasks of activated and plasma cells is longer than the naïve cells.

Though the setting of this experiment is artificial, the performance improvement with the event-scheduling update scheme is evident. The improvement arises from the avoidance of the trivial updates wasted in the uniform time-step update scheme. According to the computational complexity analysis presented in Section 4.1, with a pool of N agents, and p as the proportion of updates that are nontrivial among all the update executions, the difference between the event-scheduling update scheme and the uniform time-step update scheme per agent per smallest time advance is $2pO(\log N) + p - 1$, and the overall difference $Nn(2pO(\log N) + p - 1)$. The event-scheduling update scheme outperforms when p is very small, which translates to a heterogeneity in the timescales of the simulated events. In particular: 1) the model contains a diverse range of update frequencies due to agent heterogeneity and an uneven model granularity, 2) for the same agent, it requires different update frequencies at different states due to the change of behavior. With a large N , the absolute difference (save in the computational time) is more significant. The conditions of large

number of agents and heterogeneous timescales generally hold true for ABM applications in epidemiology. However, if in an application, the agents are relatively homogenous, and there is little or no potential difference in the required update frequencies, p approaches 1. In those cases, event-scheduling update scheme would show no substantial advantage or even appear to worsen off due to the reordering operations involved in the algorithm.

4.4. Case study 2: patient stays and HCW daily routines in an SICU ward

In this Section we present a model of the multi-timescale human host activities in a Surgical Intensive Care Unit (SICU). This work is as part of a project by ROSS Scientific Pte Ltd in 2008 to 2009 that studies the effect of staffing numbers and hand hygiene adherence levels on nosocomial disease transmissions [180]. The modeling details and experiment results will be presented in Chapter 7. Here we focus on how the proposed event-scheduling update scheme is used for realistically capturing the wide range of human activities at multiple timescales.

In an SICU ward, most activities range from seconds to weeks. Below is an illustrative but not exhaustive list¹.

- The length of stay (LOS) of patients is typically several days to more than a week. Some patients need to stay as long as 60 days due to complications.
- The treatment of diseases, due to infections from Methicillin Resistant *Staphylococcus Aureus* (MRSA) colonization, typically lasts around 2 weeks.
- The healthcare workers (HCWs) work based on their shifts, which are on a daily basis. For example, there are 3 shifts for most nurses (morning, afternoon, night), and several different kinds of shifts for physicians. Some physicians may appear only during daily ward rounds for diagnosis (e.g., for consultants and registrars) while others are on 24-hour on-calls (e.g., for medical officers). There are also staff who work for long hours in the ward during the day time (e.g., physiotherapists).
- The HCW activities range from minutes to a few hours: routine works such as temperature and blood pressure measurements, bed turning, hand-over between shifts, roll calls, typically last from

¹ The data are collected during the field study at National University Hospital, Singapore (NUH), by directly observing people's daily activities and interviewing the various types of the healthcare workers. NUH has also provided us the historical statistical data for parameters such as patients' admission rate and length of stay.

5 minute to 45 minutes.

- The movement of human hosts is at the scale of seconds. Their movement patterns cannot be conveniently omitted because when these patterns overlap in the temporal-physical space, chances of disease transmission arise.

For succinctness, we will elaborate only two examples in detail: (1) the events scheduling of patients' admission and their length of stay, and (2) the daily work of a staff nurse.

Figure 4-10 Simulation of patient activities upon admission (arrow indicates simulation time)

Figure 4-10 shows the typical activities of a patient as observed in our field study at NUH. A patient does not actively perform any activities and all events that involve the patient are triggered by HCWs (such as treatment and contact) or due to interventions such as screening tests. Each patient is assigned a bed upon admission, and will remain bedridden until the date of discharge.

Figure 4-11 Event scheduling for patient admission and stay

The hospital data¹ suggest that the SICU admission rate is 50 per 30 days, and the LOS ranges from 1 to 60 days, with the average of 7 days. Figure 4-11 shows an extract of the events involved in these processes. The Patient Admission Process is a system-level behavior rule that periodically

¹ Historical data were provided by the SICU, Ward 21, National University Hospital, Singapore (NUH).

introduces new patient into the ward, mimicking the arrival of new patients. The inter-arrival time, τ_1 had been modeled to follow an exponential distribution with the mean of 0.6 days. The labels 1 to 6 are the time points where a patient is introduced. The patient introduction event is recurring without a guard condition: after a new patient is introduced, the simulator draws a new value for τ_1 , and schedules the next patient introduction event to occur in τ_1 time; this process continues until the end of the simulation.

Once a patient is introduced in the simulation, she is assigned to a bed and changes status to HOSPITALIZED. Gender is not a risk factor in our simulation of MRSA; hence we arbitrarily use the feminine pronoun when referring individual patients. Her LOS (τ_2) is determined by drawing a random number from the Beta distribution (min=1, max=60, mean=7). For the case of Patient A, she stays until the end of LOS (label 8); then her status is changed to DISCHARGED, and her bed is released. This status change would implicitly cancel other events whose condition requires the patient to be still HOSPITALIZED. For example, a staff nurse could have scheduled her daily routines at the start of the shift, which includes several visits to this patient on that day; once the patient is discharged, these scheduled visits would be implicitly cancelled. In reality, a patient's health status is usually re-examined and the discharge decision can be revised. We model this by giving the patient a 50% likelihood of extending their hospital stay by two more days (τ_3) when the scheduled discharge event is invoked. The case of Patient B illustrates such a scenario. At the originally planned discharge time (label 10), a new event is scheduled in two days' time (label 11) to mimic an LOS extension. This patient eventually extends the LOS twice and then gets discharged (label 12).

The scheduling of a staff nurse's daily routine is more complicated. A nurse who is not on duty (at "off work" state) can be recruited for work by the respective shift manager when a shift starts. When on duty, their behavior comprises pre-assigned role-specific activities which we refer to as the staff's *mandatory activity list*. Table 4-2 is an example of mandatory activities for a morning shift staff nurse; the activities, their timings and durations were obtained through direct interviews with the staff. The morning shift for a day begins at 0700h and ends at 1430h. At 0700h, the staff nurse is required to attend the roll call in the meeting room, which takes 10 to 45 minutes, and likely to be 30 minutes. Then she needs to update the patient records at 0830h, 0930h, 1030h, 1130h and 1230h at the

assigned PC. This is specified as “starting time: 0830h; frequency: 5 (times); interval: 60 minutes” in the mandatory activity list. Each update takes 5 to 8 minutes.

Table 4-2 Sample of mandatory activity list for a morning shift staff nurse
(obtained through interview)

Activity Type	Start Time (24h)	Frequency	Interval (minutes)	Duration per Activity (minutes)			Location of Activity
				Min	Likely	Max	
Roll Call	0700	1	0	10	30	45	Meeting Room
Take Report	0730	1	0	15	20	45	Assigned PC
Prepare medication	0800	1	0	5	10	15	Nearest Injection Trolley
Assess patient	0815	2	240	15	15	15	Assigned Bed
Update computer	0830	5	60	5	6.5	8	Assigned PC
Follow rounds	0900	1	0	20	32.5	45	Ward
Break	1100	1	0	10	27.5	45	Outside ward
Follow up orders	1230	1	0	30	45	60	Assigned bed
Hand over	1330	1	0	15	20	45	Assigned PC

Figure 4-12 Event scheduling for HCW activities

Between mandatory activities, the nurses undertake one or more arbitrary activities such as visiting other PCs (of other rooms), going to the staff room, pantry or to the meeting room. Once all the mandatory activities are completed, the nurse leaves the ward, returning to the “off work” state. Figure 4-12 illustrates the simulated work cycle of a nurse. In summary, the nurse’s work cycle switches between “working” and “off work”. When “working”, they execute the mandatory activities (in the task list), which are interleaved by arbitrary activities (not in the task list).

Figure 4-13 Simulation of nurse activities

Figure 4-13 shows how a nurse’s activities are modeled with the event-scheduling update scheme. The Morning Nurse Shift Manager, which is a system-level entity (rather than an actual human host) is invoked at 0655h each morning (label A). When invoked, it recruits the required number of nurses (e.g., 10 staff nurses, 2 assistant nurses) from the “off work” staff pool to attend the morning shift. Then the Morning Nurse Shift Manager schedules the `start_work` event at 0700h for each of the recruited nurses with the scheduler. Once the recruitment is finished, the Morning Nurse Shift Manager schedules the next occurrence of its recruitment event, which is in $t_1 = 24$ hours (label B) time. That is, the Morning Nurse Shift Manager is only active for a short period at 0655h each day, and remains inactive (no operation execution) throughout the rest of the day.

A recruited nurse (shown by the `staff nurse A` in Figure 4-13) enters the ward during the `start_work` event (label C). During this event, she schedules an `end_of_shift` event with the scheduler, which is $t_2 = 7.5$ hours after 0700h, at 1430h. She also takes the mandatory activity list and schedules each activity on the list with the scheduler (only 3 incidences are shown in Figure 4-13, labeled by t_x). For example, she would schedule the `take_report` activity 30 minutes later (at 0730h) according to Table 4-2. Once the time is reached, the scheduler invokes the nurse to execute the activity specified for that time. If the nurse is not busy with other activities, she will start on the activity immediately, otherwise she will save the activity to an “on-hold” list, finish the current activity, and then pick up an activity from the “on-hold” list for execution.

For each activity execution, there are two general steps: move to the destination location (e.g., the patient’s bed, or a computer), and stay there for certain amount of time. More steps can be added for special types of activities such as shift hand-over, ward round, and covering each other’s duties. Figure 4-13 shows an example of moving from Waypoint 0 (WP₀) to Waypoint 3 (WP₃), by passing Waypoint 1 (WP₁) and Waypoint 2 (WP₂). Waypoints are hypothetical nodes in the two-dimensional map that are connected to show the possible paths a human host can take from one location (a node on the map) to another. A nurse plans her path of movement using the Waypoint network and the start and end locations. Once the path is computed, she moves along the path at 1 meter per second, with an update frequency of 1 second or less. This is shown in the enlarged visualization of the movement scheduling in Figure 4-13. Once she is close to a particular Waypoint (e.g., WP₁), the last move is less than 1 metre (taking less than 1 second), which makes her “snap” to the next Waypoint (WP₁). At the end of every 1-second-or-less movement, she schedules the next movement event (in 1 second or less), by taking into the consideration the current distance to the next Waypoint, until the destination Waypoint is reached.

Upon reaching the destination, the nurse stays there to execute the mandatory activity for a duration of t_y (actual values are specified by the “Duration per Activity” column in Table 4-2). She does so by scheduling an `end_of_activity` event with the scheduler. When the time is up, the scheduler calls back and the mandatory activity is completed. If more details of the activity need to be modeled (e.g., to count the number of physical encounters between the nurse and a patient during

an activity), more detailed steps can be scheduled as events to make up the entire activity in a similar way.

Once a mandatory activity is completed, the nurse immediately picks a mandatory activity from the “on-hold” list if there exists one, such as the case labeled by b and 2 in Figure 4-13: by the time the first mandatory activity is completed (label 2), the time of starting the 2nd mandatory activity is already passed (label b), hence the nurse should start the 2nd mandatory activity immediately. If there are still time left before the next mandatory activity starts (such as label 4 in Figure 4-13), the nurse fills the time with arbitrary activities. Each arbitrary activity also consists of two general steps: move to the destination, and stay there for some time to execute the activity. When all the mandatory activities are completed (label 6) and the shift time is up (label D), the nurse is considered to have completed the work for the shift. Then she leaves the ward, and changes her status to “off work”. That is, the nurse can leave either on label 6 or label D, whichever is later.

The above description is a much simplified version of a nurse’s daily work routine. Nevertheless, it has sufficiently demonstrated the capability of event-scheduling update scheme in effectively and realistically covering activities at various timescale ranging from seconds to days in a ward model. More complex issues such as activity prioritizing (e.g., modeling of emergencies) and activity synchronization (e.g., the ward round activity that requires cooperation of several physicians and nurses, who should visit the patients together) are not discussed here. Interested readers may refer to our paper [180] for more details.

4.5. Summary

Multi-timescale activities are the norm in ABM applications. In this chapter we aimed to develop an update scheme that is both *temporally realistic* and *computationally efficient* for ABM applications. After reviewing the time-step-driven update schemes and event-driven update schemes, we proposed an event-scheduling update scheme with adaptations made for ABM applications. This scheme is made up of a series of events with a pre-specified event structure, and a discrete event scheduler that orders and invokes events in chronological order. It supports implicit event cancellation through events’ guard conditions, and explicit event cancellation by fast searching for an event in the

scheduler. The concurrent processes are modeled such that each process schedules its own update events with the scheduler, so that the update events belonging to different concurrent processes are called in a interleaved manner, according to their timestamps. We demonstrated the application of this update scheme to the B cell life cycle modeling, patient admission and stays, and daily work routine of a staff nurse. The event-scheduling update scheme realistically models the multi-timescale events, and outperforms the equivalent models with the time-step-based update scheme considerably.

Chapter 5

Modeling Entities and Their Environment

The event-scheduling update scheme described in the previous Chapter enables us to model state changes over time; but it does not describe how the entities come into interaction and how their states, which will change over time, are represented. In this Chapter we focus on these issues: how entities can be presented, and how they relate and interact with each other through the environment.

In ABMs, entities are usually modeled at the *individual level*, which many have perceived as a defining character of ABMs [81, 82, 84, 147]. Advantages with this approach include the flexibility of modeling highly heterogeneous individuals (in both their states and behavior rules) for a closer representation of the reality. On the other hand, its disadvantages are the associated computational cost of explicitly expressing every model entity as an individual agent; this leads to the issue of model scalability, that is, the model becomes computationally expensive and even intractable when the number of agents increases. Hence we raise the doubt whether it is necessary to model every entity uniformly at individual level in an ABM simply for the sake of treating them as individuals, especially for those relative simple, homogenous entity types such as the interleukins in an immune system. If ABMs can be made flexible in its entity representations by allowing both population level and individual level representations without sacrificing model realism, it would accommodate larger quantities of entities more easily hence improve the model scalability.

The choice of entity representation is closely related to how the environment is modeled in the ABM. The environment model provides the basis of the entities' interactions, such as their positioning and the spatial closeness to each other, and hence influences their chance of contacts and interactions. An accurate environment model should enable realistic positioning of entities and chances of interactions. Besides, the environment model needs to be topologically accurate and account for the different properties of its local compartments as well. It would be sufficient to assume a three dimensional environment filled with well-mixed agents if the model is to mimic an *in vitro* condition, e.g., biological cells being cultivated in dishes. However, such a simplified assumption does not

suffice for *in vivo* conditions, where cells constantly circulate between biological compartments (for example, blood, tissues, lymph nodes, spleen). Each biological compartment differs greatly in terms of their molecular micro-environment, which in turn causes cells to behave differently. For example, naive and effector lymphocytes preferentially enter lymph nodes and infected tissues respectively, because they express different cell surface receptors that interact with endothelial cells in different parts of the blood vessels [45]. The compartmentalized topological environment can also be found in a hospital or ward model: a ward is usually divided into rooms for different functions, such as rooms for patients, doctors, equipments, clean or dirty utilities, meetings, visitors, and so on. In addition, there are special-purpose compartments, such as ventilation systems, being superimposed on the open spaces.

In this Chapter we develop the modeling approaches for these two closely connected problems: hybrid entity representations, and the environment model.

5.1. Motivations of hybrid entity representation

Entities in epidemiological applications may differ from each other drastically in terms of physical sizes and relative quantities, and at the same time they closely interact with each other in a deliberate and systematic manner. An example is the various biological cells and molecules in the human immune system: antibodies are about 15 nm [102], HIV is about 0.1 μ m [103], while a T cell is around 5-10 μ m [104]. In terms of relative quantities, CD4 epitopic density on circulating lymphocytes are measured to have a mean level of 46,000 epitopes per cell [105]. The intercellular interactions between cells take place through direct cell contact and chemokine passing. Chemokines are molecules secreted by one type of cell and received by another type of cell. These molecules affect by binding to the receptor molecules expressed on the target cells, starting a chain of molecular reactions within the cell. This may lead to state changes of the cell, such as activation or migration. For example, TH2 cells (a subset of T helper cells) release cytokines such as interleukins IL-4 and -5, which are required for B cell differentiation.

Differential equation (DE) based models for biological cell populations assume that collectively, cell interactions obey the mass action law and their movements obey the diffusion law [26, 36]. This

assumption is reflected in their use of reaction–diffusion equations [181]. The application of mass action law means that the likelihood that two cells will interact is proportional to the concentration of each of the two types of cells. The application of the diffusion law implies that biological cells do random movement, like molecules. However in reality, the movement and encounters of biological are known to be purposeful rather than random [45, 67], hence the reaction-diffusion equations may not be applicable. As an example, cell distributions within a lymph node are non-uniform: B cells and T cells concentrate in different areas known as B zones and T zones [45]. Research has revealed that T cells are recruited to the T zone through the interaction between T cell’s CCR7 receptor and its ligands, SLC and ELC [60, 182], though it has not been firmly established whether the movements are entirely attributed to chemotaxis [183]. Similarly, dendritic cells, which present antigens to T cells so as to trigger an immune response, also bear CCR7 on their surface and are attracted to the same T zone [67, 184, 185]. As a result, the chance of interaction between dendritic cells and T cells (and hence the chance that T cells recognize antigens) are significantly increased as compared to the simplistic assumption that they interact only through Brownian cell movements. These complex interaction and movement behaviors of biological cells impose great challenges to the equation-based models, since they aggregate the entities at the population level and rely on the reaction-diffusion equations to describe their interactions and movement. Hence, we believe that a population-level of entity representation is insufficient to model biological cells; an individual level of entity representation would be more appropriate.

Molecules have also been modeled as individual agents in some ABMs [44]. Compared to cells, molecules are much smaller in physical sizes, and much larger in quantities. An appropriate relative population size of cells and molecules necessarily implies a huge population of molecules with only tens of hundreds of cells, which easily causes the model to be computationally intractable. However, molecules are less complex in their behavior. Hence we can consider an alternative approach that uses the population level entity representation. Individually, molecular interactions can be considered as chemical reactions, and collectively, the resulting dynamics can be quantitatively captured by the *law of mass action* [186, 187]. The aggregation of their individual random movement can also be captured by *Fick’s law of diffusion* [188] at the population level. By using mass action law and the diffusion law, we eliminate the need to model molecules individually.

The issue of entity relative scale also exists in the inter-host epidemiological applications. In a hospital ward model we are mostly concerned with human hosts and the pathogens (bacteria or virus). We may also be interested in a particular set of physical objects which can act as fomites, such as door handles and bed railings. The interactions between these model entities are known to be the driving force of the disease spread and development [54, 120, 124]. In this application, we can also observe the similar contrast in entity relative scales between human hosts (larger in physical size, smaller in population size, and highly complicated in states and behaviors) and pathogens (tiny in physical size, numerous in quantities, and less complicated than human hosts). Therefore, a hybrid treatment of entity representation, namely, the individual level representation of human hosts, and population level representation of pathogens, could lead to a more realistic and sufficient model.

For ease of description, we refer to individually modeled entities as *agents*, and the entities grouped into populations as *population quantities*. Though the idea of a hybrid entity representation is simple – combining the *agents* and *population quantities* in one ABM, it might not be obvious here that hybridization would actually takes place at two levels: globally in the environment, and locally with an agent. We present the details in the next Section.

5.2. The hybrid modeling approach

In our proposed approach, the hybridization of entity representation is at two levels:

- *Globally in the environment*: where *agents* and *population quantities* coexist in one environment, and are able to interact with each other (e.g., through spatial proximity). This is presented in Section 5.2.1.
- *Locally with an agent*: where *agents* can possess a subset of *population quantities*. For example, cells may have certain amount of molecules on its surface, or internally within its cell body. This is presented in Section 5.2.2.

5.2.1. Layered environment modeling

The *environment* is where model entities reside and interact. The environment model handles the storage and provision of spatial or topological proximity information of these entities. It also handles environment level events that are exogenous to individual agents, such as the background

flows (e.g., blood flows) that drive the movement of the entities. Our view of the environment is not merely as a passive holder to place the agents, but as an active system-level entity with its own activities and special properties (e.g., blood flows) which can greatly influence the behaviors of the agents.

Our environment model consists of four types of functional layers superimposed in the same virtual space:

- A *Connection Layer*, which specifies the joint points between the current compartment and the neighboring compartments, such as the outlet vents of a ventilation system.
- An *Agent Holder Layer*, where all agents reside.
- Several *Population Quantities Layers*, where the distributions of the population quantities are recorded.
- Several *Flow Field Layers*, where background flows, such as blood flows and lymph flows are modeled.

Figure 5-1 Four-layer structure of the simulated environment

Each point in the environmental space is a composition of all the layer functionalities at that particular coordinate. To illustrate, Figure 5-1 shows the environmental space as stacking 4 layers of two-dimensional planes, but the concepts are not necessarily restricted to a two-dimensional space but applicable to any number of dimensions. The layers are classified into *continuous* and *discrete space* layers. In the discrete space layers, values, such as scalar values in the Population Quantities

Layer, and the vector values in the Flow Field Layer, are defined only at discrete grid points where the space is quantized. In continuous space layers, real values can be specified for the coordinates for the joint points (in the Connection Layer) and the agents (in the Agent Holder Layer), thus they can reside on any point in the space regardless of the space quantization. When composing discrete and continuous layers, values at intermediate coordinates are determined using interpolation methods such as the bilinear interpolation. We will exemplify on the interpolation method when discussing the agent's sampling of the environment in the later sections.

The *Agent Holder Layer* contains all the *agents*. The *agents* denote entities that are modeled at the *individual* level, which could be biological cells or human hosts as discussed in Section 5.1. The location of each agent is specified by its real-valued coordinates in this layer. In the simplest case where the agents are modeled as spheres, two agents collide if the distance between their coordinates is shorter than the sum of their radius. This physical contact may trigger more inter-agent interactions, as defined by agent's behavioral rules. For example, when an effector cytotoxic T cell makes contact with an infected cell, the cytotoxic T cell will kill the latter if its receptors can match the antigens presented on the surface of the latter [45].

The *Population Quantities Layer* and *Flow Field Layer* share a common mesh grid structure, which is a discretized version of the space. We use ΔL to denote the distance between two neighboring mesh grid points. The value of ΔL determines the *granularity* of the discrete space. A grid point indexed by (i, j) corresponds to a real-valued coordinate $(i\Delta L, j\Delta L)$ in the continuous space of the Agent Holder Layer. Therefore, each agent is necessarily located within some grid cell. This correspondence in spatial locations is the basis of the communications between layers.

The *Population Quantities Layer* consists of multiple sub-layers, each representing the spatial distribution of one type of population quantities. The local quantity (e.g., number of molecules) at each grid point (i, j) is denoted by $Q_{i,j}$. The change in the spatial distribution of local quantity is subject to the simultaneously occurring events: production, diffusion, external flow, consumption, and decay. Expressed formally, the net change of $Q_{i,j}$ over time Δt is the combined result of the factors above:

$$\frac{\Delta Q_{i,j}}{\Delta t} = \left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{production} + \left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{consumption} + \left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{diffusion} + \left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{decay} + \left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{flow} \quad \text{Eq. 5-1}$$

We describe each event below.

- *Production* ($\left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{production}$). Production refers to how new quantities are generated and injected

into the environment. For example, cells may synthesize and release cytokines into its neighborhood, or a patient who sneezes may transfer the aerosolized pathogens attached to him/her to the neighboring environment. This quantity change emerges from the collective interactions of the individual agents and population quantities. Hence there is not an explicit mathematical expression for computing the production amount.

- *Consumption* ($\left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{consumption}$). Consumption refers to the loss of quantities due to interactions

with agents or reactions with other types of population quantities. For example, molecules can be bound to cell receptors and become part of the cell (agent). Similar to *production*, we do not have an explicit mathematical formula for this quantity change since *consumption* emerges from interactions especially with individual agents.

- *Diffusion* ($\left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{diffusion}$). Diffusions refer to the quantity changes resulting from Brownian

(random) movement of entities. This can be expressed using the diffusion law. After applying the finite difference method to the diffusion law over a two-dimensional space, we obtain:

$$\left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{diffusion} = \frac{D}{\Delta L^2} (Q_{i-1,j} + Q_{i+1,j} + Q_{i,j-1} + Q_{i,j+1} - 4Q_{i,j}) \quad \text{Eq. 5-2}$$

where ΔL is the distance between two neighboring grid points and D is the diffusion constant. For this numerical solution to be stable, it requires $\Delta t \leq \Delta L^2 / 4D$ [189]. Intuitively, Eq. 5-2 says

that $\left. \frac{\Delta Q_{i,j}}{\Delta t} \right|_{diffusion}$, the change of the population quantity at (i, j) over a time step of Δt due to

diffusion is the net summation of the outflows and inflows at (i, j) from the neighboring grid points.

- *External flow* $(\frac{\Delta Q_{i,j}}{\Delta t} \Big|_{flow})$. Quantities may shift due to external flow fields. For example, in the blood vessels and lymph duct, the molecules are pushed by the local fluid flow; the pathogens in the hospital ward may flow towards the ventilation system due to the absorbability of the latter. We will describe the mathematical expression of this term in detail with the *Flow Field Layer* in the subsequent paragraphs.

- *Decay* $(\frac{\Delta Q_{i,j}}{\Delta t} \Big|_{decay})$. Quantities may decay (or depredate) by themselves, and disappear from the environment. For example, molecules decay through various pathways (e.g., instability, enzymic reactions). The amount of decayed population quantities within time-step Δt is proportional to the existing quantities, and is approximated by:

$$\frac{\Delta Q_{i,j}}{\Delta t} \Big|_{decay} = -\lambda Q_{i,j} \quad \text{Eq. 5-3}$$

where λ is the decay constant.

The *Flow Field Layers* characterize the background flow in the environment. Each represents a kind of systematic transport superimposed onto the random movements (by diffusion) of population quantities and agents. At each grid point (i, j) , the flow velocity is given by $\mathbf{F}_{i,j}$. A simple example is laminar flow, which is the normal condition for blood flow in most of the blood vessels: the flow velocity is the highest in the center of the vessel and lowest along the vessel wall [190]. Turbulent flows occur when laminar flow is disturbed, such as at the branch point of the vessels [190]. These flow profiles are environment-dependent and modeled by the Flow Field Layers.

Figure 5-2 Modeling the effect of flow on population quantities distribution

If a type of population quantity is distributed by background flows, the corresponding Flow Field Layer shifts its distribution towards the direction of flow. Consider a grid point indexed at (i, j) in Figure 5-2 that has the real coordinate is $(i\Delta L, j\Delta L)$. Its governing space, $V_{i,j}$ is the rectangle $(\Delta L \times \Delta L)$ centered at $(i\Delta L, j\Delta L)$. Let $Q_{i,j}$ denote the population quantity within $V_{i,j}$, and let $F_{i,j} = (F_{i,j,x}, F_{i,j,y})$ denote the flow velocity (vector) at $(i\Delta L, j\Delta L)$. Within Δt , the quantity $Q_{i,j}$ is shifted along the direction of $F_{i,j}$ by $(F_{i,j,x}\Delta t, F_{i,j,y}\Delta t)$. Let $V'_{i,j}$ be an imaginary rectangle space $(\Delta L \times \Delta L)$ centered at $(i\Delta L + F_{i,j,x}\Delta t, j\Delta L + F_{i,j,y}\Delta t)$, that is, the “shifted” space that contains $Q_{i,j}$ after Δt . Let $S_{u,v \rightarrow i,j}$ denote the overlapping space between $V_{i,j}$ and $V'_{u,v}$ (i.e., the governing space of grid point indexed at (u, v) shifted by its local flow $F_{u,v}$). Hence the proportion of $Q_{i,j}$ that remains in $V_{i,j}$ is $Q_{i,j} S_{i,j \rightarrow i,j} / \Delta L^2$, and the rest are distributed to the governing space of the neighboring grid points. The population quantity within $V_{i,j}$ after Δt is the sum of the “shifted” quantities (due to flow) from all the neighboring grid points that fall into $V_{i,j}$. Therefore, the change of population quantity at (i, j) due to flow is:

$$\Delta Q_{i,j} \Big|_{flow} = \sum_{u=i-1}^{u=i+1} \sum_{v=j-1}^{v=j+1} \frac{Q_{u,v} S_{u,v \rightarrow i,j}}{\Delta L^2} - Q_{i,j} \quad \text{Eq. 5-4}$$

The position of agents in the Agent Holder Layer is affected by both the Population Quantities Layers and the Flow Field Layers. For example, a biological cell can be passively driven by the background

flows (e.g., the blood flow). It moves not only passively, but also actively in response to the local molecular concentration and gradient, exhibiting *chemokinesis* and *chemotaxis* [30]. Formally, we model the displacement $\Delta \mathbf{p}$ of an agent in time Δt as a function of the local flow velocity $\mathbf{F}_p$, the local population concentrations $C_{m,p}$ and gradients $\mathbf{G}_{m,p}$ for a set of population quantities $\mathbf{m}$ (Eq. 5-5).

$$\frac{\Delta \mathbf{p}}{\Delta t} = \mathbf{d}(\mathbf{F}_p, C_{m,p}, \mathbf{G}_{m,p}) \quad \text{Eq. 5-5}$$

The definition of $\mathbf{d}(\bullet)$, that is, how an agent combines the effect of the external field and local concentrations, is determined by the agent's behavioral rules. Some types of agents may be highly motile while others are structural in function, hence immovable. The agent may also be observed to respond to only specific types of $\mathbf{m}$, depending on its type and state. In practice $\mathbf{d}(\bullet)$ may be expressed as mathematical equations, or if-then decision logics, or a hybrid of both; this is generally subject to the modeler's knowledge and judgment.

Figure 5-3 Sampling of the local molecular concentration and gradient by an agent

The environment needs to provide agents with discrete information on local concentrations and gradients of population quantities as well as the local flow vectors. Determining the local effect of this information for an Agent's position (within the Agent Holder Layer) requires interpolations from quantities defined at neighboring grid points. We adopt the bilinear interpolation (for a two-dimensional space) for its simplicity and efficiency as it requires considerably fewer points than the cubic or spline method (4 vs. 16 in a two-dimensional space). Consider the agent in Figure 5-3 (depicted by a circle), whose nearest grid points are indexed by (i, j) , $(i, j+1)$, $(i+1, j)$, and $(i+1, j+1)$ respectively. The local concentration of population quantities m at the agent's position obtained through bilinear interpolation is:

$$C_{m,p} = \Delta L^{-4} (Q_{i,j} (\Delta L - \Delta x) (\Delta L - \Delta y) + Q_{i+1,j} \Delta x (\Delta L - \Delta y) + Q_{i,j+1} (\Delta L - \Delta x) \Delta y + Q_{i+1,j+1} \Delta x \Delta y) \quad \text{Eq. 5-6}$$

The local flow velocity is:

$$\mathbf{F}_p = \frac{\mathbf{F}_{i,j} (\Delta L - \Delta x) (\Delta L - \Delta y)}{\Delta L^2} + \frac{\mathbf{F}_{i+1,j} \Delta x (\Delta L - \Delta y)}{\Delta L^2} + \frac{\mathbf{F}_{i,j+1} (\Delta L - \Delta x) \Delta y}{\Delta L^2} + \frac{\mathbf{F}_{i+1,j+1} \Delta x \Delta y}{\Delta L^2} \quad \text{Eq. 5-7}$$

The local gradient $\mathbf{G}_{m,p} = \nabla C_{m,p}$ is a vector with components:

$$\begin{aligned} G_x &= \Delta L^{-4} ((Q_{i+1,j} - Q_{i,j}) (\Delta L - \Delta y) + (Q_{i+1,j+1} - Q_{i,j+1}) \Delta y) \\ G_y &= \Delta L^{-4} ((Q_{i,j+1} - Q_{i,j}) (\Delta L - \Delta x) + (Q_{i+1,j+1} - Q_{i+1,j}) \Delta x) \end{aligned} \quad \text{Eq. 5-8}$$

We have described the layered environment model, and the hybrid entity representation in such an environment. Though agents and population quantities reside in layers with different spatial granularities (discrete or continuous), their interactions across layers are realized by spatial proximity through simple interpolation. In next Section we present the hybrid representation locally to agents.

5.2.2. Hybrid representation locally to agents

Entities expressed as population quantities are generally much smaller in physical size than those are modeled as agents. While these entities can be freely moving in the environment, they can also attach to other individual agents and become their local properties. In this Section, we demonstrate this local hybrid representation through an example of a receptor kinetics model for a biological cell. This model will be subsequently used for the chemotaxis experiment in Section 5.3.

In this receptor kinetics model, we are interested in three entity types: the cell (agent), its receptor molecules (population quantities), and ligand molecules (population quantities). We consider three interdependent molecular events for receptor kinetics [27]:

- *Receptor-ligand binding*, where free ligand molecules bind to receptors on the cell surface.
- *Receptor internalization and ligand consumption*, where the receptor-ligand complex is internalized into the cytosol, and the ligand is used up or digested by other biological processes.
- *Receptor recycling*, where the previously internalized receptors are returned to the cell surface, to be made available for binding again.

Though our hybrid approach treats molecular interactions as quantities at the population level, it is instructive to understand how the collective effects of these interactions give rise to the macroscopic

descriptions. Readers who are familiar with mathematical modeling approach would find the following equations familiar. However, we need to emphasize that instead of a global description of the molecular kinetics, these equations are part of the agent's behavioral rules, that is, at the scale of individual agents.

Consider a microscopic event where an individual ligand L meets an individual receptor R and binds to it, producing an occupied receptor LR . This event can be described as a chemical equation of the form:

The left hand side of Eq. 5-9 describes that for a single binding event, one L and one R are consumed, and the right hand side is the result that one LR is produced. Microscopically, the parameter k_b is a quantitative indication of how likely the binding would happen given that the two reactant molecules meet. Macroscopically, each binding event reduces the R and L quantities by 1 each, and increases the LR quantity by 1. The rate at which the binding process takes place is proportional to k_b and the quantities (or concentrations) of R and L . Based on the mass action law, it is straightforward to derive the corresponding macroscopic description of Eq. 5-9 as differential equations:

$$\left. \frac{d[L]}{dt} \right|_b = \left. \frac{d[R]}{dt} \right|_b = - \left. \frac{d[LR]}{dt} \right|_b = -K_b [L][R] \quad \text{Eq. 5-10}$$

Traditionally, the notation $[\bullet]$ denotes the *concentration* of molecules in the whole space (with the unit M or mole/L), but here we redefine it as the *quantity* of molecules *in the proximity of*, or *within* the agent. $[L]$ is the *local* number of ligand that the agent samples from its immediate proximity, which is the concentration of the ligand at the agent's location multiplied by the agent's volume. $[R]$ is the number of receptors expressed on the agent (cell) surface, as one of its state variables. The rate constant K_b is proportional to its microscopic counterpart k_b and denotes the number of bindings per unit time per pair of reactants. One would recognize K_b to be similar to the *forward rate constant* or *association rate constant* in enzyme kinetics [191] except for the difference in the unit. Enzyme kinetics typically deals with concentrations (assumed uniform distribution in space) while here we deal with quantities that are local to each agent. For simplicity, we assume that the disassociation of R and L is rare and can be ignored. The equations in Eq. 5-10 describe the change in population quantities local to a single agent, that is, it is a behavioral rule of the agent

expressed in the equation form.

Next we consider the cell's behaviors of receptor internalization/ligand consumption and receptor recycling. One example of these processes is receptor-mediated endocytosis [192], where a biological cell takes in molecules that are too big to enter the cell through diffusion or transport proteins. The bound ligand is disassociated from the receptor and used by other biological processes. The freed receptor is then returned to the cell surface. The microscopic chemical equations are:

where k_i is the rate at which a single LR molecule converts to a R^* , a free receptor molecule still in the cytosol, and k_r is the rate at which the receptor is returned to the surface. The corresponding macroscopic (but local to the particular agent's environment) differential equations are:

$$\left. \frac{d[LR]}{dt} \right|_i = - \left. \frac{d[R^*]}{dt} \right|_i = -K_i[LR] \quad \text{Eq. 5-13}$$

$$\left. \frac{d[R^*]}{dt} \right|_r = - \left. \frac{d[R]}{dt} \right|_r = -K_r[R^*] \quad \text{Eq. 5-14}$$

If Alive Then

Sample the environment: get current local ligand quantity $[L]$:

Get the values of individually stored quantities $[R]$, $[LR]$, and $[R^*]$

Perform following updating rules *in parallel*:

a. reduce $[L]$, $[R]$ by $K_b[L][R]\Delta t$, and increase $[LR]$ by $K_b[L][R]\Delta t$

(receptor-ligand binding)

b. reduce $[LR]$ by $K_i[LR]\Delta t$, and increase $[R^*]$ by $K_i[LR]\Delta t$

(receptor internalization and ligand consumption)

c. reduce $[R^*]$ by $K_r[R^*]\Delta t$, and increase $[R]$ by $K_r[R^*]\Delta t$

(receptor recycling)

End If

Listing 5-1 Agent rule: receptor-ligand interaction

The effects of macroscopic equations Eq. 5-10, Eq. 5-13, Eq. 5-14 are expressed by an agent's behavioral rules (shown in Listing 5-1). Here, Δt refers to how long it is since the last invocation of

the rule. Δt is not necessarily a constant; one example of variable Δt is given in Eq. 5-17, where Δt varies with the speed of the cell. One way is to randomize the order of their execution. The microscopic molecular events in Listing 5-1 are taking place concurrently, hence we must avoid a predefined order of performing these updates; that is, they should be executed *in parallel*. Alternatively, we can achieve true parallelism by combining equations Eq. 5-10, Eq. 5-13, Eq. 5-14 so that the factors affecting the molecular quantity $[X]$ are arranged to appear on the right-hand-side of the same equation. The rearrangement is mathematically expressed as follows:

$$\frac{d[X]}{dt} = \sum_s \frac{d[X]}{dt} \Big|_s, \text{ where } s \in \{b, i, r\}, X \in \{R, L, LR, R^*\} \quad \text{Eq. 5-15}$$

This gives rise to the system of coupled equations shown in the right-hand column of Table 5-1. The revised version of the agent rule form is given in Listing 5-2. These equations are used in the action part of the agent rule to update the local molecule quantities $[L]$, $[R]$, $[LR]$, and $[R^*]$. As the ligand quantity $[L]$ varies with the agent's location, it adds nondeterministic factors to the results, even the rules are expressed in terms of deterministic equations.

Table 5-1 Microscopic rules and macroscopic equations for receptor-ligand interactions that are local to an agent's environment

Individual Reactions	Quantity Changes (combined updating rule)
<p>Receptor-ligand binding</p> $L + R \xrightarrow{k_b} LR$ <p>Receptor internalization and ligand consumption</p> $LR \xrightarrow{k_i} R^*$ <p>Receptor recycling</p> $R^* \xrightarrow{k_r} R$	$\frac{d[L]}{dt} = -K_b[L][R]$ $\frac{d[R]}{dt} = -K_b[L][R] + K_r[R^*]$ $\frac{d[LR]}{dt} = K_b[L][R] - K_i[LR]$ $\frac{d[R^*]}{dt} = K_i[LR] - K_r[R^*]$

The rule we specify here is quite flat as there is only one proposition and all the equations appear in the action part. However, it is possible to specify more complex situations that an agent can be in, and the different responses it could make. Readers may refer to Listing 6-3 for sample agent rules where nested Boolean logic is used for deciding the agent's actions.

In the next Section, we apply the proposed hybrid entity representation modeling approach to a case

study of chemotaxis.

```
If Alive Then
```

```
    Sample the environment: get current local ligand quantity  $[L]$ :
```

```
    Get the values of individually stored quantities  $[R]$ ,  $[LR]$ , and  $[R^*]$ 
```

```
    Compute  $d[L]$ ,  $d[R]$ ,  $d[LR]$ ,  $d[R^*]$  using equations in Table 5-1
```

```
    Update local quantities:  $[X] = [X] + d[X]$  ( $X \in \{R, L, LR, R^*\}$ )
```

```
End If
```

Listing 5-2 Agent rule: receptor-ligand interaction (rewritten for parallel updating)

5.3. Case study 1: chemotaxis

Chemotaxis refers to the directed cell movement towards or away from a chemical source (chemoattractant or chemorepellent) by sensing and following the *chemical gradient* [30] in its immediate neighborhood. Through chemotaxis, a sperm cell finds the ovum, a white blood cell finds the place of injury or inflammation, and a nerve cell grows its axon to establish synaptic connections [30]. Chemotaxis relates the movement patterns of cells to the spatial distribution of molecules; hence they serve as a good benchmark for illustrating the proposed hybrid entity representation modeling approach.

The phenomena of chemotaxis has been investigated as well as modeled extensively. The most popular mathematical model is the Keller-Segel's Model [26, 35, 36], which is a PDE model describing both cells and molecules as population level quantities. The model is *phenomenological* in that it abstracts away molecular details of the transduction of extracellular signals to cell locomotion as the detailed molecular interactions were not well-understood at that time. In the past decade however, much of bioscience research has been devoted to discovering the mechanisms that transduce the extracellular chemical signals into cell locomotion [34, 68, 69, 193-195]. These include molecular events such as G-protein activation, lipid remodeling, protein kinase activation, calcium elevation [30, 193], and cellular events such as cell polarization, pseudopod extension, excitation and adaptation [30, 68]. Hence newer chemotaxis models may need to incorporate intracellular mechanisms to model the newer findings at the intracellular level. Our model presented

here is a preliminary version of such an attempt. The simple receptor kinetics model presented in Section 5.2.2 is used as the intracellular mechanisms of the chemotaxis model.

Our chemotaxis model consists of a hypothetical motile cell population, which we shall refer to as M cells, and the chemoattractant molecules L (Ligand). An M cell is capable of both random movement and directed movement towards higher concentration of L . We describe the total displacement $\Delta \mathbf{p}$ over an arbitrary update interval Δt , as a linear summation of the random displacement $\Delta \mathbf{p}_{random}$, displacement due to local flow $\Delta \mathbf{p}_{flow}$, and the chemotactic displacement $\Delta \mathbf{p}_{chemo}$ that take place over Δt :

$$\Delta \mathbf{p} = \Delta \mathbf{p}_{random} + \Delta \mathbf{p}_{flow} + \Delta \mathbf{p}_{chemo} \quad \text{Eq. 5-16}$$

Eq. 5-16 is an instantiation of Eq. 5-5. $\Delta \mathbf{p}_{random}$ has an arbitrary direction (random movement). Its magnitude is the product of the update interval Δt and the random movement speed (a random value between 0 and 1). $\Delta \mathbf{p}_{flow}$ is the product of the local flow velocity $\mathbf{F}_p$ and time duration Δt . $\Delta \mathbf{p}_{chemo}$ converts local ligand concentration $C_{m,p}$ and gradient information $\mathbf{G}_{m,p}$ into a vector, which will be detailed below.

For a particular M cell, the update interval Δt is the period of time between two consecutive calls to its *move* method. We adopt the event-scheduling update scheme presented in Section 4.2 which simulates all the biological processes as a series of discrete events and allows arbitrary lengths of time between events. For efficiency reasons, the discrete event scheduler allows us to incorporate a simple negative feedback loop between the update interval Δt and the cell's movement speed. Specifically, we assume the following relationship:

$$\Delta t_{n+1} = 5e^{\frac{-\Delta \mathbf{p}}{\Delta t_n}} + 1 \quad \text{Eq. 5-17}$$

That is, the slower the current movement speed ($\frac{-\Delta \mathbf{p}}{\Delta t_n}$), the less likely that the cell will experience rapid movement in the near future, hence the next update interval Δt_{n+1} can be set longer to reduce the computational cost. Using Eq. 5-17, the update interval ranges from 1 to 6 (unit: minute).

For the chemotactic displacement $\Delta \mathbf{p}_{chemo}$, we adopt the quantitative hypothesis that chemotactic

response is proportional to the difference in newly bounded receptors on the front and the rear of the cell [29]. The equations for receptor-ligand binding, receptor internalization and receptor recycling have been described in Section 5.2.2 (Equations in Table 5-1). Here we extend its formulation to determine the effect on chemotactic movement. Let r and V represent the radius and the volume of an M cell, and let $[R]$, $[LR]$, $[R^*]$ represent the quantity of the free surface receptors, occupied surface receptors, and the internalized receptors respectively. The amount of ligand available at the front and the rear of the cell is approximated by:

$$[L]_{front} = \frac{(C + |G|r)V}{2} \quad \text{Eq. 5-18}$$

$$[L]_{rear} = \frac{(C - |G|r)V}{2} \quad \text{Eq. 5-19}$$

The number of newly bound receptors (LR) for the period Δt at the front and rear of the cell are:

$$\Delta[LR]_{front} = K_b [L]_{front} [R]_{front} \Delta t \quad \text{Eq. 5-20}$$

$$\Delta[LR]_{rear} = K_b [L]_{rear} [R]_{rear} \Delta t \quad \text{Eq. 5-21}$$

where $[R]_{front} = [R]_{rear} = [R]/2$ by assuming the free receptors are equally distributed on the cell surface.

Recall from Eq. 5-10 that K_b is the binding rate of the ligand and the receptor.

From Eq. 5-20 and Eq. 5-21, the difference in newly bounded receptors in Δt is therefore

$$\Delta[LR]_{diff} = \frac{K_b ([L]_{front} - [L]_{rear}) [R] \Delta t}{2} \quad \text{Eq. 5-22}$$

Assuming linear dependency of the chemotactic displacement on the difference in newly bounded receptors gives rise to

$$\frac{\Delta \mathbf{p}_{chemo}}{\Delta t} = k \left(\frac{\Delta[LR]_{diff}}{\Delta} \right) \frac{\mathbf{G}}{|\mathbf{G}|} \quad \text{Eq. 5-23}$$

where k is a proportional parameter. The larger the value of k , the more responsive the cell movement is to the occupancy difference in newly bounded receptors at its front and its rear.

Figure 5-4 Illustration of the under-agarose assay (figure adopted from [32])

Figure 5-5 Sample results of the under-agarose assay (figure adopted from [196])

We conduct a simulation that mimics the Under-Agarose Assay, which was designed to investigate the *in vitro* chemotactic behavior of leucocytes (e.g., neutrophils, macrophages, and monocytes) [28, 32, 33]. The configuration of the assay is illustrated in Figure 5-4. The tissue culture dish is filled with Agarose (gel) and two wells are cut into the gel. One is filled with cells and the other is filled with a chemoattractant. The cells were described to have migrated towards the chemoattractant-containing well by squeezing under the gel. Figure 5-5 shows the results obtained from the laboratory experiment [196].

Figure 5-6 Visualization of simulated chemotactic movement

We construct a virtual environment that mimics the under-agarose assay, shown in Figure 5-6(a). The experiment set-up, parameter values, and the detailed explanation of the visualization can be found in Appendix C. Here we only highlight some key facts. Initially, the cell well (indicated by the red circle on the right) is filled with 1000 cells, and the chemoattractant well (indicated by the red circle on the left) is filled with attractant molecules at a concentration of 2500 particles per

20 μm ×20 μm area (for a total of about 1.2×10^6 molecules). Panel (a) in Figure 5-6 shows the top view of the cell distribution (cells as black dots) and molecule distribution (green corresponds to high concentration and white corresponds to low concentration). The shape of the cell migration pattern is measured by five parameters: x_{min} , the population rear; x_{max} , the population front; x_{avg} , the average position of the population; L_x , the width of the spread, and L_y , the length of the spread. Panel (b) and (c) show the cross-sectional view of the cell histogram, molecule concentration, gradient, and relative gradient.

Figure 5-7 Simulation results at $t=5000$ with different k values

Table 5-2 Cell migration parameters at $t=5000$

k	x_{avg}	x_{min}	x_{max}	L_x / L_y
3	1018.98	409.29	1644.49	1.275
10	1055.59	525.61	1990.38	1.375
25	1130.81	553.06	2622.43	1.604
50	1235.53	482.27	2858.61	1.380

As defined in Figure 5-3, k is the proportional parameter that relates a cell's chemotactic displacement with the difference in newly bound receptors between this cell's front and rear.

Therefore, the larger the k , the larger the chemotactic displacement can be expected. We test with four k values: 3, 10, 25, and 50. The simulation results are shown in Figure 5-7 and the measurable parameters are shown in Table 5-2. The model produces a biased random walk pattern which is qualitatively close to the laboratory observation (Figure 5-5) for low k values. Besides, following observations are made:

- The larger the k , the greater the displacement of the cells towards the chemoattractant. However, it can be seen from Figure 5-7 that the cell population spreading towards right is sparsely distributed. The majority of the cell population is still concentrated in the cell well. This is further confirmed by values in
- Table 5-2. The cell population front $x_{\max}$ significantly increases with the increase of k , but the population center x_{avg} only slightly increases on a much smaller scale than $x_{\max}$.
- The pattern elongation along the y axis is evident from both the figures and the measurement of the parameters (i.e., $L_x/L_y > 1$) in Table 5-2.
- As for the cell population rear $x_{\min}$, it decreases as k increases, that is, the cell population rear moves towards left.
- The molecule distribution exhibits a horseshoe shape, and its shape coincides spatially with the edge of the cell population. The area occupied by cells is low in attractant concentration, reflected by their lighter background.

The above observation can be attributed to the interactions between cells and attractant. As defined by the agent movement rules, the receptor-ligand binding leads to internalization of the ligand (attractant), effectively reducing the local concentration of the attractant. The cell population thus falls in a basin of the attractants concentration created by the cells themselves. The cells spread towards left due to *chemotactic* movement (as opposed to *random* movement), as the gradient at the population rear is pointing to the left.

Figure 5-8 shows the measurable parameters for $k = 10$ and 25, from $t = 0$ to $t = 5000$. The plots show the consistent increase of $x_{\max}$ (population front) and the consistent decrease of $x_{\min}$ (population rear). The pattern elongation L_x/L_y gradually increases then decreases, implying that

the spread along y axis becomes relatively more significant at the later stage of the simulation.

Figure 5-8 Cell migration parameters: (a) $k = 10$ (b) $k = 25$

To further investigate the correlation between the cell population distribution, attractant concentration and gradient, we plot the snapshots of the simulation for $k = 25$ at $t=1000$ and 5000 , and compare them against the scenario where no cells exist in the environment, as shown in Figure 5-9. The distinctions in the smoothness of the gradient and relative gradient between the two scenarios are evident. When there are no cells, the molecular distributions and the landscape of the gradient and relative gradient are smooth, as the molecular distribution is only affected by the diffusion equations and no other factors. However, with the presence of cells, the landscape of the gradient and relative gradient become rugged, reflecting the local interactions between cells and molecules (that is, the cell receptors bind and consume the molecules in the local space). At $t=1000$, the front of the cell population towards the chemoattractant is evident, and it spatially coincides with the peak of the gradient. The negative relative gradient starts to appear at the population rear, indicating that the gradient at that point is pointing towards left, that is, the cells at the population rear will be attracted towards left instead of right. This is in contrast to the scenario when cells are absent, where the gradient is always positive on the left to the attractant well (i.e., the gradient is pointing towards right). For $t=5000$, almost half of the cell population falls in the area of negative gradients (which implies that cells at the rear are moving towards left) and the other half fall in the positive gradient (i.e., cells at the front are moving towards right). When compared to the scenario where the cells are absent, it can be seen that the gradient landscape is different, and the molecular concentrations are much lower. This further confirms that the basin-like distribution of molecular concentrations is indeed caused by

the cells' consumption of ligand molecules within the environment.

Figure 5-9 Snapshots of the cell migration pattern, cell histogram, attractant concentration and gradient at $t=1000$ and $t=5000$ with $k = 25$

We further study the long-term simulation results by extending the simulation to $t=20,000$ and $t=100,000$. The results are shown in Figure 5-10. It can be seen from the cell distribution (panel (a)) as well as the cell histogram (panel (b)) that the cells are further spread over the space. The attractant concentration becomes so low that their effects on the cells are almost negligible. In the long run, cell movements become independent of attractants and collectively, exhibit a normal diffusion pattern.

The laboratory experiment for the Under-Agarose assay typically takes at least 2-4 hours for

neutrophils (not counting the time for preparing materials and solutions), and can take as long as 18 hours for slower cells such as monocytes [32]. The ABM simulation with hybrid entity representation has the obvious advantage that it completes much faster, in the order of minutes. Our present model omits much of the intracellular mechanisms, but with the hybrid modeling approach, improvements on both the environment level (such as compartmentalized environment in the in vivo system) and individual level (such as adding intracellular signaling pathway that converts receptor-ligand binding into cell locomotions), can be made as and when increased granularity is justified by the hypothesis in question. In Chapter 6, we further this case study as an example for the proposed agent behavior network model and iterative strategies for model extension.

Figure 5-10 Long-term results at $t=20,000$ and $t=100,000$ with $k=25$

5.4. Case study 2: airborne transmissions and faulty ventilation systems

In this Section, we apply the hybrid entity representation and the layered environment model to a case study of airborne disease transmissions. Airborne diseases are caused by pathogens which can remain in the air for long periods. They can be transmitted through coughing, sneezing, laughing or through close personal contact. For example, influenza is typically transmitted through the air by coughs or sneezes, creating aerosols containing the virus, though it also can be transmitted by direct contact with bird droppings or nasal secretions, or through contact with contaminated surfaces [197]. SARS (Severe acute respiratory syndrome) is primarily transmitted by bio-aerosol droplets or direct personal contacts [198], but airborne transmission was found to be possible [199].

Nosocomial airborne transmissions are influenced by the effectiveness of the ventilation systems [200, 201]. The ventilation system controls the space temperature and humidity, dilutes airborne bacterial contamination, controls air movement for minimizing the transfer of airborne bacteria [202]. For example, the isolation rooms need to keep a negative pressure so that air flows inward but not outward, to prevent leakage of the air in the room into the corridors [200]. A faulty or ineffective ventilation system would have its functionality reduced, or even lead to increased risks of airborne disease transmission.

Figure 5-11 A hypothetical ward environment

In this Section we build a hypothetical ventilation system within an imaginary hospital environment. The basic hospital model we have used is ASINE (stands for Agent-based Simulator for Infections in the Nosocomial Environment) [203], which includes an ABM implemented based on MASON [84]. This model includes basic types of human host agents: patients, nurses, and physicians. Figure 5-11 shows part of the environment, which is a ward that consists of four 6-bedded patient rooms, an activity room, and a nurse station. Each patient room has a toilet. The activity room has several seats and two toilets.

We assume that there exists a ventilation system with following properties: it consists of ducts that run across the building to cover all the ward areas. It has outlets that connect to the open space in

the ward and their function is to absorb and filter away pathogens in the air from the ward. Conceptually, this environment is made up of two distinct compartments:

1. The *ward compartment* where objects such as seats and beds are held, and human hosts move and perform their activities. Since we only consider the two-dimensional movements of the agents, we treat this compartment as a two-dimensional open space.
2. The *duct compartment* where the absorbed air is transported. Since the flow in the ducts is mostly unidirectional, we treat this compartment as a one-dimensional space with a fishbone like topology (see Figure 5-12).

Figure 5-12 The hypothetical ventilation system in the ward environment

Figure 5-13 Connection between 2 compartments via outlets

The two compartments are connected at the points of duct outlets, as illustrated in Figure 5-13. The Connection Layer of the ward compartment contains the connecting points that represent the channels

to the ventilation outlets. Each connecting point imposes a local air attraction field around it, implemented by a flow field that contains vectors pointing towards the connecting point. The vectors' magnitudes decrease with the increase in distance from the connecting point. The connecting points in the Connection Layer of the duct compartment shares the same locations of the outlets. The Connection Layer and Flow Field Layer together create the airflow pattern where the air in the ward compartment is attracted into the duct compartment via duct outlets.

Figure 5-12 shows the overall topology of the hypothetical ventilation system in relation to the ward environment. The grey arrows indicate the Flow Field Layer of the duct compartment, that is, the direction of the airflow in the ducts. We further assume that each outlet, shown as square green or black node in the figure, has the following properties:

- Absorbability (value ranges from 0 to 1). The weaker the absorbability (when the value approaches 0), the smaller the magnitude of the attraction field it creates around it in the ward compartment.
- Leakage (Boolean values: true or false). If there is no leakage (value = false), the inter-compartment channel at the outlets is always unidirectional from ward compartment to the duct compartment. If leakage exists (value = true), inter-compartment channel at the outlets is bidirectional, hence the air in the duct compartment may flow back to the ward compartment.
- Filtration efficacy (values range from 0 to 1). At the full efficacy (value = 1), the filters at each outlet eliminate a proportion (set at 100% for leaf outlets, and 10% elsewhere) of the pathogen in the governing space per time unit. The elimination probability proportionally decreases with the decrease in the filtration efficacy.

Perfect outlets are shown as dark green nodes (except for leaf outlets which are in light green) in the visualization. Imperfect outlets (namely, absorbability < 1, or leakage = true, or filtration efficacy < 1), the outlet is considered faulty and shown as a black node.

In this model, human hosts are modeled at the individual level as *agents* in the Agent Holder Layer. Airborne pathogens, which are much smaller in physical size and more numerous in quantity, are modeled as population quantities in the Population Quantities Layer. Airborne pathogens are driven by the airflows specified in the Flow Field Layer. In the simulation, we randomly pick one patient to

be infected with the airborne disease. She intermittently coughs or sneezes, during which the new pathogen quantities are created and spread into the nearby environment. The pathogen quantities are shown in red when they are in the ward compartment, and in orange when they are in the ventilation ducts.

Figure 5-14 Simulation result – without ventilation system

Figure 5-15 Simulation result – with working ventilation system

Figure 5-14 shows a situation where no ventilation system exists in the ward. In such a situation, the pathogens are subject to random movement only (i.e., diffusion) but have no directional movement due to airflows; they are also not cleared. The infected patient contaminates her local environment when she is in bed (Figure 5-14a), and the patients staying next to her become direct victims. When this patient moves towards the activity room (Figure 5-14b), she leaves a trace of the airborne pathogens and those who follow her are likely to be infected.

Figure 5-15 shows the instance where a working ventilation system is in place. In Figure 5-15a, the infected patient still contaminates her nearby environment (shown as red spreads), but the pathogens are quickly absorbed in to the ventilation conducts (shown as orange spreads along the ventilation ducts) and then eliminated by the filters. The same scenario happens when the patient moves along the corridors, as shown in Figure 5-15b.

Figure 5-16 Simulation result – with faulty ventilation system

Finally we model a scenario with a faulty ventilation system. The faulty outlets are shown as black nodes in Figure 5-12. Part of the outlets in the top-middle patient room, and the outlets in the activity room are faulty, in that they have weaker absorbability, leakage, and ineffective filters. Figure 5-16a shows that the patients at the downstream of the leaking ventilation duct become victims of the infected patient whose bed is at the upstream of the ventilation duct. More pathogens reside in the ventilation ducts due to the ineffective filters. In Figure 5-16b, the activity room that the infected patient has just visited is contaminated with airborne pathogens, which cannot be cleared in time due to the faulty ventilation system.

The ventilation system model presented here is intended as a showcase of using the hybrid entity representation in modeling the nosocomial transmission of airborne diseases. The disease model used here is much simplified, for example, only 2 health states are modeled for a patient – infectious (spreading pathogens through coughs and sneezes at random time points) or non-infectious. The disease transmission does not take into account the human factors such as age or varied susceptibility.

With limited research on the working mechanisms of the ventilation systems and their relations to the disease spread, we are not in a position to declare the realism of this simplistic model. However, if given the chance to collaborate with hospitals on such a subject, we believe a credible ABM can be developed to facilitate experiments and analysis of airborne transmissions and the role of ventilation systems.

5.5. Summary

In this Chapter we focused on the two related tasks: *hybrid entity representation* that targets at bringing down the computational complexity of ABMs without sacrificing the model realism, and the *layered environment modeling* that aims at a realistic environment representation and supporting structure for hybrid entity representations. The population level representation and equation-based behavior descriptions are traditionally associated with DE based models (see Section 2.2.1), but we have shown how they can be incorporated into ABMs to achieve proper entity relative scales, computational tractability, and performance. The hybrid entity representation is realized at two levels: globally in the environment, and locally with an agent. *Agents*, the entities modeled at the individual level, and *population quantities*, the entities modeled at population level, need to coexist and interact with each other in one environment. This is supported by Agent Holder Layer and Population Quantity Layer(s) in the environment model. The environment model is stratified into 4 functional layers; the other two are the Connection Layer, that specifies how multi-compartments can be connected, and Flow Field Layer(s), which specify the background flows that systematically transport population quantities and agents. We have demonstrated the efficacy of this approach with two case studies. The first is a chemotaxis model that specifies biological cells as agents and chemoattractants as population quantities; this model successfully produces cell migration patterns that are qualitatively comparable to the observations of an Under-Agarose Assay. The second is a model of nosocomial airborne transmission, where humans are modeled as agents and pathogens are modeled as population quantities. Its environment has two connected compartments: the open space and the ventilation ducts. The model is used to predict the situations of absent, working, or faulty ventilation systems. Though the settings of these two cases studies are relatively simple and they lack validation from domain experts due to our limited resources, they can potentially be expanded to more complete ABMs for computer experiments when there are chances of such collaborations.

Chapter 6

Modeling Agents for Extensibility

Chapter 4 and 5 have laid out the basics for *model realism* in time, space, and entity representation. In Chapter 3, we described another desired property for computer experiment oriented ABM, which the *model extensibility*, that is, to revise the model with wider scope and greater granularity with ease. Choosing the level of detail for a model is challenging if one considers that there are possibly infinite number of details that one can add to the model. Attempting to build models that are truly realistic duplicates of the real world system is impractical. Simplifications and abstractions are necessary for a computationally tractable model that enables the researchers to interpret, analyze, and understand the model structures and results. Based on Occam's Razor [41], simpler models are favored given that the explanation power is the same among the competing models; it advocates that model complexity should be increased only if increased explanatory power is traded for model simplicity.

For a computer experiment, two kinds of models need to be built (see Section 1.3): a *null computational simulation model*, or simply *null model*, that acts as a control group which represents the real world system without experimental manipulations; and one or more *experiment models*, that act as treatment groups which are modified from the *null model* by implementing experiment-specific changes. Determining the model scope and entity granularities for the null model in this context is especially challenging. Producing results that correlate with the reality is a necessary but insufficient condition for the null model. The null model has to tune its *model scope* (in the form of a set of entity types, see Definition 1-9) and *entity granularities* (in the form of the sets of entity state variables, see Definition 1-10) by taking into consideration the experimental requirements. As illustrated in Section 1.3.1 and 1.3.2, if the null model is insufficient in detail, such as too restricted in model scope or too coarse in entity granularity, the experiment model would not be able utilize it for the intended investigations or discoveries. If dramatic changes are made on the experiment model to make up for the missing details, it will be difficult to deduce which part of the change accounts for observed changes in the experimental outcomes.

The discussions in Section 1.3 lead to the observation that in the context of computer experiment, it will be normal to find null models inadequately detailed for the experiment; hence iterative revisions with incremental extensions to scope and granularity will be needed. Hence we set out to investigate the systematic strategy for *model extensibility* with two specific tasks which we will present in this Chapter: an *agent behavior network model* (or ABN) that targets at a formal method for describing agent states and behaviors; and the *iterative strategies for model extension*, that utilize ABNs to systematically extend models in terms of scope and entity granularities. We demonstrate the proposed approach with a case study of three chemotaxis models of increasing entity granularities.

6.1. Agent behavior network model

As defined in Definition 1-1, a computational simulation model is made up of a set of *state variables* (or *SVs*), a set of *behavioral rules* (or *BRs*) that operate on the *SVs*, and a set of constant parameters (or *Cs*). A more useful alternative view is defined in Definition 1-4 and Definition 1-5, based on the notions *entity instances* and *entity types*: the model can be seen as a collection of *entity instances*, where each possesses a subset of the models' *SVs*, *BRs*, and *Cs*. The *BRs* of an entity instance can be classified into local rules and interaction rules. A *BR* is a local rule if it only operates on the *SVs* of the concerned entity instance; it is an interaction rule if it involves *SVs* of other entity instances as well. *Entity types* specify the template definitions of *SVs*, *BRs*, and *Cs* that can be used to derive one or more entity instances. For example, a model may define a T helper cell *entity type*, from which several thousands of T helper cells (i.e., *entity instances*) can be created. In ABMs, an entity instance is equivalent to a stateful, autonomous agent. Hence we will use the terms entity instance and agent interchangeably in the following Sections.

6.1.1. Agent composition

In this Section, we take a closer examination of the agent composition. Formally, an agent is composed of:

1. A set of state variables $\{SV_i(t) \mid i = 1 \dots n\}$, that describe the state of the agent at any time point.
2. A set of behavioral rules $\{BR_j \mid j = 1 \dots m\}$ that describe how the state variables of that agent, and of other agents that it interacts, are changed over time. Each rule consists of:
 - a) *Inputs*, which is the set of state variables that the rule *reads*;

- b) *Outputs*, which are the state variables that the rule *writes to*; and
 - c) *Decision logic*, that determines how the rule translates inputs into outputs.
3. A set of constant parameters $\{C_k(t) | k = 1 \dots p\}$, that are tunable parameters for *SVs* and *BRs*, such as the length of the bit string that represents the T cell receptor, the mutation rate of the HIV virion, etc.

We use the well-known John Conway's Game of Life [204] for illustration. Each agent, which is a cell in the Cellular Automata, has one Boolean *SV* that takes two possible values: *alive* or *dead*. Each agent has 4 *BRs*, each of which reads 9 inputs: the agent's own *SV*, and the *SVs* of the 8 neighboring agents. Each rule writes 1 output, which is the agent's own *SV*. We re-write the rules below to illustrate the agent composition described above:

1. *Rule of under-population*: If the cell is alive (the rule reads the agent's own *SV*), and it has fewer than 2 neighboring cells that are alive (the rule reads other 8 agents' *SVs*), then the cell will be dead on the next time step (the rule writes to the agent's own *SV*). Here, the threshold of 2 neighboring cells is a constant parameter used by this rule.
2. *Rule of survival*: If the cell is alive, and it has two or three neighboring cells that are alive, then the cell will continue to be alive on the next time step.
3. *Rule of overcrowding*: If the cell is alive, and it has more than three neighboring cells that are alive, then the cell will be dead on the next time step.
4. *Rule of reproduction*: If the cell is dead, and it has exactly three neighboring cells that are alive, then the cell will be alive on the next time step.

The Game of Life is a type of *Synchronous Cellular Automata*, which means that it uses a synchronous, time-step based update scheme for executing the *BRs*, and a grid-based environment topology to define the closeness of the individual agents. The update scheme and environment are system-level modeling decisions that are beyond the scope of each individual agent. For example, the update scheme may be changed to an asynchronous update scheme with arbitrary visiting order [205]; the environment can be changed from square grids to hexagonal grids [206, 207]. We have discussed the update scheme and the environment model of ABMs in Chapter 4 and 5; here we will only focus on the agent-level states and behaviors.

Game of Life is a simplistic ABM in that there is only one *entity type*, with one *SV* of data type Boolean, four *BRs* with minimal computations, and a few constant parameters that set the thresholds of the number of living neighboring cells. Advanced ABMs may have much richer set of state variables for their agents, possibly using different data types. Take a T helper cell as an example. Below are the kinds of state variables of different data types we can possibly define:

- Activation state: a symbolic variable, which can take the values “Naive”, “Activated”, “Effector”, or “Memory”;
- Infection state: a Boolean variable that indicates if the cell is infected (true) or not infected (false);
- Specificity: a vector variable (such as a bit string) that represents the T cell receptor specificity in the kind of viral sequence it can recognize;
- CD4 receptor levels: a floating point variable that represents the amount of receptors available on the cell surface;
- Local IL1 level: a floating point variable that represents the amount of IL1 molecules available within the vicinity of the cell. The value of this variable is not locally stored in the agent; instead, the agent needs to sample it from the local environment;
- Physical position: an array of floating point variables that represents the x and y coordinates (and z coordinate if the model is three-dimensional) of the cell.

With the increased complexity in state variables, the corresponding agent behavioral rules may possess more sophisticated computational logic, including the use of equations. In general, a rule reads none or a few state variables as inputs, processes the information, and writes to one or more state variables as outputs. In the chemotaxis model presented in Section 5.3, the rule *chemotactic movement* reads the local gradient of IL-8 molecules (the rule input) and the current [x, y] coordinate of the agent (another rule input). Then this rule computes a displacement [dx, dy] as a chemotactic step towards the higher IL-8 concentration (the decision logic), and add it to the agent’s coordinates [x, y] (the rule’s output).

6.1.2. Agent behavior network (ABN)

Theoretically, an agent's behavioral rule may have multiple inputs and multiple outputs. A state variable may be associated with inputs and/or outputs of one or more rules. This motivates the graphical representation of the interrelationship between the state variables and rules of an agent, which we will term as the *agent behavior network*, or ABN.

In an ABN, state variables are represented as nodes called *state variable nodes*. The behavioral rules are also represented as nodes called *rule nodes*. If a state variable is an input of the rule, an *input link* exists from that state variable to the rule; if a state is an output of the rule, an *output link* exists from the rule to the state variable. Links only exist between state variable nodes and rule nodes; no link directly connects two rule nodes or two state variable nodes.

Figure 6-1 ABN representation of the chemotactic movement rule.

Figure 6-1 shows the example of the *chemotactic movement* rule and its links with the input and output state variables. The state variable nodes are represented by ellipses and the rule by a rectangle. The input links are denoted by unshaded arrowheads and the output link is indicated by a shaded arrowhead.

Figure 6-2 Example of a 3x2 agent behavior network

Figure 6-2 shows a possible ABN with 3 state variable nodes and 2 rule nodes or simply a 3x2 ABN. A link I_{ij} points from state variable S_i to behavioral rule R_j implies that R_j takes S_i as one of its inputs.

A link O_{jk} points from rule R_j to state variable S_k implies that the execution of R_j will update S_k . Here, state variables may be inputs for multiple rules (e.g., S_2) or outputs for multiple rules (e.g., S_3).

Since entity instances of a given entity type share the same definitions of SV s, BR s, and C s, it suffices to draw ABNs on the entity type level; that is, that ABN would represent the behavior network of all instances of a certain type of entity.

ABNs may seem to resemble a kind of rule-based system due to its use of state variables and behavioral rules. It is certainly possible to use a rule-based implementation of agent behaviors. However, we distinguish an ABN from the conventional rule-based system [208] as follows:

- The terms inputs and outputs of the rules used in ABN should not be confused with the conditions and actions as in a rule-based system. Inputs are merely state variables that the rule reads; inputs may appear in both conditions (for criteria checking) and actions (for computation) in the rule.
- When and in what order the rules are executed is determined by the discrete-event scheduler based on the timestamp associated in the event. Typically, if a rule responds to an external event (e.g., collision after the movement), it is invoked at the time of collision. If a rule represents an ongoing process in the agent regardless of conditions (e.g., quantity-based molecular dynamics such as receptor recycling to the cell surface), it is scheduled as a recurring event. An agent is allowed to register multiple recurring events with the scheduler at different update rate: e.g., *move* every t_1 time units, and *update receptor levels* every t_2 time units. The rule execution sequence is uniquely determined by the expected timestamps, and this event-based update scheme is asynchronous in nature. Therefore, the rule execution in ABN is distinctly different from the scheme used in conventional rule-based systems, which first put rules in an *agenda* and then resolve the conflicts.

An ABN might also be contrasted to a finite state machine (FSM). The state of an agent at a time is defined by all the state variables' values at that time. Hence the number of distinct states of an agent, which are the combinations of all the values that all the state variables could take, is potentially infinite, since state variables may take floating point numbers. The nodes in ABN graph represent

state variables and not state (where a *state* in a FSM comprises all state variables). The edges in the ABN do not represent state transitions either. Therefore, ABNs are different from FSMs.

6.1.3. Model detail verification with ABNs

An ABN provides the modeler with a parsimonious prescription of an agent's behavior. For example, it may be used to summarize the biological properties and behaviors of a type of cell from information scattered throughout research literature.

The ABN can be used for model details verification even before a single simulation is run. One can use it to inspect the completeness of the model dynamics, such as whether there are uninhibited, monotonically increasing state variables due to absence of negative regulation rules. For example, one can easily construct activation rules for lymphocytes (when a lymphocyte is activated upon initial recognition of antigen, it starts cell division for 2-4 days) from information obtained from standard texts in immunology [45]. With only the activation rule and no inhibition rules (that inhibit cell divisions or trigger programmed cell death) the number of cells will increase exponentially without bound. Similarly for molecules, both production rules (e.g., how the molecules are produced by cells) and destruction rules (e.g., how the molecules are degraded or consumed) must be specified; otherwise the molecular level would increase indefinitely. These runaway processes can be inspected from an ABN as follows. If one finds a state variable S_i for which all its updating rules O_{ji} have a positive effect on it, it is likely that the model is incomplete (in biological terms) and more rules should be added to make the dynamics complete (e.g., to specify rules that have negative regulation on the state).

6.2. Iterative strategy for model extension

The need of iteratively revising and extending model scope and granularity arises from the imperfect guess of appropriate model scope and granularity when a model is initially constructed. The complication imposed by computer experiment reinforces such a need. In this Section, we propose an interactive strategy that incrementally and systematically refines the granularity of agent models based on ABN representation.

Model extension takes place in two dimensions:

1. *In breadth in model scope*: This is when the model scope is expanded by adding new entity types. For example, to model syncytia formation in HIV pathogenesis, the new entity type syncytia needs to be introduced. Another way to introduce additional entity types is through sub-classification: two or more subtypes of entities are considered distinct enough to become two entity types, which will replace the original single entity type. For example, staff nurses and clerk nurses may be treated as a single entity type “nurse” at the beginning, and be split into two distinct entity types given that their behavior rules, in terms of work routines, are sufficiently different and require possibly different sets of state variables.
2. *In depth in entity granularity*: This is when the granularities of one or more entity types are increased by adding more state variables, or replacing state variables of simple data type with more sophisticated data types. For example, to model the ward round behavior of staff nurses, additional state variables are needed to record status information such as when the nurse needs to join the ward round. A coarse model may simply represent the disease status of a human host with a categorical variable, with values such as “healthy”, “colonized”, or “infected”. It is possible to replace it with a floating point value that represents the Colony-forming unit [209] quantities, if the modeler deems it necessary for a more accurate calculation of the chances of inter-host disease transmission.

Model extension in either dimension boils down to the changes of *SVs*, *BRs*, and *Cs* of the existing model. This results in a new model with possibly larger sets of *SVs*, *BRs*, and *Cs*. Complex changes sets can be viewed as a combination of two basic change patterns:

- 1) *Additions* (see Figure 6-3): One adds a new set of *SVs*, *BRs*, and *Cs* (shown in light blue; *Cs* are not shown), and links them up with the existing model by modifying a few *BRs* (shown with red outlines) that link up the existing model’s *SVs* and the newly added *SVs*.
- 2) *Replacements* (see Figure 6-4): One removes some *SVs*, *BRs*, and *Cs* from the existing model (shown as shaded nodes; *Cs* are not shown), resulting in one or more *BRs* that become incomplete because they depend on (either read or write) the removed *SVs* (shown with read outline). Then he adds the new, more granular *SVs*, *BRs*, and *Cs* as replacement (shown in light blue), fixes the incomplete *BRs* (shown in white with red outline), and links up new

BRs with the existing SVs (shown in light blue with red outline), if any.

Figure 6-3 Illustration of model extension by addition

Figure 6-4 Illustration of model extension by replacement

It is worth noting that for each type of changes, a large part of the original model remains and only a small part is changed, hence we term them as *incremental model changes*. Such changes do not strive for uniform model entity granularity; instead, they promote heterogeneous entity granularities across the model, in that the entities perceived to be more important with respect to the purpose of the study are modeled at a greater level of details, while others may remain at a relatively coarse level if a higher granularity is deemed unnecessary by the modeler. This is a desired property as it balances between the model realism and computational efficiency.

Based on the notion of incremental model extension, now we propose an iterative strategy that incrementally and systematically refines model granularities in the context of null model construction for computer experiment, shown in Figure 6-5.

Figure 6-5 Iterative strategy for model extension

The procedure starts with an initial, relatively coarse model M_0 : (SV_0, BR_0, C_0) , representing the modeler's initial guess on the model details that are likely to be sufficient for the study. Then the model enters the process of iterative revisions. For each iteration, the model is first checked against the experimental needs, in that whether the experimental manipulations required by the study are possible to be implemented on this model, without drastic changes to the SV s, BR s, and C s. If not, the model needs to be incrementally revised and enters the next iteration. If yes, we proceed to check the model details for runaway process with the help of ABN. Runaway processes are modeling artifact and should be fixed before proceeding to simulation. Once the runaway processes are eliminated, we conduct the simulation, and verify whether the results correlate with the reality. The validation criteria are subject to the purpose of the study, and may take a few forms:

- Qualitatively compare the results such as movement patterns, distribution of entities, with the illustrations in the literature.

- Qualitatively demonstrate the results to domain experts for opinions on the level of matching with the reality.
- Quantitatively compare the measurable outcomes such as flux of quantities, infection rate, change rate, concentration levels, with published values in the literature; or consult the domain expert on the heuristic ranges of these values.

The use of published facts in the literature and the domain expert's advice is an important part of this step. We believe that computer experiments are multi-disciplinary in nature and for them to be useful to real-life applications rather as toy models, modelers should be prepared to approach domain experts for advice, to show them the preliminary or intermediate results, and be willing to revise models towards the need of experiment. Some questions we as modelers asked the domain experts were:

- Do you think the graph / numbers derived from the simulations resemble the real-life observations? Are there any published figures we can compare with?
- What could be the reason for the deviation? Have we misinterpreted the behavioral rules? Have we set the input parameters wrongly? From which papers can we find more details on the process? Which papers contain values we can use for the input parameters?

Once the likely causes of the deviation between results and observations are identified, the model is incrementally revised and enters the next iteration. A model that passes the above checkpoints can be regarded as a sufficient null model for the subsequent computer experiment.

6.3. Case study: entity granularity extension in chemotaxis models

To demonstrate the agent behavior network and its use in model extension, we present a case study based on the chemotaxis model described in Section 5.3. In this Section, we focus on the model detail of how an agent (i.e., a biological cell) processes the chemoattractant concentrations and gradient that it senses and converts these inputs into cell movement. We construct three models with increasing model granularities (shown in Figure 6-6):

Figure 6-6 ABNs for three models of chemotaxis (a) Model I: saturated dependency on gradient (b) Model II: expanded with receptor kinetics (c) Model III: expanded with simple intracellular signaling pathway (specific to neutrophil)

1. Model I: Assumes the cell velocity has a saturated dependency on the ligand gradient.
2. Model II: Expands Model I by incorporation of receptor kinetics, including receptor-ligand binding, internalization, and receptor recycling. This model is much the same as the one presented in Section 5.3.
3. Model III: Expands Model II by including simple intracellular signaling pathways that are responsible for formation and extension of the lamellipodium and formation of the uropod (the cell tail), which collectively enable the cell to crawl towards the higher attractant concentration.

We describe each model in detail in the following Sections. We will use r and V to denote the cell radius and volume; Δt for the update interval (which is not necessarily uniform, as explained in Section 5.3), C_L and G_L for the ligand (chemoattractant) concentration and gradient, respectively.

6.3.1. Model I: Saturated Dependency on Gradient

Model I assumes that the cell movement consists of two components: a random displacement and a

chemotactic displacement component.

<p>R₁: Random Movement</p> <p>If alive then</p> <p> Get random angle θ, random step size d ($0 < d < d_{\max}$)</p> $\Delta \mathbf{p} = [d \cos(\theta), d \sin(\theta)]^T$	Eq. 6-1
<p>R₂: Chemotactic Movement</p> <p>If alive then</p> $\Delta \mathbf{p} = \frac{2r\mathbf{G}_L}{k_I + 2r \mathbf{G}_L } \Delta t$	Eq. 6-2

Listing 6-1 Agent rules of Model I

It can be seen from Figure 6-6(a) and Listing 6-1 that two rules affect the cell's position (x, y) . Rule R₁ computes the random displacement by specifying a random direction and step size (Eq. 6-1). Rule R₂ computes the chemotactic displacement. It takes the local ligand gradient $\mathbf{G}_L$ as input and converts it to a displacement vector (Eq. 6-2). This rule is based on the simple, intuitive assumption that the steeper the gradient, the faster the cell moves (the larger the step size per unit time). As the cells cannot move infinitely fast, the speed should “saturate” at a certain level. Based on Eq. 6-2, when $\mathbf{G}_L$ becomes large, the speed saturates at $1 \mu\text{m}$ per unit time. k_I is a parameter that controls the responsiveness of the cell: the smaller the k_I , the faster the cell moves.

6.3.2. Model II: Receptor Kinetics

Model II expands on Model I by replacing the simplistic chemotactic movement rule R₂ with a network of rules and state variables that model the ligand-receptor interactions. This is shown in Figure 6-6(b) and the detailed rule implementation is shown in Listing 6-2. $[L]$ represents the amount of ligand around the cell, which depends on the local ligand concentration C_L and cell volume V . $[R]$, $[LR]$, and $[R^*]$ are the amount of the free surface receptors, the bound surface receptors (filled with ligand), and the internalized receptors that are possessed by the cell, respectively.

Rules R_{2.1}, R_{2.2}, and R_{2.3} model the receptor-ligand interactions:

- R2.1 – Ligand-receptor binding. Ligands L bind to the cell receptors R to form the ligand-receptor complex LR. The rate of LR formation is proportional to the availability of molecules L and R, and the proportionality parameter is given by K_b . This rule reduces [L] and [R] but increases [LR].
- R2.2 – Receptor Internalization & Ligand Consumption. The complex LR is internalized into the cytosol, and the ligand is ‘used up’ or degraded by other biological processes. This rule decreases [LR] and increases the amount of internalized receptors [R*]. The rate of internalization is proportional to [LR] and the proportionality parameter is given by K_i .
- R2.3 – Receptor Recycling. Internalized receptors are returned to the cell surface, to be made available for binding again. It decreases [R*] and increases [R]. The rate of recycling is proportional to the amount of internalized receptors [R*] and the proportionality parameter is given by K_r .

Rule R_{2.4} (Eq. 6-11) is based on the hypothesis that the chemotactic response is proportional to the difference between the newly bound receptors at the front and the rear of the cell [29]. The derivation of the formula has been presented in Section 5.3. By introducing the interactions between receptors and ligands, the agent is able to create an impact on the local environment which was absent in Model I.

R_{2.1}: Receptor-Ligand Binding

If alive then

Get local ligand level: $[L] = C_L V$

$$\Delta[L] = -K_b [L][R] \Delta t \quad \text{Eq. 6-3}$$

$$\Delta[R] = -K_b [L][R] \Delta t \quad \text{Eq. 6-4}$$

$$\Delta[LR] = K_b [L][R] \Delta t \quad \text{Eq. 6-5}$$

$$\Delta C_L = \Delta[L] / V \quad \text{Eq. 6-6}$$

R_{2.2}: Receptor Internalization & Ligand Consumption

If alive then

$$\Delta[R^*] = K_i [LR] \Delta t \quad \text{Eq. 6-7}$$

$$\Delta[LR] = -K_i [LR] \Delta t \quad \text{Eq. 6-8}$$

R _{2.3} : Receptor Recycling	
If alive then	
$\Delta[R^*] = -K_r[R^*]\Delta t$	Eq. 6-9
$\Delta[R] = K_r[R^*]\Delta t$	Eq. 6-10
R _{2.4} : Chemotactic Movement	
If alive then	
$\Delta p = k_{II}K_b rV \frac{[R]}{2} G_L \Delta t$	Eq. 6-11

Listing 6-2 Agent rules of Model II

6.3.3. Model III: Intracellular Signaling Pathway

Model III expands Model II by replacing rule R_{2.4} (which implements the hypothetical relationship between chemotactic displacement and receptor binding differences) by more detailed intracellular signaling pathways that transduce the binding signal to cell locomotion through a series of molecule activations. Figure 6-6(c) shows the model extension while the rules are detailed in Listing 6-3.

R _{2.4.1} : G protein activation	
If alive then	
Get random angle θ , random step size d	
$\Delta[LR]_{diff} = K_b G_L rV \frac{[R]}{2} \Delta t$	Eq. 6-12
$p_{activate} = \frac{\Delta[LR]_{diff}}{\Delta[LR]_{diff} + 0.005}$	Eq. 6-13
If random number $\rho: \rho \in (0,1]$ < $p_{activate}$ then	
G = active	
Else	
G = inactive	
R _{2.4.2} : Cdc42 activation	
If alive then	
If random number $\rho: \rho \in (0,1]$ < p_{follow} then	
Cdc42 = G	
Else	
Cdc42 = inverse of G	

<pre> R2.4.3: Actin Polymerization If alive then If Cdc42 = active Poly-Actin = formed Else Poly-Actin = not formed </pre>
<pre> R2.4.4: Rho Activation //Similar to R2.4.2 </pre>
<pre> R2.4.5: Uropod Formation //Similar to R2.4.3 </pre>
<pre> R2.4.6: Chemotactic Movement If alive then If Poly-Actin = formed AND Uropod = formed Get random step size d $\Delta \mathbf{p} = \frac{\mathbf{G}_L}{ \mathbf{G}_L } d \Delta t$ Eq. 6-14 Else $\Delta \mathbf{p} = 0$ </pre>

Listing 6-3 Agent rules of Model III

The signaling pathways we implement here are specific to neutrophils (a type of white blood cells that are attracted to the site of infection within 30 – 60 minutes after the initial insult [30, 45]) The neutrophils migration process is analogous to amoeboid chemotaxis, which means that they crawl on a substratum or through a matrix by continuously extending thin, veil-like protrusions (called lamellipodia), adhering to the substratum, and releasing the tail (called uropod) [28, 30]. The signaling pathway is a complicated process which involves a chain of molecular activation, including receptor-associated G protein, PI3K (phosphatidylinositol 3-kinase), PIP₃ (phosphatidylinositol 3,4,5-triphosphate), GEFs (guanine nucleotide exchange factors), small GTP-binding protein (such as Rac, Cdc42, Rho) and their kinases, WASP/Scar family, and Arp2/3 complex [30]. Here we simplify the process to two main events: actin polymerization which enables the cell to polarize and extend the lamellipodium (the thin, veil-like protrusions at the leading edge of the cell), and MLC (myosin light

chain) mediated contraction so that the uropod (the cell tail) can be formed and hence the cell can be released from the substratum during migration.

We added 5 state variables to the agent model: G (G protein), Cdc42, Rho, poly-actin (actin polymerization status), and uropod (uropod formation status). All these state variables are binary, that is, they are either *on* or *off*. This is reflected in the rules that update them (R_{2.4.1} to R_{2.4.5}). The rules are similar to those used in the Random Boolean Networks (RBN) [210] where the state variables are purely Boolean. Essentially, the difference in the newly bound receptors on the front and rear of the cell is mapped to the probability of G protein activation (p_{activate}). The Cdc42 activation and Rho activation follows the activation status of G protein with probability p_{follow} . The cell is capable of chemotactic movement only if both the actin is polymerized and the uropod is formed. The chemotactic displacement is a vector along the direction of gradient with a random step size d ($0 < d < d_{\text{max}}$).

We have demonstrated how an agent model can be extended systematically with increased granularity with the help of ABN and how the model allows heterogeneity in state variables and behavioral rules. In next Section we present the experimental results of these chemotaxis models.

6.3.4. Experiment results

The experiment is to simulate the Under-Agarose Assay, using the same settings and measurable parameters defined in Section 5.3. Figure 6-7 shows the Model I simulation results at $T=2000$ and $T=5000$. Briefly, the middle and bottom panels show the cell histogram along the x axis (in grey), molecule concentrations (in green), x component of the molecular gradient (in pink), and x component of the relative molecular gradient (defined as gradient over concentration, in blue). A positive gradient points to the right. It can be observed that the cells migrate towards the chemoattractant with a relatively sharp front and elongated cell migration pattern. Since the cells only sample the concentration (and the gradient) of the attractant, but do not modify the attractant distribution by consuming the attractant molecules, the attractant distribution remains smooth throughout the simulation since it is only modified by the diffusion equations.

Figure 6-7 Model I simulation snapshot

Figure 6-8 Model I migration pattern measurements

Figure 6-8 plots the measurements of $x_{\max}$, $x_{\min}$, x_{avg} and L_x/L_y against simulation time. The population rear moves slightly towards the left ($x_{\min}$ decreases) due to the cells' random movements. The population as a whole (x_{avg}) moves towards the right. The population front moves ($x_{\max}$) towards the right before $T=3000$, and hardly moves after that since it reaches the concentration peak. The area on the immediate right of the concentration peak has a negative gradient (pointing to the left) hence preventing the cells from moving beyond the peak. The elongation of the cell pattern is reflected by L_x/L_y which is consistently greater than one (i.e., cell spread along axis is greater than cell spread long y axis).

Figure 6-9 Model II simulation snapshot

Figure 6-9 shows the Model II simulation results at $T=2000$ and $T=5000$. Figure 6-10 shows the corresponding migration pattern measurements over time. Model II differs from Model I in that 1) receptor kinetics is explicitly modelled; 2) Due to receptor-ligand interactions, not only are the cell movements affected by attractant distributions, the attractant distributions are also modified by the cells as well through receptor-ligand binding. The simulation results differ from that of Model I in that the cells spread (rather than concentrate) while moving towards the right. The attractant distribution exhibits a horseshoe shape, and its shape coincides spatially with the edge of the cell population. The area occupied by the cells is low in attractant concentration, reflected by their lighter background. This is because the receptor-ligand bindings lead to the internalization (consumption) of the ligand (attractant), which effectively reduces the local concentration of the attractant. The cell population thus falls into a horseshoe-shaped “basin” of the attractant concentration created by the cells themselves. The rear of the cell population pattern moves towards the left ($x_{\min}$ decreases) due to *chemotactic* movement, as the gradient at the population rear is pointing to the left. Though the population front moves faster to the right ($x_{\max}$ increases quickly), the majority of the cells remain in the cell well, which is confirmed by the fact that the population center (x_{avg}) moves at a much slower speed than the population front. The pattern elongation ($L_x/L_y > 1$) is evident at the start, but due to the spread in both x and y directions, L_x/L_y slowly decreases, which results in a more roundish pattern than that in Model I.

Figure 6-10 Model II migration pattern measurements

Figure 6-11 Model III simulation snapshot

Figure 6-11 shows the Model III simulation results at T=2000 and T=5000. Figure 6-12 shows the corresponding migration pattern measurements over time. The results are similar to Model II except that the cell pattern is consistently roundish throughout the simulation (i.e., L_x/L_y is close to 1). The cells are less concentrated in the cell well. In Model III, the chemotactic movement is modeled such that a higher gradient leads to a higher chance of signaling pathway activation (Rule $R_{2.4.1}$); however, the magnitude of the displacement after the pathway activation is independent of the steepness of the gradient (Rule $R_{2.4.6}$, Equation Eq. 6-14). The breakdown of the direct numerical relationship between chemotactic movement and molecular gradient contributes to the more roundish

cell pattern observed in Model III.

Figure 6-12 Model III migration pattern measurements

All three models exhibit the pattern of *biased random walk*, which is a defining characteristic of chemotaxis [36]. When compared to the laboratory observations (Figure 5-5), Model I does not correlate with the reality well in that the cells should spread instead of concentrate. Model III appears to have a closer agreement with the laboratory observation than Model II in terms of the shape of the cell migration pattern. This suggests that the extension of model granularities, namely, the receptor kinetics and signaling pathways does improve the model's realism, and as a result, improves its agreement with the laboratory observations.

6.4. Summary

Determining the appropriate model scope and entity granularities for ABMs is difficult, and the need of experimental manipulations in computer experiment applications further complicates the challenge. This raises the need of incrementally and progressively revising the model, in the form of widened model scope and increased entity granularities. We have observed that most ABMs in the literature do not explicitly address this important concern. For this reason, we have proposed an *iterative strategy for model extension* that allows incremental improvement to be made to model until it is sufficiently detailed to serve as the null model for the computer experiment. For each iteration, checks are done for experimental needs, runaway process, and the correlation of simulation result with reality. Even if the model is not built primarily for computer experiment, one can still apply

this strategy to incrementally refine model with higher level of details. The strategy relies on the *agent behavioral network model* (ABN), which formally describes an agent as a network of state variables and behavioral rules. ABN facilitates the specification of heterogeneous granularity levels across the model to balance the model realism and computational efficiency. The incremental changes on entity granularity are realized by a combination of addition and replacement operations on the existing ABN. We demonstrated the proposed approach with three models of chemotaxis of increasing granularities, and simulations are conducted to validate these models. Higher level of details, such as the molecular dynamics and even the intracellular signaling pathways are introduced to the model progressively with the help of ABN. Results show that the models' results approximate the reality more closely with the increased granularity.

Chapter 7

Computer Experiment I: Nosocomial Transmissions

From Chapter 4 to Chapter 6, we have devoted our effort in developing the building blocks of a modeling and simulation framework for credible and extensible ABMs, as laid out in Chapter 3. Equipped with this modeling and simulation framework, we are ready for applying ABMs to practical computer experiments.

In this Chapter, we present an application that focuses on how the hand hygiene adherence (or HH) and the staffing levels influence the dynamics of nosocomial MRSA transmissions [180]. The domain background of MRSA pandemics and their intervention measures have been described in Section 2.1.2 and will not be repeated here. Briefly, HH and staff cohorting are two intervention measures that believed to be impactful on the nosocomial MRSA transmission dynamics, but the actual implementation of these interventions is usually far from perfect. The motivation of the work is to analyze and predict the impact of HH adherence and staffing level, so as to answer the operational questions that concern the hospital decision maker, such as whether a low staffing level, due to shortage of nurses, will result in increased colonization rates.

In an environment such as SICU, staff cohorting is directly influenced by the staffing level. Staff cohorting is practiced through staff nurses where each staff nurse is assigned to one isolation room except when there is insufficient staff, then one staff nurse may be assigned to manage 2 or 3 rooms (or equivalently, patients). For other staff roles such as assistant nurses and physicians, they take care of all the patients (e.g., visit all the patient rooms in arbitrary sequence during ward rounds), so there are no dedicated patient assignments for these staff roles.

ABMs are suitable for this application due to the involved heterogeneity in human factors, such as individualized adherence rates, health risk factors, activities set, and movement patterns. Such an

environment is also highly stochastic, hence fluctuations due to individual differences cannot be ignored. For example, the existence of a super-spreader may drastically change the incidence levels. Human interactions in such an environment cannot be assumed to be uniform; for example, nurses and physicians are the main staff types who interact with patients, while interactions with visitors are relatively less. Almost all movement patterns of these staff are driven by task-oriented goals (e.g., to visit the patient) and daily routines (for healthcare workers, or HCWs). The movement patterns are non-uniform, location-based and highly individual. In this Section, we develop a null ABM for the MRSA transmissions in an SICU environment and conduct formal experiment design for investigating the effects of staff HH and staffing levels on nosocomial transmissions.

7.1. Model scope and methodologies

Our model is based on an ISCU ward with 12 single-bedded isolation rooms. The model is developed and analyzed in three broad stages: 1) Field profiling of human host activities and the environment, 2) Null model construction, verification, and validation, and 3) Experimental design and factor analysis for staff adherence rates and staffing levels. At the field profiling stage, we collect data about the environment and the daily routines of the HCWs, patients, and visitors in the SICU through direct observation, interviews, and the use of TrackMETM devices¹. We have summarized our findings in Section 2.1.2. The detailed description and analysis about this stage has been documented in a separate paper [211] and will not be discussed in detail here. With data from field profiling, we construct a null ABM that represents the existing state of the SICU. The model details, including the activities and movement patterns generated by the simulations were face-validated with hospital stakeholders through the use of visual animations of various group activities such as roll calls and ward rounds. The model has been revised for several rounds after the discussions with the domain expert, until its predicted disease transmission dynamics quantitatively matches the clinical observation, which is that around 25% of the patients are colonized at discharge.

Once the null model is validated, we proceed to conduct a full factorial experiment [212] to analyze the sensitivity of the outcome measure (patient discharge colonization rate) due to three factors under study; namely, the number of staff nurses allocated to each shift (i.e., the staffing level, denoted by F_1),

¹ Provided by ROSS Scientific Pte Ltd.

the nurse adherence rate (denoted by F_2) and the physician adherence rate (denoted by F_3). For nurse and physician adherence rates, 4 levels were tested: 20%, 40%, 60%, and 80%. For number of staff nurses, 3 levels were tested: 12 nurses (which translates to a 1-to-1 assignment to patients), 6 nurses (1-to-2 assignment), and 4 nurses (1-to-3 assignment). In total, this experimental design gives 48 design points. These settings are summarized in Table 7-1. We simulate the null model using this 48 design point experimental design. The result of the simulations provide us with a ranking (and its associated Confidence Interval and p values) of the factors and any confounding effects among them.

Table 7-1 Full factorial experimental design (3 factors and 3 to 4 levels per factor)

Factors	F_1: Staffing level	F_2: Nurse adherence rate	F_3: Physician adherence rate
Natural Levels	L_{11} : 12 staff nurses	L_{21} : 20%	L_{31} : 20%
	L_{12} : 6 staff nurses	L_{22} : 40%	L_{32} : 40%
	L_{13} : 4 staff nurses	L_{23} : 60%	L_{33} : 60%
		L_{24} : 80%	L_{34} : 80%

7.2. Null model descriptions

The role of the null computational simulation model is to provide a representation of the real ward environment and a control group that predicts the reference results in the absence of experimental manipulations. While it is impossible to describe every detail of the null model here, we highlight the modeling of some key processes as an attempt to clarify the level of detail and the kind of model realism we aim to achieve with ABMs. The key parameter values are summarized in Table 7-2.

Table 7-2 Key model parameters (partial listing): [V] single-valued [U] uniform distribution [E] exponential distribution [B] beta distribution – (min, likely, max)

Parameter	Value
Number of distinct visitors each patient has	[B] (0, 4, 10)
Number of visitations per visitation hours per patient	[V] 0.45 (afternoon hours) [V] 0.58 (night hours)
Number of visitors per visitation	[B] (1, 3, 5)
Length of visit (minutes)	[U] (5, 30)
Nurse HH adherence rate before entering the isolation room	[V] 60%
Nurse HH adherence rate when inside the isolation room	[V] 30%
Physician HH adherence rate before entering the isolation room	[V] 40%
Physician HH adherence rate when inside the isolation room	[V] 20%

The inter-arrival time of new patients (Days)	[E] 0.6
Length of Stay (LOS) for patients (Days)	[B] (1, 7, 60)
MRSA prevalence among the admitted patients	[V] 10%
MRSA prevalence among visitors (community prevalence)	[V] 0.1%
MRSA prevalence for HCWs who just enter the ward	[V] 25%
The sensitivity of the MRSA screening test	[V] 65%
The specificity of the MRSA screening test	[V] 95%
Waiting time for screening results (Days)	[V] 2
Actual treatment duration (Days)	[V] 21
Required treatment duration (Days)	[E] 18
Required time of transmission (minutes)	[E] 200

7.2.1. The SICU environment

Figure 7-1 shows the floor plan of the simulated SICU ward. In the centre of the ward is the nurse station (shown as connected grey rectangles). The 12 single-bedded isolation rooms are located around the nurse station and can be identified by the beds within them (shown as green rectangles). Outside each isolation room, there is a computer (labeled “PC” in the figure) where the nurses and physicians use to update the patient’s records. Besides isolation rooms, there are other rooms in the ward, such as the staff room, doctor’s room, toilets, nurse office, equipment rooms, meeting room, visitor’s lounge, clean utilities room and dirty utilities room. They are labeled in the figure.

The additional layer of the Waypoint network (shown as edges and nodes on top of the floor plan) denote the pathways that individuals in the simulation are likely to take. When a human host needs to move from one location to another, she plans the path using the Waypoint network, and then moves along the planned path. The red nodes denote the points where hand hygiene and wearing of contact precautions may take place. The main door (with an antechamber) is located between the dirty utility room and Room 13. Visitors usually enter and leave the ward via the visitor’s lounge, which is located at the bottom-right of the map. The HCWs enter and leave the ward via the staff door, which is located at the right end of the map.

Figure 7-1 The scaled floor plan of the SICU ward and a photo of activity outside room 10 (used with permission from the National University Hospital)

7.2.2. Behavior of healthcare workers, patients, and visitors

We have described the modeling of Patients and HCWs in detail in Section 4.4 (Figure 4-10 to Figure 4-13, and Table 4-2) as examples of the application of the event-scheduling update scheme. In summary, each patient is assigned a bed upon admission, and will remain bedridden until the date of discharge. The HCWs enter and leave the ward according to their assigned shift timings. When on duty, they carry out the mandatory activities at the specific timings, and carry out arbitrary activities if no mandatory activity is scheduled.

Each patient brings his/her own set of visitors (observed to range from 0 to 10 individuals, likely 4).

Visitors may enter the ward to visit the patients during the two visitation hours: afternoon from 1200h to 1400h, and during the evening from 1700h to 2000h. Figure 7-2 shows an overview of the visitors' activity execution. Visitors are recruited by a visitor shift manager and enter the ward during the specified visitation hours. Each visitor walks directly to the bed of the patient they are associated with, and remains there for some time (uniformly distributed between 5 minutes to 30 minutes) before leaving the ward (see Figure 7-2). The visitor is assumed to be touching the patient from the moment they arrive next to the patient's bed until they leave the patient. Once the associated patient is discharged, the same visitor is will not return to the ward.

Figure 7-2 Simulation of visitor activities (arrow indicates simulation time)

7.2.3. Adherence to hand hygiene

HCWs in our model are simulated to probabilistically adhere to WHO HH guidelines [213] and contact precaution practices in the SICU. In particular, we identify two types of HH opportunities, that is, the situations where hand hygiene should be performed. The sequence of events during a HCW's visit to a patient is shown in Figure 7-3. Firstly, before entering the patient's room, the HCW has to perform HH and wear contact precautions; this is a rule imposed on all staff in the SICU. Secondly, HH should be performed before procedures when HCW is in the room. For example, after some procedures which involve contact with the patient, HCW should perform HH. This may require removal of gloves and putting on a new pair after HH. Subsequently, more touch-surroundings events and procedures that involve patient touch may occur. Between each of these activities, HH needs to be performed. Before leaving the room, the contact precautions (i.e. mask, gown and gloves) are removed, and HH is performed again while HCW is still in the patient's room.

As summarized in Table 7-2, the null model uses differentiated HH adherence rates for the two types of HH opportunities. Nurses' HH adherence rate is set to 60% before entering the room, and 30% while inside the room. For physicians, the corresponding HH adherence rates are set to 40% and

20%, respectively. This is to reflect the observation and experts' experience that HH adherence is usually lower between procedures when inside the isolation room. For simplicity, we set adherence rates to the same value regardless of the type of HH opportunities in the full factorial experiments, as summarized in Table 7-1.

Figure 7-3 Illustration of simulated hand hygiene opportunities during a HCW's visit to a patient

Figure 7-4. Hypothesized probability distribution of touch events per HCW visit to a patient

We hypothesize a probability distribution of the number of touch events (be it to touch the patient or the patient's surroundings) for every HCW's visit to a patient, as shown in Figure 7-4. This distribution assumes that for most of the visits, there are likely 1 to 2 touch events between which HH are required. During the simulation, whenever a HH opportunity arises for a HCW, a probabilistic computation is made whether or not to comply with the HH procedure with the following algorithm:

1. Get HCW's current HH adherence rate, h .
2. Generate a random number x , from 0 to 1.
3. If $x < h$, then perform HH.
else skip HH.

7.2.4. Disease natural history and transmission

With respect to MRSA, the states in its natural history in our model are Susceptible (S), Colonized Asymptomatic (CA), and Colonized Symptomatic (CS) (illustrated in Figure 7-5). For HCWs and

visitors, they are assumed to be healthy and do not usually get infected due to colonization, hence in their natural histories, there are only the Susceptible (S) and Colonized Asymptomatic (CA) health states.

Figure 7-5 Disease Natural History for Patients, Staff and Visitors in our Model

During the simulation, the health state of a human host may change (i.e., state transition takes place) when certain conditions are met. Such changes take a finite amount of time to occur, which we will call transition time. A human host in the susceptible (S) state may get colonized (transit to the CA state) if he or she has direct contact with another colonized (CA or CS) host, thereby entering to the Colonized Asymptomatic (CA) state. The colonized individual then may spontaneously clear and re-enter the susceptible state, though spontaneous clearance is unlikely. We implement this by setting the transition time to 1 year, so that it is almost certain that some events will occur prior to that occurring, such as being discharged from hospital stay. For HCWs, they may be cleared of colonization (transit to the S state) by performing hand hygiene (this assumes that HCW's hands are the main output medium of MRSA transmission). For patients, they may get infected and develop symptoms (transit to the Colonized Symptomatic (CS) state) due to their immunocompromised health state. Treatment can be performed on CS patients for an average of 3 weeks. If the treatment is successful, the patients re-enter the CA state; otherwise, the patients remain in the CS state. Upon successful treatment, patients do not return back to the S state because colonization is difficult to clear and the treatments are unlikely to bring the patient to a completely clean (Susceptible) state, especially for the SICU where most colonized cases are long-term carriers.

Based on direct observations, most of the direct contacts take place between patients and HCWs, and between patients and their visitors. Therefore, we assume in the model that there are only patient-HCWs transmissions and patient-visitor transmissions through direct contact. We pre-determine the type of mandatory activities that involve direct contact through staff interviews

during the field profiling stage. Direct contacts among HCWs, and between HCWs and visitors are rare, though transmission may still take place via the fomites they are in contact with, such as tables, equipments, or keyboards. For simplicity, fomite based transmission is not modeled in this study. The averaged required contact time for the transmission to occur (i.e., to make the Susceptible party to be colonized) is set to 200 minutes.

The chance of transmission via direct contact can be reduced by wearing of contact precautions. We define the susceptibility (for a Susceptible host) and the transmissibility (for a Colonized host) during a particular person-to-person contact as:

$$\text{Susceptibility or Transmissibility} = 1 - b_{\text{gown}} - b_{\text{glove}} - b_{\text{mask}} \quad \text{Eq. 7-1}$$

where

$$b_{\text{gown}} = \begin{cases} 0.3 & \text{if gown is worn} \\ 0 & \text{otherwise} \end{cases},$$

$$b_{\text{mask}} = \begin{cases} 0.2 & \text{if mask is worn} \\ 0 & \text{otherwise} \end{cases},$$

$$b_{\text{glove}} = \begin{cases} 0.5 & \text{if glove is worn} \\ 0 & \text{otherwise} \end{cases}$$

The Transmission Likelihood r , between an Infector (colonized host) and an Infectee (susceptible host) is defined as:

$$r = \text{Infector's Transmissibility} \times \text{Infectee's Susceptibility} \quad \text{Eq. 7-2}$$

The algorithm is as follows:

1. Compute r with (4).
2. Generate a random number x , from 0 to 1.
3. If $x < r$, then the transmission starts.
Else the transmission does not start (is blocked).

7.2.5. Admission and periodic screening

Some patients are already colonized with MRSA at the time of admission. A screening test is performed on every admitted patient. As the test is imperfect, the simulated test result is obtained by the following procedure:

1. Generate a random number from 0 to 1.
2. If the patient is colonized then: if the number is smaller than the sensitivity of the test, then the test result is positive (true positive). otherwise the test result is negative (false negative).

3. If the patient is not colonized then: if the number is greater than the specificity of the test, then the test result is positive (false positive). otherwise the test result is negative (true negative).

The sensitivity and the specificity of the tests are set to 65% and 95%, respectively corresponding to the known efficacies of culture-based tests. The test result is usually available after 2 days¹. Therefore, the patient's colonization status remains unknown to HCWs until the results are returned. In the SICU, once a patient is known to be colonized, a label is posted on the door of the corresponding isolation room to indicate that the patient inside is colonized. This serves as a reminder to HCWs to adhere to the HH procedures when entering the room. In the simulation, if a HCW sees the MRSA+ label on the door before entering the isolation room, the HCW's adherence rate will be temporarily increased by 0.2 (capped at 1), and this is used to compute adherence to the HH and contact precaution procedures. After the HCW leaves the patient's room, the HCW's adherence rate is re-adjusted to the original level.

Besides the admission screening, the patient is also screened on a daily basis (based on frequencies observed in the administrative data) through opportunistic culture tests. Again, the test results are only available in 2 days. That is, at any time, the current colonization status of the patient known to the HCWs is based on the test performed on the patient 2 days ago.

7.2.6. Symptomatic colonization and treatment

We assume that HCWs will take action immediately once a patient develops clinical symptoms from blood stream infections (BSI). In the simulation, once a colonized asymptomatic patient develops symptoms (transits to CS state from CA), the treatment starts the next time he or she is visited by a HCW. The treatment is applied for a standard period of 3 weeks (21 days). The required treatment duration varies with each patient. We model this as a random number drawn from an exponential distribution with the mean of 18 days. The random number could fall below 21 days, or be greater than 21 days. The algorithm that determines the outcome of the treatment is shown below. For example, if the required treatment duration is 16 days (which is shorter compared to 21 days), at the end of 16 days, the treatment is considered successful and the patient transits to CA state. If the required treatment duration is longer than 21 days, say, 27 days, the treatment is considered

¹ Data provided by National University Hospital, Singapore.

unsuccessful and the patient will remain in CS state.

1. Generate a random number, called *required treatment duration*, from the exponential distribution with the mean of 18 (days).
2. Mark that the patient is undergoing treatment, and set the end date of treatment to be 21 days later.
3. After the time period of *required treatment duration*, check the following:
 - i. If the treatment is still going on (i.e., actual treatment time $\geq$ *required treatment duration*), the treatment is successful. Set the patient's disease state to CA.
 - ii. If the treatment has already ended (i.e., actual treatment time $<$ *required treatment duration*), the treatment is unsuccessful. The patient's disease state remains in CS.

7.3. Experimental results

In this Section, we present the simulation results of the null model and the experimental models. The former represents the default conditions of the SICU, and the latter incorporate the experimental manipulations that set hypothetical values for staffing and HH adherences. The simulations were conducted with the following configuration. Each simulation run of the model was replicated 100 times, each run comprises 720 simulation days (24 months). Data from the first month was discarded and the experimental results are based on the remaining 23 months. The simulations were run on a server running Ubuntu Hardy 8.04, on a quad-core 2.88GHz intel processor with 8GB of main memory. Each replication took on average 3.47 minutes (with standard deviation of 10.78 seconds) to complete.

7.3.1. Null model simulation results

The null model is an analogy of a control group that represents the cross-sectional state of the SICU and its interventions in a prospective cohort study design. A real-time animated visualization was shown to the hospital stakeholders to verify that the simulated daily routines and movement patterns in the ward were reasonably credible [214]. Figure 7-6 is a screenshot of a sample simulation run. It shows the scene on simulated Day 5 at around 9am where the ward is fully occupied (since all the 12 beds are occupied by patients). The ward round activity is taking place at Room 4, where some physicians have entered the room to check on the patients, while others are waiting outside the room. Such a pattern was directly observed during the field profiling stage.

Figure 7-6 Screenshot of the simulation on day 5 (morning ward round)

To create a credible model of the spread of MRSA within the SICU, we tuned the model parameters by matching the simulated outputs with the observational data. The final set of values used in the null model has been summarized in Table 7-2. The observational data to be matched and corresponding results are:

1. *The patient discharge colonization rate*, which is defined as the percentage of patients who are in the colonized state at time of discharge. It is calculated from each replication, as the total number of discharged *and* colonized cases over the total number of discharged cases that take place from month 2 to month 24. The simulated output colonization rate is 27.01% (standard deviation or std: 1.47%), which is close to the observed value of 25% estimated by retrospective studies.
2. *The occupancy level*, which is calculated from each replication, by averaging the number of occupied beds (out of the 12 beds) at the end of every simulated day from day 31 to day 720 (690 days). The simulated occupancy level is 10.29 (std: 0.13). This agrees with the field observation that occupancy level in the ward is around 10 beds most of the time. Out of the 690 simulation days, there are on average 7.5 days (std: 4.35 days) when the occupancy level drops below 6. This agrees with the observation that the occupancy level rarely drops below 6.

3. *Prevalence of hospitalized MRSA+ patients*, which is defined as the average daily number of patients who are in a colonized state (CS or CA) from day 31 to day 720 (690 days). During field profiling, around 2-3 patients were observed to be colonized (evidenced by the notice of contact precaution that was posted on the door of their isolation rooms to indicate that the patient within the room was a MRSA+ case). From the simulation, an average of 2.21 patients were in colonized state daily; in which 2.15 (std: 0.16) were in colonized asymptomatic state, and 0.06 (std: 0.02) were in colonized symptomatic state. Our patient prevalence data agreed well with the observations.

Figure 7-7 shows the daily patient occupancy levels over 690 days based for a single simulation run. It shows that the occupancy is around 10 and rarely drops below 6; the occupancy is capped at 12 since there are in total only 12 beds. The colonized patient count varies from 0 to 7, and most of the time it is around 2.

Figure 7-7 Patient occupancy level for a single simulation run of 690 days (day 31 to day 720)

Over 690 simulation days, there were a total of 224 incidences (std: 11.89), where 82.07 (std: 8.79) or 37% were due to (1), 140.99 (std: 11.28) or 63% were due to (2), and 0.94 (std: 0.95) were due to source (3). Therefore, transmissions due to contact with visitors were relatively rare and transmissions from staff were found to be the major causes of nosocomial infections.

7.3.2. Full factorial experiment results

For conciseness, we will only present the summarized results of the effect of nurse, physician adherence rates, and staffing levels on the nosocomial transmissions in this Section. Interested

readers may refer to our paper [180] for more detailed descriptions on the experiment results.

The full factorial simulation experiment tests 4 HH adherence levels (20%, 40%, 60%, 80%) for *both* nurses and physicians, and 3 staffing levels (1-to-1, 1-to-2, 1-to-3), which gives rise to $4 \times 4 \times 3 = 48$ design points. For ease of exposition, let DP_{ijk} denote the design point with the following levels chosen for the factors: $F_1=L_{1i}$, $F_2=L_{2j}$, and $F_3=L_{3k}$. For example, DP_{132} represents the design point where staffing level is set to 12 staff nurses (L_{11}), adherence rates for nurses and physicians are set to 60% (L_{23}) and 40% (L_{32}) respectively. For each design point, the simulation is replicated 100 times and each replication comprises 720 simulation days (of which, 690 are used for analysis). Let $\mathbf{D}_{ijk}$ denote the dataset of the output measures (i.e., patient discharge colonization rates) from all the replications of the design point DP_{ijk} . The 100 data points (derived from 100 replications) for dataset $\mathbf{D}_{ijk}$ are denoted as $d_{ijk,1}, d_{ijk,2}, \dots, d_{ijk,n}, \dots, d_{ijk,100}$.

Table 7-3 Change in patient discharge colonization rate due to 20% increases in nurse adherence rates

Increase in nurse's HH adherence:	20% -> 40%	40% -> 60%	60% -> 80%
+Changes in patient discharge colonization rate: mean (std)	-21.92% (1.03%)	-10.11% (1.16%)	-5.31% (1.20%)
95% confidence intervals of change in patient discharge colonization rate due to increase in nurse's HH adherence
Legend:			
<div><div>—●— HH: from 20% to 40% (j=1)</div><div>—●— HH: from 40% to 60% (j=2)</div><div>—●— HH: from 60% to 80% (j=3)</div></div>			
+ Change in patient discharge colonization rate: positive value represents increased colonization rate with increased nurse adherence rate; negative value represents decreased colonization rate with increased nurse adherence rate.			

The increase in nurse adherence rates and physician adherence rates both reduce the patients' discharge colonization rates, where the increase due to nurses' improvement in adherence is much more evident quantitatively. Table 7-3 and Table 7-4 summarize the impact of 20% point increases in

nurse and physician adherence rates on patient discharge colonization rates. By “20% point increase in adherence”, we mean an absolute change of 20% in the adherence rate value, e.g., when it changes from 20% to 40%, or when it change from 40% to 60%. For Nurses, with other parameters fixed (see Table 7-3), the decrease in colonization rate ranges from 5.31% to 21.92%. For Physicians; the decrease in colonization rates ranges from 0.69% to 3.03%. The statistical and clinical significances of the decreases are shown by the 95% confidence intervals in the table (the boundary values of the confidence intervals are given in our paper [180]). For a 20% increase in nurse adherence rates, all the confidence intervals are far below zero, which implies that the decreases are both statistically and clinically significant. For a 20% increase in physician adherence rates, the confidence intervals are generally below zero but not far from it. For HH increase from 40% to 60% and from 60% to 80%, some confidence intervals include zero. This implies that though the decreasing trend in colonization due to an increase in physician’s HH adherence is clear and can even be verified to be statistically significant, the change is not clinically significant because they are very close to zero.

The strong effect of nurse adherence on the output colonization rate can be explained by the frequent contacts between patients and nurses, especially staff nurses. Further analysis (not shown here) reveals that, 82.48% of patient-staff contact time occurs between patients and nurses; while for patient and physician contacts, they contribute only 13.62% of the total patient-staff contact time.

Table 7-4 Change in patient discharge colonization rate due to 20% increases in physician adherence rates

Increase in physician’s HH adherence:	20% -> 40%	40% -> 60%	60% -> 80%
Changes in patient discharge colonization rate: mean (std)	-3.03% (0.62%)	-1.07% (0.33%)	-0.69% (0.22%)
95% confidence intervals of change in patient discharge colonization rate due to increase in physician’s HH adherence Legend: <div> <div>●</div> HH: from 20% to 40% (k=1) <div>●</div> HH: from 40% to 60% (k=2) <div>●</div> HH: from 60% to 80% (k=3) </div>			

We model the effect of changes in staffing level by changing the number of staff nurses available in the simulation. This is because staff nurses are the largest HCW subpopulation in the SICU ward (ideally and maximally, 12 per shift) and the ideal assignment strategy is for each staff nurse to be assigned to 1 patient only so as to reduce the cross transmissions due to one staff contacting multiple patients. In all, we include three staffing levels in this study: 12 staff nurses, 6 staff nurses and 4 staff nurses corresponding to 1-to-1 assignment, 1-to-2 assignment and 1-to-3 assignment of patients respectively. Intuitively, we *expect* a decreased staffing level would increase the colonization incidences due to two reasons:

1. Reduced staff cohorting leads to shared staff among patients [215] since a MRSA+ patient transmits to another patient via the common staff nurse, and
2. Increased workload causes the staff nurse to be less adherent to HH compliance policies [124]

Figure 7-8 Disease spread graph shows transmission routes among individuals¹

¹ Data are provided and owned by ROSS Scientific Pte Ltd.

For the first reason, supporting evidence can be found since the transmission incidences in the simulations are logged and can be visualized in the form of a disease spread graph (see Figure 7-5). It is then a simple matter to trace the transmission routes to verify whether the transmissions via a common staff nurse had indeed increased due to change in the staff-to-patient assignment.

The second reason implies a correlation between the staffing level and adherence rates; such that low staffing level is translated into lower adherence rates, which in turn influences the patient discharge colonization rate. Lacking estimated values for this correlation from literature, we hypothesize a simple negative linear relationship where a staff nurse's adherence rate drops by $x\%$ for every extra patient assigned to her. We test four settings for x :

(a) $x = 0$ (b) $x = 5$ (c) $x = 10$ and (d) $x = 15$.

As an example, if the base adherence value is 50% and the staff nurse is assigned 3 patients (where 2 are extra), then under setting (d), his or her adherence drops to $50\% - 2 \times 15\% = 20\%$.

From the simulation results, it is interesting to find that decreasing staffing levels do not have a uniform effect on the patient discharge colonization rate as commonly thought. For zero correlation between staffing level and adherence rate, the observed trend is that average patient discharge colonization rate decreases with decreasing staffing level. With a 15% adherence drop for every extra patient assigned, the colonization rate increases with decreasing staffing levels, resulting in a trend reversal through the 5% and 10% adherence drops. We perform comparisons for all pairs of design points (Table 7-5 and Table 7-6) for reducing the staffing level from 1-to-1 to 1-to-2, and 1-to-2 to 1-to-3, respectively), where a pair is defined to have identical settings in nurse and physician adherence rates and differing only in the staffing levels. For example, "DP211 vs DP111" is the setting where the nurse and physician adherence rates are set to 20%, and we wish to know the effect on patient discharge colonization rate caused by decreasing staffing levels from 1-to-1 to 1-to-2. The line segments in the plots show the 95% confidence intervals of the difference in patient discharge colonization rates. If it covers zero, it means the result is inconclusive (clinically and statistically insignificant). If the interval falls on the right (or left) of zero, it means the patient discharge colonization rate increases (or decreases) with an increase in staffing levels. The further the interval is from zero, the more clinically significant is the impact on discharge colonization rates. It

can be seen that that at 0% correlation between workload and adherence drop (black lines), the changes fall below zero, implying a decrease in patient discharge colonization rate due to the decrease in staffing level. That is, if increased workload is assumed not to cause a corresponding drop in HH adherence, then the results predict that discharge colonization rate will decrease.

Table 7-5 Analysis of effect on discharge colonization rate due to increase staffing level from 1-to-1 to 1-to-2

Change in Staffing Level:	From 1-to-1 to 1-to-2			
Percentage HH adherence drop per extra patient assigned (%):	0	5	10	15
Percentage change in patient discharge colonization rate (%): mean (std)	-3.17 (1.32)	-1.23 (1.45)	1.24 (3.09)	4.40 (5.68)
95% confidence intervals of change in patient discharge colonization rate	<p>Legend:</p> <ul style="list-style-type: none"> 0% adherence drop per extra patient assigned 5% adherence drop per extra patient assigned 10% adherence drop per extra patient assigned 15% adherence drop per extra patient assigned 			

The confidence intervals start to cross the zero threshold when the correlation between adherence drop and work load increases from 5% to 15%. In most settings (except for a few design points with the correlation set to 10% and 15%), the changes are not clinically significant since they are not greater than 5% and can be attributed to the stochastic effects. However, the results do suggest that if the adherence drop for every extra patient is 15% or greater, the colonization rate will increase with

an increased workload. If however, the adherence drop is not as great as 15%, the staffing constraints may not be a severe problem as it does not worsen the situation.

Table 7-6 Analysis of effect on discharge colonization rate due to increase staffing level from 1-to-2 to 1-to-3

Change in Staffing Level:	From 1-to-2 to 1-to-3			
Percentage HH adherence drop per extra patient assigned (%):	0	5	10	15
[†] Percentage change in patient discharge colonization rate (%): mean (std)	-1.60 (0.69)	1.03 (2.52)	6.04 (7.70)	9.19 (7.05)
95% confidence intervals of change in patient discharge colonization rate	<p>Legend:</p> <ul style="list-style-type: none"> 0% adherence drop per extra patient assigned 5% adherence drop per extra patient assigned 10% adherence drop per extra patient assigned 15% adherence drop per extra patient assigned 			

We further investigate of reason of the inconsistent trends of change in the staffing levels by tracing the incidences of patient colonization via 4 routes:

1. *Patient is admitted with MRSA.* Patient is already colonized with MRSA from unknown sources (including community or from previous hospital stays) at the time of admission to the SICU.
2. Patient is colonized with MRSA from direct contact with a Visitor. We denote this route as Visitor → Patient, where the symbol → shows the direction of transmission.
3. *Patient is colonized with MRSA from another patient via direct contact with a Staff* (denoted as

Patient $\rightarrow$ *Staff* $\rightarrow$ *Patient*). It is expected that the transmissions via this route will reduce with dedicated assignment of nurses to patients (e.g., 1-to-1 assignment).

4. *Patient is colonized with MRSA from a Staff who was colonized outside the ward* (denoted as *Outside ward* $\rightarrow$ *Staff* $\rightarrow$ *Patient*). By “outside the ward”, we mean that the staff is already colonized at the time he or she enters the SICU (e.g., from home, canteen, operating theatre, etc), and the external source of colonization is unknown.

Analysis of the individual transmission events and routes (details are not shown here) reveals that incidences of colonization at admission, and caused by the route *Visitor* $\rightarrow$ *Patient* do not drastically change with the change of staffing levels, and they together contribute about 80-90 patient incidences consistently across all design point. The other 2 routes, *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* and *Outside ward* $\rightarrow$ *Staff* $\rightarrow$ *Patient* produce contradicting trends in patient discharge colonization rates with respect to change in staffing level, and the overall trend follows the dominating routes:

When people are more adversely affected by increased workload (i.e., a high correlation between the adherence drop and low staffing level, e.g., 15%), the number of transmissions via the route *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* will dominate, resulting in the net total number of incidences to increase.

When people are less or not adversely affected by increased workload (i.e., a low correlation between the adherence drop and low staffing level, e.g, 0% - 10%). The route *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* is dominant only when nurse adherence rates are low (20%). With higher nurse adherence rates (40% and above), incidences via the route *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* reduces and the route *Outside ward* $\rightarrow$ *staff* $\rightarrow$ *patient* becomes more dominant. The net effect is a decreasing trend: patient discharge colonization rates reduces with decreased staffing level.

We apply MANOVA analysis with 48 design points (at $\alpha = 0.05$) to uncover confounding effects and to rank factors according to their ability to explain the variance in the outcome (details are not shown here). The results confirm that nurse adherence rate is the most important factor that influences the patient discharge colonization rate. Regardless of drops in HH adherence rates, it explains at least 80% of the variance and is therefore ranked as the first for all factors and factor interactions. Physician HH adherence and staffing level are far less impactful compared to nurse HH adherence rates. The results also confirm that the effect of staffing level is controversy, and it is closely

dependent on the setting of HH adherence drop with the extra workload.

7.4. Summary

We have conducted a computer experiment using ABMs for an SICU ward. The purpose of the experiment is to investigate the effect of hand hygiene adherence and staffing constraints on the patient discharge colonization rate. The questions that drive the experiment, which are realistic operational questions that hospital stakeholders would like to know, are that considering the imperfect implementation and compliance to the intervention measures in real life situations, what would happen if the staffing level is extremely low (e.g., half of the nurses are on leave)? And what if the actual HH adherence is much lower than the audit results (which is usually the case as HCWs are more compliant to HH measures when they are observed by auditing staff)?

The construction of a null model that is suitable for this computer experiment took a lot of preparation work, including getting approval and support from the hospital stakeholders, setting up the movement tracking devices and persuading HCWs in the ward to cooperate with us, analyzing the logged movement data to compute the timing parameter values, making field observations 24 hours a day (in 3 shifts with the team members from ROSS Scientific and student helpers), and interviewing a wide range of types of HCWs to ensure their daily activities are understood. When the null model was built, it took several meetings with the hospital stakeholders to adjust and validate the animated simulations and the preliminary quantitative results.

The reasonably realistic null model enables us to conduct a full factorial computer experiment which, to the best of our knowledge, is an unprecedented piece of work. The experiments confirm the general belief that the adherence rates of HCWs have clinically significant effects on the patient discharge colonization rates, where nurses' adherence rates are shown to be more impactful (explains at least 80% of the variance) than the physicians' adherence rates. The experiments also reveal the insights of the seemingly confusing results, that with the decrease of staffing level, the patient discharge colonization rates decrease with some settings but increase with other settings. By tracing the individual transmission events and routes in the simulations, we confirm that the incidences caused by route *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* do increase when the staffing level is low

(where one staff nurse has to take care of 2 or more patients while in the ideal case, each should be only assigned to one patient). It is the dominance of either the route *Patient* $\rightarrow$ *Staff* $\rightarrow$ *Patient* or *Outside Ward* $\rightarrow$ *Staff* $\rightarrow$ *Patient* that determines the overall effect of the staffing level. The practical conclusion from this experiment is that with a low correlation between the adherence drop and low staffing level (for example; 0% to 10%), the constraint of staffing level is not severe in terms of the risk of causing more nosocomially infected patients. These are examples of predictions that are valuable to hospital stake holders' what-if concerns, and they potentially can be used as references for practical decision makings.

Chapter 8

Computer Experiment II: Chemotaxis

In this Chapter, we present the application of computer experiments to sophisticated thought experiments. Thought experiments are mental exercises which would consider a hypothesis, theory or principle [216]; it may or may not be possible to actually perform thought experiments [57]. Famous examples of thought experiments include Schrödinger's cat [217] and Galileo's demonstration that falling objects [218]. In these two examples, the experiments demonstrated paradoxes by applying the theories; thus they pointed out the shortcomings or invalidities of the theories.

Computer experiments can be seen as thought experiments since they are not actually performed on real world objects, yet they can make verifiable or falsifiable predictions by modeling after a hypothesis or theory. Computer experiments are able to handle difficult scenarios where predictions need to be made through computations or simulations, which are too complicated for mental inferences.

The computer experiments presented in this Chapter is motivated by the research history of chemotaxis [30], which we have introduced in Section 5.3. Before the mechanisms that transduce the extracellular chemical signals into cell locomotion [34, 68, 69, 193-195] were made clear, the then models such as Keller and Segel's Model [26] made direct assumptions on the quantitative relationship between chemoattractant concentrations and cells' movement, without referencing the underlying molecular details that caused it. With the advances of knowledge in the cellular mechanisms, one would question whether adding these molecular details would be beneficial, since according to Occam's Razor [41], simpler models are preferred unless increased explanatory power is obtained with the more complicated models.

We can identify two related problems for this application: 1) the process of discovering a plausible model of the given phenomenon; and 2) the balance of the model simplicity and its explanatory power. Our computer experiment on chemotaxis is to demonstrate a discovery process through thought

experiments with the justifications of added model details. The procedure is outlined below. Assume that we had no prior knowledge of molecular and cell locomotion. When encountered an unknown phenomenon of chemotaxis, we progress by making intuitive assumptions, or informal hypotheses, on the quantitative relationships between the observed variables. We use ABMs to model after these assumptions, run simulations, and get immediate feedback on whether the assumptions lead to results that can correlate with the reality. Then we increase the model granularity by incorporating molecular-level interactions. If the result is a better correlation with the reality, it will justify the increase in model complexity.

8.1. Intuitively assumed chemotaxis models

We develop four chemotaxis models in this Section. The first three models mirror an early stage of the discovery process when molecular mechanisms, such as the interactions between cell receptors and ligands, were still unknown. Each of them directly assumes a seemingly logical quantitative relationship between the chemotactic cell movement and the gradient of the chemoattractants. These assumptions represent the best guesses a researcher might make without further knowledge about the molecular details. Through the computer-based thought experiments, one would get immediate feedback on the likely correctness of these assumptions. The fourth model mirrors an advanced stage in the discovery process where more and more molecular mechanisms are discovered; these details can be added to the models to achieve greater realism but at the price of added model complexity. If the experiment results show better agreement with the reality, then it can be concluded that the added model complexity is justifiable.

- *Model I: Linear Dependency on the Gradient*

This model is based on the intuition that the steeper the gradient, the faster is the chemotactic movement. The simplest assumption is to define the chemotactic speed (displacement per unit time) to be proportional to the attractant gradient. Given the local gradient $\mathbf{G}$, cell radius r , the concentration difference across the cell along the direction of the gradient is $2r|\mathbf{G}|$. The chemotactic displacement $\Delta \mathbf{p}_{chemo}$ assuming linear dependency on the chemoattractant gradient is therefore

$$\frac{\Delta \mathbf{p}_{chemo}}{\Delta t} = k_1 \cdot 2r|\mathbf{G}| \cdot \frac{\mathbf{G}}{|\mathbf{G}|} = k_1 \cdot 2r\mathbf{G} \quad \text{Eq. 8-1}$$

where k_I is the proportionality parameter that determines the sensitivity of the cell to the gradient.

- *Model II: Saturated Dependency on the Gradient*

It is intuitive to imagine an upper limit to the cell's velocity, since the cells cannot move infinitely fast. Let $l_{\max}$ represent the largest possible chemotactic movement per unit time. The chemotactic displacement $\Delta \mathbf{p}_{chemo}$ assuming saturated dependency on the chemoattractant gradient is therefore

$$\frac{\Delta \mathbf{p}_{chemo}}{\Delta t} = l_{\max} \frac{2r|\mathbf{G}|}{k_{II} + 2r|\mathbf{G}|} \cdot \frac{\mathbf{G}}{|\mathbf{G}|} = l_{\max} \frac{2r\mathbf{G}}{k_{II} + 2r|\mathbf{G}|} \quad \text{Eq. 8-2}$$

where k_{II} is a tunable parameter. k_{II} can be interpreted as the “half-maximal concentration difference”; that is, when the cross-cell concentration difference $2r|\mathbf{G}|$ is k_{II} , the chemotactic displacement is half of the maximum chemotactic displacement. In this case, the smaller the value of k_{II} , the more sensitive the cell is to the gradient, subject to a ceiling of $l_{\max}$.

- *Model III: Linear Dependency on the Relative Gradient*

The above models consider only the attractant gradient but not the attractant concentration. According to Weber-Fechner's law, the larger the stimuli, the larger a difference in stimuli would be required in order to entail the same amount of response. Intuitively, this motivates a model where the chemotactic difference is proportional to the *relative gradient* $\mathbf{G}/CV$, where C is the local concentration of the attractant, and V is the volume of the cell. Therefore

$$\frac{\Delta \mathbf{p}_{chemo}}{\Delta t} = k_{III} \frac{2r\mathbf{G}}{CV} \quad \text{Eq. 8-3}$$

where k_{III} is the proportional parameter. Similar to Model I, the larger the value of k_{III} , the more responsive is the cell to the gradient, but the responsiveness is now also modulated by the chemoattractant concentration. The loss of responsiveness with increased concentration may be interpreted as the saturation of cell receptors by ligand molecules. However, Eq. 8-3 is purely phenomenological and there is no underlying model of receptor kinetics as suggested.

- *Model IV: Nonlinear Dependency based on Receptor Kinetics*

This model is the same as the version we presented in Section 5.3, that it takes into account the

receptor kinetics (including: receptor-ligand binding, receptor internalization, and receptor recycling).

The chemotactic movement is computed by the equation below (derivations shown in Section 5.3):

$$\frac{\Delta \mathbf{p}_{chemo}}{\Delta t} = k_{IV} \frac{\Delta[LR]_{diff}}{\Delta t} \frac{\mathbf{G}}{|\mathbf{G}|} \quad \text{Eq. 8-4}$$

where k_{IV} is the proportional parameter, and $\Delta[LR]_{diff}$ is the difference of newly bounded receptors at the front and rear of the cell. The larger the value of k_{IV} , the more responsive is the cell movement is to the occupancy difference in newly bounded receptors at its front and its rear.

8.2. Experimental results and discussions

We conduct computer experiments with the same setup as described in Section 5.3. The experiments mimic an Under-Agarose Assay.

- *Model I vs II: The Impact of Speed Saturation*

For both Model I (Linear Dependency on the Gradient) and Model II (Saturated Dependency on the Gradient), the cell movement speed is dependent on the molecular gradients. The simulations run for 10,000 time units, which is about when the attractant becomes evenly distributed, and the movement of the cell population becomes negligible due to insufficient molecular gradients.

For Model I, it is expected that the larger the value of parameter k_I , the more responsive (reflected as faster speed) are the cells to the attractant; but for Model II, we expect the converse: higher responsiveness for smaller the values of k_{II} . The simulation results in Figure 8-1 confirm our expectations. These temporal patterns from the two models are placed side by side for comparison of their qualitative similarities. The measurements of x_{avg} (mean x-position of cell population), x_{min} (left-most cell position), x_{max} (right-most cell position) and L_x/L_y (pattern elongation) show that the asymptotic results (t=10000 here) for Model I and Model II are similar (measurements are not shown here).

The distinction between Models I and II is evident only in the short term for differences in responsiveness of cells to the attractant ($k_I = 100$ and $k_{II} = 0.01$). The measurements for the entire time course of the simulation are plotted in Figure 8-2. In Model I, the cell movement is much more

abrupt than that in Model II, causing a more dispersed cell distribution during the early stages: in Model I, the front of the cell population (x_{max}) shapely changes from 1250 (original position) to 2000 within the first 1000 time units; while in Model II, the same amount of change only occurs by 2000 time units. Figure 8-3 shows the snapshots of the simulation at the specific time instances ($t=1000, 2000, 3000$). The more dispersed cell distribution can be observed in both the top-views and the cell histograms (Figure 8-3, Model I, plot (a)).

Figure 8-1 Top-view of the cell patterns at $t=10000$: Model I on the left, (a)-(d): $k_1 = 3, 10, 30, 100$; Model II on the right (a)-(d): $k_{II} = 0.3, 0.1, 0.03, 0.0$

This difference between Model I and II can be explained by the difference in their agent movement rules: Model II imposes a maximum cell speed which is absent in Model I. In the latter, a large k_1 causes the cells to move so rapidly that the maximum speed exceeds the speed saturation level of Model II. The dispersed distribution of cells coincides spatially with the peak of the attractant gradient. Recall that for Model I, the cell speed is directly proportional to the attractant gradient; therefore, the dispersed cell population is caused by rapidly-moving cells in response to the high attractant gradient. However, as observed in Figure 8-1, in the longer term, the circular cell

population pattern is restored (which are confirmed by the decrease in the pattern elongation in Figure 8-2). The cells reach highest concentration point (zero-gradient point) and hardly move afterwards. If they do move towards the right by random movement, they will be brought back by strong chemotactic attraction).

Figure 8-2 Migration parameters for Model I (left) and Model II (right)

Figure 8-3 Snapshots of the cell migration pattern, cell histogram, attractant concentration and gradient at t=1000, 2000, 3000 for Model I and II

When compared to the laboratory experiment observations (Figure 5-5), the results of Model I and II are most realistic with low responsiveness ($k_I = 3$ and $k_{II} = 0.3$), where the cells both spread and drift,

exhibiting the behavior of a biased random walk. That is, the intuitive assumptions of direct or saturated dependency between cell speed and attractant gradient produce sensible results with these parameter settings. The results obtained from higher responsiveness (larger k_I and smaller k_{II}) do not correlate with real observations in that the cell population *shifts* but does not *spread*; and for very high responsiveness, it even *concentrates* at a point. These hypothetical higher-than-necessary responsiveness are falsified as their predictions do not agree with the reality.

- *Model III: The Impact of Rapid-Changing Relative Gradient*

In Model III (Linear Dependency on Relative Gradient), cell movements are set to be directly proportional to the relative gradient (defined as the ratio of the local gradient and the concentration). The simulations run for about 5000 time units. Figure 8-4 shows the top-view of the cell migration pattern at $t=5000$ for k_{III} set to 30, 100, 300, 1000 respectively. As expected, larger k_{III} results in a greater horizontal displacement towards the chemoattractant. A distinctive feature of Model III is that the final cell population pattern is roundish. The observations from the simulation with a higher rate of sampling (detailed plots are not shown here) indicate that the whole population appears to *shift as a whole*, and the circular migration pattern is not distorted much throughout the simulation. Given that the cell speed is defined directly proportional to the *relative* gradient, it implies that the all cells experience a similar relative gradient and hence move at similar speeds.

Figure 8-4 Simulation results of Model III ($t=5000$)

To view the transient behavior of the cell migration pattern, we plot the measurable parameters over the entire simulation for $k_{III} = 300$ and $k_{III} = 1000$, shown in Figure 8-5. Measurements are made every 10 time units. Two distinct stages can be observed. A brief transient period (from $t=0$ to

$t=500$), where the pattern elongation L_x/L_y rapidly increases then decreases, implying that the population front accelerates first, then the population rear quickly catches up. The change of the population front and rear can also be observed from the same figure. After $t=500$, the cell population shifts as a whole, where the front and the rear of the population progress at about the same speed. The pattern elongation L_x/L_y gradually reduces to less than 1.

Figure 8-5 Migration parameters for Model III. $k_{III}=300$ (left), $k_{III}=1000$ (right)

Figure 8-6 Snapshots of the cell migration pattern, cell histogram, attractant concentration and gradient at $t=40, 60, 80$ for Model III with $k_{III} = 1000$

To investigate the cell movement pattern during the transient period, we plot the snapshots for $k_{III} = 1000$ at $t=40, 60, 80$ (Figure 8-6). The peaks of the relative gradient, which correspond to regions of fastest-moving cells, are indicated by the arrows in the bottom panel of each plot. The peak of the relative gradient rapidly sweeps to the left, which is expected since the relative gradient is the ratio between gradient and concentration. It is maximal at the edge of the molecular spread, since the concentration there (hence the denominator of the ratio) is very small. As the peak of the relative gradient sweeps through the cell population, it causes the cells to accelerate line by line, producing a

dispersed distribution of cells in the middle of the population. The relative gradient values following the peak is much smaller than the peak, hence the cells that have just experienced the peak soon decelerate, creating a dense population front. By $t=80$ the whole cell population is within a relative gradient distribution which is monotonically decreasing towards the right. Such a distribution helps the population rear to catch up with the front, reduce the elongation, and to finally restore the circular pattern of the cell population.

When compared to laboratory observations (Figure 5-5), the results of Model III shown in Figure 8-4 are not realistic in the sense that they do not spread but rather concentrate. The restoration of the circular cell population pattern does not agree with the experiment result either. The agent movement rule of Model III is based on the rationale that cells within a higher attractant concentration band require a higher attractant gradient to achieve the same level of response as compared to a lower concentration band. However, assuming a direct linear dependency of cell speed on relative gradient appears to be too simplistic and unrealistic in its quantitative predictions.

- *Model IV: The Impact of Cell-Molecule Interaction*

Model IV uses the same settings presented in Section 5.3. Its results have been shown from Figure 5-7 to Figure 5-10, Table 5-2). Model IV is more realistic in model granularity in that it takes into consideration the molecular details such as receptor kinetics and receptor-ligand interactions. The results resemble the laboratory observation (Figure 5-5) more closely than those from Model I, II and III. We have made various attempts to tune parameter values for Model I, II, and III, but they always produced a similar, relatively sharp and concentrated front. These 3 models are unable produce a widely spread front as that of Model IV.

To measure the extent of the frontal spread, we select 10% of the frontal cells from the population (100 cells), and compute their deviations in both x and y directions. The larger the values, the more widely spread the cell pattern. As shown in Table 8-1, the spread of the population front is significantly larger for Model IV than the other three models, which shows that Model IV is a better approximation of the reality (refer to laboratory observations in Figure 5-5), with its added model granularity.

Table 8-1 Extent of the cell spread: measured in deviation in x and y directions of 10% of the frontal cells

Model	Deviation in x direction	Deviation in y direction
Model I, $k_I = 10$, $t = 5000$	49.04	79.35
Model II, $k_{II} = 0.1$, $t = 5000$	47.32	80.69
Model III, $k_{III} = 30$, $t = 5000$	33.56	91.68
Model IV, $k_{IV} = 25$, $t = 5000$	210.62	287.88

8.3. Summary

We have explored the use of computer experiments as thought experiments, in an application that models and validates the intuitive quantitative assumptions for the biological phenomena, chemotaxis. Model I (Linear Dependency on Gradient) and Model II (Saturated Dependency on Gradient) assume a direct monotonically increasing dependence of cell speed on the attractant gradient. Model III (Linear Dependency on Relative Gradient) is based on the rationale that cells within higher attractant concentrations require higher stimuli (in the form of attractant gradient), hence we assume a linear dependence of cell speed on the attractant *relative* gradient. These assumptions are all logically intuitive and without simulation or experimentation, it is difficult to discover whether or why they are flawed. After the simulation and comparison with the laboratory observation, it is evident that Model III is far from reality, and Model I and II resemble the reality only with certain parameters settings. By further investigating the spatio-temporal relationship between cell patterns and attractants' concentration, gradient, and relative gradient, we are able to trace how the quantitative formulation leads to the unrealistic results. Such analysis further reveals why Model IV (Nonlinear Dependency based on Receptor Kinetics) succeeds in reproducing the pattern of *biased random walk*: not only the cells are inflected by the attractants' distribution, but attractants' distribution is also modified by the cells through receptor-ligand binding. The change in attractant distribution in turn influences the cell movement, creating a wide-spread cell pattern. These are the kind of insights not obtainable via real *in vivo* or *in vitro* laboratory experiments.

In scientific research, it is common to have competing hypotheses, where each seems to be a sensible explanation as they provide logically intuitive reasoning. However, the crux of the problem is

whether the inferred mechanisms or assumptions lead to quantitative agreement to reality. The costly way is to experiment the assumptions with real system, which could be time-consuming and difficult to control in the face of numerous interfering factors. Computer experiments provide an alternative way to test the competing hypotheses. We might hope that models make predictions that agree with reality hence confirming the hypotheses; however, it is equally valuable if the model proves that the hypotheses are incorrect by showing predictions that deviate from reality. By analyzing the model structure and simulation data, we gain insights on what could go wrong, which leads to the proposals of new hypothesis or assumptions, possibly with more detailed model granularity, of what could make it right. This could lead to revision of models and more computer experiments, or guide new *in vivo* or *in vitro* experiment designs for new data collection.

Chapter 9

Conclusions

9.1. Summary

In this thesis, we focused on the theories and methodologies of conducting computer experiments due to their great potential in providing a less costly, safer, ethical and potentially more powerful alternative to the real world experiments. We believe that for a rigorous computer experiment, it needs to mirror a real world experimental study design, namely, to have a null model as the control group, and one or more experimental models as the treatment groups. With this methodology, we identify the primary challenge to be constructing a credible and extensible null model that has sufficient model scope and granularity to enable experimental manipulations. Given that Occam's razor is usually adopted as the philosophy for deciding on model granularity, our work has focused on the computational structures and scientific processes that are needed to realize it. It may require extensive efforts of modelers to conduct field studies in order to understand the domain, to confirm the modeling details and parameter values, which may give the impressions that the models for computer experiments are "expensive" to build. However, given the questions asked for the experiment, and comparing to the cost if the experiment would be conducted in real-life, the cost of a computer experiment is still much lower and ethical.

In this thesis, we focus on applying computer experiments to the domain of epidemiological applications. Through the literature review and a feasibility study of HIV pathogenesis, we identify Agent-Based Models (ABMs) as a more suitable modeling technique for supporting computer experiments, compared to the traditional Differential Equation (DE) based models. ABMs typically represent entities as stateful, autonomous individuals (or agents), which make modeling of heterogeneous entities straightforward. This suits the problem domain of epidemiological applications, where entities such as biological cells or human hosts usually have complex states and interact non-uniformly in a topologically complex environment. It also suits the need of conducting

experiments since experimental manipulations are usually specified at a very detailed level. Our feasibility study of HIV pathogenesis experiment has also revealed several practical modeling issues that may sacrifice the model credibility and extensibility, hence impeding the model's usefulness for computer experiments.

In this thesis, we propose a multi-scale agent-based modeling and simulation framework that formally addresses these modeling issues so as to support the construction of credible and extensible ABMS. The framework consists of following building blocks:

1. An *event-scheduling update scheme* that addresses the requirement of modeling multi-timescale activities that span a range of update frequencies. The scheme uses variable time advances to invoke behavioral rules only when required, thus both supporting realistic time semantics and improving the computational efficiency. As cancellations of the scheduled events frequently arise from the complex entity interactions and life cycles, we build the implicit and explicit cancellation operations into the scheme. Through the case studies of modeling B cell life cycles, patient admission and stays, and daily work routine of a staff nurse, we show that the proposed event-scheduling update scheme realistically models the multi-timescale events, and outperforms the equivalent models implemented with the time-step-based update scheme.
2. A *hybrid entity representation* that addresses the requirement of proper entity relative scales, that is, the model needs to realistically reflect the relative quantities and the physical sizes of entities of different types. The hybrid entity representation models the smaller, simpler, numerous entities at the population level, and the larger, complex, less numerous entities at individual level as agents. This approach balances the model realism and computational tractability. The hybridization is realized at two levels: globally in the environment, and locally with an agent. Hybrid entity representation needs to be supported by the Agent Holder Layer and Population Quantity Layer(s) of a layered environment model, which is described below.
3. A *layered environment model* that addresses the requirement of building topologically accurate environment. The environment model is stratified into four functional layers: an Agent Holder Layer that position the agents, a set of Population Quantity Layer(s) that position the

population-level entities, a Connection Layer that specifies the way multiple compartments are connected, and a set of Flow Field Layer(s) that specify the background flows that systematically transport entities. The hybrid entity representation and the layered environment model are closely related hence we demonstrate their efficacy together via two cases studies. In a chemotaxis model, two entity types (i.e., molecules and cells) whose quantities differ by the order of 10^3 are efficiently modeled with the hybrid entity representation. The capability of the layered environment model is fully demonstrated in a case study of ventilation systems. Two compartments, the open space and the ventilation ducts, are modeled. The pathogen entities get transported by the Flow Field Layer and cross the compartments through the Connection Layer. With these building blocks, we are able to model the inter-compartment behaviors such as pathogen absorption from the open space and filtration in the ventilation ducts. The system outage scenarios are also modeled, for example, the ventilation ducts leak the absorbed pathogens back to the open space.

4. *An iterative strategy for model extension based on an agent behavior network model (ABN)* that addresses the requirement of appropriate entity granularity and model extensibility. The ABN formally describes an agent as a network of states variables and behavioral rules. Based on the ABN formalization, the incremental model changes can be described as a combination of additions and replacements of nodes and links in the ABN. We propose an iterative strategy for model extension that conducts checks and applies incremental changes for each iteration, until the refined model is sufficiently detailed and accurate to serve as the null model. We demonstrate the proposed strategy through three chemotaxis models with gradually added granularities.

We apply the proposed multi-scale agent-based modeling and simulation framework to full scale computer experiments. In this thesis, we focus on two epidemiological applications. The first experiment assesses the effect of healthcare workers' (HCWs) adherence rates and staffing levels on the nosocomial transmission of MRSA in an SICU ward. The aim of the experiment is to examine the operational questions that concern the hospital decision makers. We construct a null model that represents the existing state of the SICU. This model is face-validated with hospital stakeholders and its predicted disease transmission dynamics is quantitatively validated to match the clinical

observations. Then we apply the experimental manipulations to the null model by setting hypothetical values for hand hygiene adherence rates and staffing levels. The experiment shows that the nurse adherence rate is the most impactful factor on the patients discharge colonization rates. Comparing to the physician adherence rates and the staffing levels, it explains 80% of the variance in the model output. It also shows that with a low correlation between the adherence drop and low staffing level (for example; 0% to 10%), the constraint of staffing level is not severe in terms of the risk of causing more nosocomial infections to patients. These results may be used by hospital stakeholders as references for policy making or devising further studies.

Our second experiment mirrors a thought experiment. It comprises of four chemotaxis models that correspond to four intuitive assumptions of the quantitative relationship between chemoattractant concentrations, gradients, and the locomotion of cells. This study mimics a step in the scientific research process, when one is faced with a few competing hypotheses and each seems to be a sensible explanation since they provide logically intuitive reasoning. By capturing the intuitive assumptions into models, those that produce nonsensical results through simulation have to be rejected. The experiment results show that the model with the details of receptor kinetics produces results that correlate with the reality most closely. In addition, analysis on the simulation data reveals that it is the change in the attractant distribution that is missing in the other three models which cause their simulated cell migration patterns to be unrealistic. These are unique insights from computer experiments that are not usually obtainable via real *in vivo* or *in vitro* laboratory experiments.

9.2. Suggestions of Future Work

We have proposed the ABN representation for agent description that enables a modeler to visually inspect the model structure for incompleteness such as runaway processes. Given the complexity an ABM can possibly have, manual inspections or checking may not be always viable. We wish to investigate further in this direction to develop formal model checking techniques for computational simulation models [219]. Ideally, with a given specification that contains the requirements of the model, the model checking algorithm should automatically verify the correctness of the model design in that undesired properties such as runaway processes are absent.

Some case studies and experiments presented in this thesis can be extended to more interesting and possibly more influential applications. In Section 5.4 we presented a hypothetical hospital environment model with a ventilation system. This case study can be extended to a realistic hospital model to simulate the nosocomial airborne transmissions. When such a model is in place, it can be used as an experimental testing bed to assess the hospital's readiness in the outbreak events such as H1N1 transmissions. The ventilation system model can also be improved for operating theatre models where the quality and the flow of the air are vitally important [202]. In the nosocomial transmission experiment presented in Chapter 8, we did not consider the scenario of re-admitted patients. This fact may interest researchers [74, 220] in the future nosocomial transmission experiments. These studies would require involvement of the hospital management, the domain experts, and the staff who can advise on the domain knowledge and validate the simulation results.

The HIV pathogenesis hypothesis verification study presented in the preliminary studies (Section 2.3) is another application we wish to study further. With the proposed modeling and simulation framework, we are now in a better position to solve the issues discussed in Section 2.3.3 and develop a more realistic model for HIV pathogenesis and human immune systems. Similar to any other real life applications, such a study would need domain experts' input in giving guidance on the relevant literatures, verifying and validating the model, advising the sensible parameter values and model scales, and suggesting the practical problems or questions that they expect the model to answer.

The need of model scalability through the full utilization of the available computational resources is another important practical topic we wish to address. For the SICU study presented in Chapter 7, we distributed the simulation replications to 4 servers (2 Windows and 2 Linux servers) manually for the start, and then moved onto simple coordinator-simulators architecture to automate the process. It can be expected that the simulation efficiency can be greatly improved with technologies such as grid computing or cloud computing infrastructures [221, 222].

Besides the epidemiological applications, our modeling framework and experiment methodology can potentially be applied to other areas such as ecology, economics, and social sciences. ABMs have been actively investigated and applied in these areas [160, 223-227]. These problem domains share many common characteristics with the epidemiological domains that motivate the use of ABMs. For

example, the system is made up of a large number of heterogeneous entities with complex states and behaviors; as a result, the system level dynamics emerges from the massive interactions of the individuals. The real world experiments in these areas are facing similar challenges as the epidemiological experiments. For example, numerous confounding factors exist, which are difficult to eliminate or control; it can be costly or even impossible to get a large sample size for the experiment in order to meet the statistical significance. If carefully planned and designed, computer experiments may be applied in these areas as less costly alternatives that may produce useful insights or suggest further studies.

Appendix A

Author's Publications

Journals

- Joc Cing Tay, Zaiyi Guo, Wei Boon Chan, Hardiyanto Wibisono, Paul A. Tambyah, “The Effect of Staffing Levels on Hand Hygiene Adherence: New Insights from an Evidence-based Behavioral Model of MRSA Transmission within an SICU,” in Preparation.
- Zaiyi Guo, P. M. A. Sloot, Joc Cing Tay, “A Hybrid Agent-Based Approach for Modeling Microbiological Systems,” *Journal of Theoretical Biology*, Volume 255, Issue 2, 163-175, 2008.
- Zaiyi Guo and Joc Cing Tay, “Multi-Timescale Event Scheduling in Multi-Agent Immune Simulation Models”. *Journal of Biosystems*, Volume 91, Issue 1, 126-145, 2007.

Conferences

- Joc Cing Tay, Zaiyi Guo, Wei Boon Chan, Hardiyanto Wibisono, Paul A. Tambyah, "Assessing the Efficacy of Hand Hygiene & Contact Precaution Adherence Rates on Nosocomial MRSA Transmission given Staffing and Behavioural Constraints”, Abstract published at the 1st International Congress of Infection Control Association (Singapore), 2009.
- Zaiyi Guo and Joc Cing Tay, "Granularity and the Validation of Agent-based Models," In *Proceedings of Agent-Directed Simulation Symposium (ADS'08)*, pp. 153-161, 2008.
- Zaiyi Guo and Joc Cing Tay, “Assessing Causal Intuitions in Agent-based Microbiological Models,” In *Proceedings of the European Conference on Complex Systems (ECCS 2007)*. Dresden, Germany, Oct 1 – 4th, 2007.
- Zaiyi Guo and Joc Cing Tay, “A Hybrid Agent-based Model of Chemotaxis,” In *Proceedings of the 7th International Conference on Computational Science (ICCS 2007)*, LNCS 4487, pp. 119–127, 2007
- Z. Guo and J. C. Tay, “A Comparative Study of Modeling Strategies of Immune System Dynamics under HIV-1 Infection,” In *Proceedings of the International Conference for Artificial Immune Systems (ICARIS 2005)*, LNCS 3627, pp220-233, 14 – 17th August, Banff, Alberta, Canada.
- Z. Guo, H. K. Han, J. C. Tay, “Sufficiency Verification of HIV-1 Pathogenesis based on

Multi-Agent Simulation,” In Proceedings of the ACM Genetic and Evolutionary Conference 2005 (GECCO 2005), vol I, pp305-312. Best Paper Nominee.

- Z. Guo., H. K. Han., J. C. Tay, “Modeling Immune Dynamics of HIV-1 Infection Using Multi-Agent Simulation,” Abstract published at the Graduate Student Symposium for the 3rd Asia Pacific Bioinformatics Conference (APBC 2005).

Technical Report

- Z. Guo, J. C. Tay, “On Agent Interactions, States and Rules: A Technical Specification of CAFISS II”. 2005. ER Lab, School of Computer Engineering, Nanyang Technological University, Singapore.

Appendix B

B Cell Life Cycle Model

B.1.Event definitions and executions

Figure B-1 depicts the temporal relationship of the events involved a B cell life cycle. The events (white and black dots) are labeled in the order that they are scheduled. Their relative positions on the timeline indicate their chronological order. The coexistence of multiple timelines indicates the concurrency of the multiple processes on the same agent. Table B-1 lists the event definitions, which follow the event structure defined in Section 4.2.1, Figure 4-3.

Figure B-1 B cell life cycle event scheduling (replicated from Figure 4-5)

Label	Event Name	Operations	Next Events
	Conditions		
1	create_from_bone_marrow	state = naive	t ₁ : naive_death
	none		
2	naive_death	state = dead	
	state = naive	remove from environment cancel all events	

3, 6	bind_virion (virionX)	receptorBinding.add (virionX)	t ₂ : internalize (virionX)
	none		
4, 7	internalize (virionX)	internalized.add (virionX)	t ₃ : present_antigen (virionX)
	receptorBinding.contains (virionX)		
5, 8	present_antigen(virionX)	presenting.add (virionX.peptide)	
	internalized.contains (virionX)		
9	activation	state = activated	t ₄ : divide l = p ₁ +p ₂ t ₅ : p ₁ : convert_plasma t ₅ : p ₂ : convert_memory
	none		
10, 13-15	divide	Create one newB add to environment	t ₄ : divide newB 0: create_divide
	state = activated		
11a	convert_plasma	state = plasma	t ₆ : produce_antibody t ₇ : plasma_death
	state = activated		
11b	convett_memory	state = memory	
	state = activated		
12	create_divide (parent_cell)	state = activated t _c = parent's time to convert	t ₄ : divide l = p ₁ +p ₂ t _c : p ₁ : convert_plasma t _c : p ₂ : convert_memory
	none		
16, 18-24	prduce_antibody	create n _{AB} antibodies add them to environment	t ₆ : produce_antibody
	state = plasma		
17	plasma_death	state = dead remove from environment cancel all events	
	state = plasma		

Table B-1 Event definitions of B cell life cycle

A naïve B cell (B cell instance A) produced by the bone marrow executes its first event `create_from_bone_marrow` (label 1), which sets its state to `naïve`. As a resting naïve B cell does not have much activity until its receptors bind to an antigen, it schedules only one event, `naïve_death` (label 2) with waiting times t_1 , that is, after the scheduling, the scheduler waits for t_1 units of time, and then invokes the `naïve_death` event. The duration parameter t_1 represents the lifespan of the a naïve B cell, which is in the order of weeks to years [45]. If the naïve B cell does not match (or encounter) any antigens, it naturally dies after t_1 elapses. The `naïve_death` event is conditional on the cell being in the `naïve` state. As shown in Figure B-1, this particular B cell does encounter events during which it switches its state; so eventually the `naïve_death` event is implicitly cancelled.

The external event of collision with virionA (label a) does not result in matching, so no event is scheduled. In a complete ABM simulation, collisions occur when the distance between the spatial coordinates of two agents is less than the sum of their radii. In this partially developed *testing* model, the collision signal is *generated* in a predefined manner. The collision with virionB (label b) results in receptor matching. This invokes the event `bind_virion` (label 3) which starts the receptor-mediated endocytosis, antigen processing and presentation. The simplified endocytosis process involves events `bind_virion`, `internalize`, and `present_antigen` (label 3-5), which schedules each other in cascade. While the B cell is processing the antigen of virionA, more collisions can occur concurrently as shown by the parallel timelines. Another collision with virionC (label c) occurs, starting a new endocytosis and antigen presenting process (label 6-8) that runs concurrently with the first antigen presenting process. The endocytosis and antigen presentation process is relatively short, in the order of minutes to hours [192].

When an antigen-presenting B cell collides with an effector TH cell (label d), the B cell's surface is scanned by the TH cell for MHC-II peptide complex expression. If the complex can be matched by the TH cell's receptor, the TH cell then activates the B cell. As shown in the figure, by the time of collision, antigen presentation of virionB will be completed, while antigen presentation of virionC remains in progress. The activation, therefore, is caused by matching between TH cell receptor and the peptides from virionB.

The activation by virionB invokes event `activation` (label 9) which switches the B cell state to `activated` (and effectively voids the `naive_death` event), and schedules the next cell division event (`divide`, label 10). This clonal expansion takes 2-3 days, then the cell differentiates into either a plasma or memory cell, with probabilities p_1 and p_2 , respectively. The `divide` event produces a new B cell, which starts with an `activated` state, and its lifespan being approximately the same as the remaining time of the parental B cell (event `create_divide`, label 12). This prevents the daughter cells from proliferating perpetually. A daughter B cell can produce more daughter cells (as indicated by B cell instance B to H), each makes its own probabilistic decision whether to become a memory or a plasma cell. The event `divide` is self-replicating, that is, each `divide` schedules another `divide` (label 10, 13-15), and this process goes on until the condition checking fails. The time duration

between two divides, t_4 , is estimated to be 6-8 hours [45].

Figure B-1 shows both situations where the B cell eventually becomes a plasma cell, (upper timeline), or a memory cell (bottom timeline). Both the event `convert_plasma` (Table B-1 entry 11a) and event `convert_memory` ((Table B-1 entry 11b) switches the state, effectively voiding the last divide event (label 15) that requires the state to be `activated`.

The `convert_memory` ((Table B-1 entry 11b) event does not schedule any more events, as we assume that the memory cell is as long-lived as the human body itself. Though resting, a memory cell can be reactivated (re-entering the `activated` state) if it collides with some antigen that it matches.

The `convert_plasma` event ((Table B-1 entry 11a) schedules a `plasma_death` event (label 17), at which the plasma cell will die in 2-4 days [45], and a `produce_antibodies` event (labeled 16), where the antibodies are created. Antibody production is rapid, hence this process is more continuous than discrete, which implies that the interval of `produce_antibodies`, t_6 , is relatively smaller (in the order of seconds). The last event, `plasma_death`, causes cell death, and the plasma cell is removed from the system. In this case, the `produce_antibodies` event and the `naive_death` event which are still in the execution list will both be immediately cancelled.

B.2. Experiment setup and parameter settings

The models are developed in Java and all experiments are run on Windows XP system with 3.4GHz Pentium dual CPU and 2GB physical memory. The experiment is set up as follows.

- The environment is initially populated with 3000 naïve B cells (equivalent to the B cell count in 10 μ l of blood). Initially there are no antibodies.
- From day 1 to day 5, the environment remains free of antigens.
- From day 6 to day 10, a series of antigen binding events are scheduled to take place at a time interval of 0.001 days, which mimics the intrusion of antigens. This is also the period where the specific T helper cells undergo clonal expansion although they are not explicitly modeled here.
- From day 11 to day 13, a total of 5 B cell activation events take place at a time interval of 0.3 day. This mimics the fact that only a small subpopulation of B cells are capable of antigen binding,

hence are activated. The majority of B cells are not involved in the response.

- The population dynamics is sampled at a time interval of 0.2 days.
- The simulation is run for a duration of 40 days.
- The simulation is repeated for 100 times for each update scheme.

The parameter values are summarized in Table B-2. The event semantics of the time durations are given in Figure 4-6, Chapter 4. They follow a normal distribution (mean $\pm$ standard deviation), except for τ_1 (the B cell lifespan). The majority of the B cells (about 90%) are long-lived, with 1-2% turnover during each day [45]. Assuming that each B cell has a mortality rate of 1-2% per day, the individual B lifespan (τ_1) approximates an exponential distribution (with decay constant 0.015), the mean lifetime is then about 70 days.

Parameter	Values	Parameter	Values
t_1	70	p_1, p_2	0.95, 0.05
t_2	0.002 ± 0.001	t_{monitor}	0.2
t_3	0.015 ± 0.005	n_{AB}	1.7×10^5
t_4	0.3 ± 0.02	$n_{\text{B}}(\text{initial})$	3000
t_5	2.5 ± 0.2	$t_{\text{AB_loss}}$	0.001
t_6	0.001	$r_{\text{AB_loss}}$	3.08×10^{-5}
t_7	3 ± 0.5		

Table B-2 Parameter values (time unit: day)

Antibody production and loss are events that occur at the smallest timescale. They are numerous relative to the cell's replication and death events that we will consider them as continuous processes: Each plasma cell can produce 2000 antibody molecules per second [177], and the half-life of the IgG, the most abundant antibody isotype, is 21-24 days [179]. These effects are grouped as quantity changes. Correspondingly, the uniform time-step update scheme sets its time-step to 0.001 day.

Appendix C

Chemotaxis Experiment Settings

Our chemotaxis experiment that mimics an Under-Agarose assay is set up as follows. The virtual environment is a 3mm by 2mm enclosed 2-dimensional space. The space granularity ΔL (the distance between neighboring grid points) is set at $20\mu\text{m}$. The cell and attractant wells are both $250\mu\text{m}$ in radius and are placed 1mm apart. Initially, the cell well is filled with 1000 cells, and the attractant well is filled with attractant molecules at a concentration of 2500 particles per $20\mu\text{m} \times 20\mu\text{m}$ area (for a total of about 1.2×10^6 molecules). By assuming a well height of $50\mu\text{m}$, we create a cell concentration of 10^3 cells/ μL and a molecular concentration of 10^{-10} mol/L so as to be comparable to the original experiment setting [32]. The upper panel in Figure C-1 (same as Figure 5-6) shows the two-dimensional positioning of the wells, and the lower panel shows the M cell histogram along the x axis and the cross-sectional view (at central y position, $y=1000$) of the attractant distribution. The cell radius is $4\mu\text{m}$ and the maximum random speed is $2\mu\text{m}$ per minute. The molecules diffuse with diffusion constant $D = 10 \mu\text{m}^2\text{min}^{-1}$. Table C-1 summarizes the model parameters mentioned above and for the various chemotactic movement rules.

Figure C-1 Visualization of simulated chemotactic movement (same as Figure 5-6)

Figure C-1 shows a sample of the visualization of the simulation results. The three panels are aligned by their x coordinates. Panel (a) is the top view of the cell distribution (cells as black dots) and molecule distribution (green corresponds to high concentration and white corresponds to low concentration). It is cropped to appropriate size to fit the page without loss of information. Panel (b) displays the agents (cells) distribution in the form of a histogram, with a bin width of ΔL . The height of each bin corresponds to the number of cells whose x coordinates happen to fall in the range that bin represents. Panel (b) also displays the sectional view of the molecular concentration distribution (in green) with the sectional plane indicated in panel (a). Panel (b) is *fixed in height* and the figure is *not* downscaled when the concentration drops or the cell count decreases. This facilitates the comparison of simulation patterns at different time instances. Panel (c) displays the x component of the molecular gradient at the sectional plane (i.e., the first partial derivative of concentration over x direction, *positive* means gradient pointing to the right) and the relative gradient (i.e., gradient divided by concentration, *positive* means relative gradient pointing to the right). Panel (c) display is *relative in height* that the figures are always re-scaled as the values change, so that the displayed heights of the maximal gradient and maximal relative gradient are fixed. The maximum values are displayed in the legend.

Parameter	Value	Description
Δt	1 min	Update interval of molecular diffusions
ΔL	20 μm	Granularity of the site grid
D	10 $\mu\text{m}^2\text{min}^{-1}$	Diffusion constant of molecule L
λ	0 min^{-1}	Decay constant of molecule L (no spontaneous loss of molecules)
V	50 μm^2	Volume of M Cell
r	4 μm	Radius of M Cell
l	2 $\mu\text{m min}^{-1}$	Random motility of M cell: max movement speed
R_{tot}	50 000	The total amount of receptor molecules R per M cell
K_r	0.5 min^{-1}	The recycle rate of internalized receptor R^*
K_b	$2 \times 10^{-6} \text{ min}^{-1}$	The association rate of L and R
K_i	0.3 min^{-1}	The internalization rate of occupied receptor LR

Table C-1 Model parameters for simulating chemotaxis

To quantify the cell migration patterns, we measure the parameters shown in Figure 5-6(a). The x coordinates of the front and the rear of the M cell population are denoted by $x_{\max}$ and $x_{\min}$ respectively, and the mean x-coordinate of the entire cell population (i.e. center of mass) by x_{avg} . The initial values of x_{avg} , $x_{\max}$ and $x_{\min}$ are 1000, 1250, 750 respectively. L_x and L_y are the

extensions of the pattern in the x and y directions. Their ratio L_x/L_y is used as an estimate of the elongation of the migration pattern. Pattern elongation is caused by the speed difference between cells at the front and at the rear of the population. When the cell pattern is circular, L_x/L_y is 1. The larger the value of L_x/L_y , the more elongated the pattern.

The simulation model is implemented in Java. The simulation is run on a Windows XP Professional system with 2.4 GHz Pentium 4 CPU and 1GB of Memory. One single run of the simulation lasts 5000 time units unless otherwise stated, and usually takes approximately half a minute.

References

1. Griffith, W.T., *The Physics of Everyday Phenomena: A Conceptual Introduction to Physics*. 2001: New York: McGraw-Hill Higher Education.
2. *Experiment*. 2010 [cited; Available from: <http://en.wikipedia.org/wiki/Experiment>.
3. Johnson, P.D. and D.G. Besselsen, *Practical Aspects of Experimental Design in Animal Research*. ILAR Journal, 2002. **43**(4): p. 202-206.
4. Bailey, R.A., *Design of Comparative Experiments*. 2008: Cambridge University Press.
5. Solway, A., *Exploring forces and motion*. 2007: The Rosen Publishing Group.
6. *List of experiments*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/List_of_experiments.
7. *The Physics of Aristotle versus the Physics of Galileo*. 2010 [cited; Available from: http://csep10.phys.utk.edu/astr161/lect/history/aristotle_dynamics.html.
8. *Aristotelian physics*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Aristotelian_physics.
9. Fox, S.W. and K. Dose, *Molecular Evolution and the Origin of Life*. 1972: W.H Freeman and Company, San Francisco.
10. Wiener, P.P., *Dictionary of the History of Ideas*. 1973: New York: Charles Scribner's Sons.
11. *What is spontaneous generation?* 2010 [cited; Available from: <http://www.allaboutscience.org/what-is-spontaneous-generation-faq.htm>.
12. Dunn, P., *James Lind (1716-94) of Edinburgh and the treatment of scurvy*. Archive of Disease in Childhood Foetal Neonatal. **76**: p. 64-65.
13. *James Lind*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/James_Lind.
14. Carlisle, R., *Scientific American Inventions and Discoveries*. 2004, New Jersey: John Wiley & Sons, Inc.
15. Beck, H.P., S. Levinson, and G. Irons, *Finding Little Albert: A journey to John B. Watson's infant laboratory*. American Psychologist, 2009. **64**(7): p. 605-614.
16. *Litter Albert experiment*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Little_Albert.
17. Beall, A., *Microbiology lab automation arrives*. Medical Laboratory Observer, 2008. **Tuesday, July 1 2008**.
18. Schulz, K. and D. Grimes, *Generation of allocation sequences in randomised trials: chance, not choice*. Lancet, 2002. **359**(9305): p. 515-519.
19. Calinski, T. and S. Kageyama, *Block designs: A Randomization approach, Volume I: Analysis. Lecture Notes in Statistics. 150*. 2000: New York: Springer-Verlag.
20. Calinski, T. and S. Kageyama, *A Randomization approach, Volume II: Design. Lecture Notes in Statistics, 170*. 2003, New York: Springer-Verlag.
21. Hinkelmann, K. and O. Kempthorne, *Design and Analysis of Experiments, Volume I: Introduction to Experimental Design*. 2008: Wiley.
22. Orel, V., *Gregor Mendel: the first geneticist*. 1996: Oxford University Press.
23. Bardoe, C., *Gregor Mendel: The Friar who grew peas*. 2006: HN Abrams.
24. Chalmers, T., et al., *A method for assessing the quality of a randomized control trial*. Control Clin Trials, 1981. **2**(1): p. 31-49.

25. Bratman, S., *The Alternative Medicine Sourcebook*. 1997: Lowell House.
26. Keller, E.F. and L.A. Segel, *Model for chemotaxis*. J. Theor. Biol., 1971. **30**(2): p. 225-234.
27. Zigmond, S.H., J.L. Sullivan, and D.A. Lauffenburger, *Kinetic analysis of chemotactic peptide receptor modulation*. J. Cell Biol., 1982. **92**: p. 34-43.
28. Lauffenburger, D.A., C. Rothman, and S.H. Zigmond, *Measurement of leukocyte motility and chemotaxis parameters with a linear Under-Agarose Migration Assay*. J.Immunol., 1983. **131**(2): p. 940-947.
29. Devreotes, P.N. and S.H. Zigmond, *Chemotaxis in eukaryotic cells: A focus on leukocytes and dictyostelium*. Ann. Rev. Cell Biol., 1988. **4**: p. 649-686.
30. Eisenbach, M., et al., *Chemotaxis*. 2004: Imperial College Press, London.
31. Van Haastert, P.J.M. and P.N. Devreotes, *Chemotaxis: signalling the way forward*. Nature Reviews: Molecular Cell Biology, 2004. **5**: p. 626-634.
32. Heit, B. and P. Kubes, *Measuring chemotaxis and chemokinesis: the under-agarose cell migration assay*. Sci. STKE, 2003. **170**: p. p15.
33. Tranquillo, R.T., S.H. Zigmond, and D.A. Lauffenburger, *Measurement of the chemotaxis coefficient for human neutrophils in the under-agarose migration assay*. Cell Motil. Cytoskeleton, 1988. **11**: p. 1-15.
34. Jones, G.E., *Cellular signalling in macrophage migration and chemotaxis*. J. Leukoc. Biol., 2000. **68**: p. 593-602.
35. Keller, E.F. and L.A. Segel, *Traveling bands of chemotactic bacteria: a theoretical analysis*. J. Theor. Biol., 1971. **30**(2): p. 235-248.
36. Sherratt, J.A., *Chemotaxis and chemokinesis in eukaryotic cells: the Keller-Segel equations as an approximation to a detailed model*. Bul. Math. Biol., 1994. **56**(1): p. 129-146.
37. Tijms, H.C., *A First Course in Stochastic Models Chapter 9: Algorithmic Analysis of Queues*. 2003: Wiley, Chichester.
38. *Scientific Modeling - the process of generating a model*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Scientific_modelling#The_process_of_generating_a_model].
39. Rappaport, S., *Abstraction and Unrealistic Assumptions in Economics*. Journal of Economic Methodology, 1996. **3**(2): p. 215-236.
40. Boland, L.A., *Assumptions controversy*, in *The New Palgrave Dictionary of Economics, Second Edition.*, S.N.D.a.L.E. Blume, Editor. 2008, Palgrave Macmillan.
41. *Occam's razor*, in *Merriam-Webster's Collegiate Dictionary (11th ed.)*. 2003, New York: Merriam-Webster.
42. Camerer, C., G. Loewenstein, and M. Rabin, *Advances in Behavioral Economics*. 2003.
43. Diamond, P.A. and H. Vartiainen, *Behavioral Economics and its Applications*. 2007.
44. Kleinstein, S.H. and P.E. Seiden, *Simulating the immune system*. Computing in science and Engineering, 2000. **2**: p. 69-77.
45. Janeway, C.A., et al., *Immunobiology: the immune system in health & disease*. 2001: Garland Science Publishing.
46. Kidd, P., *Th1/Th2 balance: the hypothesis, its limitations, and implications for health and disease*. Alternative Medicine Review, 2003. **8**(3): p. 223-246.
47. Harrington, L., et al., *Interleukin 17-producing CD4+ effector T cells develop via a lineage distinct from the T helper type 1 and 2 lineages*. Nat Immunol., 2005. **6**(11): p. 1069-1070.
48. Ariew, R., *Ockham's Razor: A Historical and Philosophical Analysis of Ockham's Principle of Parsimony*. 1976: Champaign-Urbana, University of Illinois.
49. Hacking, I., *Telepathy: Origins of Randomization in Experimental Design*. Isis, 1988. **79**: p.

427-451.

50. Hopewell, S., et al., *The quality of reports of randomised trials in 2000 and 2006: comparative study of articles indexed in PubMed*. BMJ, 2010. **340**: p. c273.
51. *Blind Experiment*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Blind_experiment.
52. Lachin, J., J. Matts, and L. Wei, *Randomization in clinical trials: conclusions and recommendations*. Control Clin Trials, 1988. **9**(4): p. 365-374.
53. *Random Number Generation*. 2010 [cited; Available from: http://www.gnu.org/software/gsl/manual/html_node/Random-Number-Generation.html.
54. Tay, J.C., H. Wibisono, and P.A. Tambyah, *Collection and Analysis of Behavioral Data for Simulating Nosocomial Transmission of Infectious Diseases*. 2009.
55. Gendler, T.S., *Thought Experiment: On the Powers and Limits of Imaginary Cases*. 2000: Garland (New York).
56. Buzzoni, M., *Thought Experiment in the Natural Sciences*. 2008: Koenigshausen+Neumann, Wuerzburg.
57. *Thought experiment*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Thought_experiment.
58. *Immune system*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Immune_system#Helper_T_cells.
59. De Boer, R. and A.S. Perelson, *Target cell limited and immune control models of HIV infection: a comparison*. Journal of Theoretical Biology, 1998(169): p. 375-390.
60. Weninger, W., et al., *Migratory properties of naive, effector, and memory CD8+ T cells*. J. Exp. Med., 2001. **194**(7): p. 953-966.
61. Hershberg, U., et al., *HIV time hierarchy: winning the war while, loosing all the battles*. Physica A, 178-190. **289**: p. 178-190.
62. Nowak, M.A. and A.J. McMichael, *How HIV defeats the immune system*. Scientific American, 1995. **273**: p. 58-65.
63. Rowland-Jones, S.L., *AIDS pathogenesis: what have two decades of HIV research taught us?* Nature Reviews Immunology, 2003. **3**: p. 343-348.
64. Fauci, A.S. and T.-W. Chun, *Latent reservoirs of HIV: obstacles to the eradication of virus*. PNAS, 1999. **96**: p. 10958-10961.
65. Montagnier, L., *A history of HIV discovery*. Science, 2002. **298**(5599): p. 1727-1728.
66. Gougeon, M.-L., *Apoptosis as an HIV strategy to escape immune attack*. Nature Reviews Immunology, 2003. **3**: p. 392-404.
67. Cyster, J.G., *Chemokines and the homing of dendritic cells to the T cell areas of lymphoid organs*. J. Exp. Med., 1999. **189**(3): p. 447-450.
68. Sanchez-Madrid, F. and M.A. Del Pozo, *Leukocyte polarization in cell migration and immune interactions*. EMBO J., 1999. **18**(3): p. 501-511.
69. Parent, C.A. and P.N. Devreotes, *A Cell's sense of direction*. Science, 1999. **284**(5415): p. 765-770.
70. Haddadin, A.S., S.A. Fappiano, and P.A. Lipsett, *Methicillin resistant Staphylococcus aureus (MRSA) in the intensive care unit*. British Medical Journal, 2002. **78**: p. 385.
71. Rabound, J., et al., *Modelling Transmission of MRSA Among Patients Admitted to a Hospital*. Infection Control and Hospital Epidemiology, 2005. **26**(7): p. 607-615.
72. Kunori, T., et al., *Cost-effectiveness of different MRSA screening methods*. Journal of Hospital Infection, 2002. **51**: p. 189-200.

73. WHO, *WHO Guidelines on Hand Hygiene in Health Care*. 2009, World Health Organization.
74. Cooper, B.S., et al., *Isolation measures in the hospital management of methicillin resistant Staphylococcus aureus (MRSA): systematic review of the literature*. BMJ, 2004. **329**(7456): p. 533.
75. Cepeda, J., et al., *Isolation of patients in single rooms or cohorts to reduce spread of MRSA in intensive-care units: prospective two-centre study*. Lancet, 2005. **365**(9456): p. 295-304.
76. Perelson, A.S. and P.W. Nelson, *Mathematical analysis of HIV-1 dynamics in vivo*. SIAM Review, 1999. **41**: p. 3-44.
77. Perelson, A.S., *Immunology for physicists*. Review of Modern Physics, 1997. **69**: p. 1219-1267.
78. Perelson, A.S., *Modelling viral and immune system dynamics*. Nature Reviews Immunology, 2002. **2**: p. 28-36.
79. Ho, D.D., et al., *Rapid turnover of plasma virions and CD4 lymphocytes in HIV-1 infection*. Nature, 1995. **373**: p. 123-126.
80. Bonabeau, E., *Agent-based modeling: Methods and techniques for simulating human systems*. PNAS, 2002. **99**(suppl 3): p. 7280-7287.
81. Bonebeau, E., *Agent-based modeling: methods and techniques to simulate human systems*. PNAS, 2002. **99**(3): p. 7280-7287.
82. Grilo, A., A. Caetano, and A. Rosa, *Agent based artificial immune system*, in *Genetic and Evolutionary Computation Conference*. 2001.
83. Lees, M., et al., *Simulating agent-based systems with HLA: the case of SIM AGENT Part-II*, in *Proceedings of the 2003 European Simulation Interoperability Workshop*. 2003: Stockholm, Sweden.
84. Luck, S., et al., *MASON: A multiagent Simulation Environment*. Simulation, 2005. **81**(7): p. 517-527.
85. Sloot, P.M.A. and A.G. Hoekstra, *Multi-scale modelling in computational biomedicine*. Brief in Bioinform, 2010. **11**(nr 1): p. 142-152.
86. An, G., *Introduction of an agent-based multi-scale modular architecture for dynamic knowledge representation of acute inflammation*. Theoretical Biology and Medical Modelling, 2008. **5**(11).
87. Castella, J.-C., T.N. Trung, and S. Boissau, *Participatory simulation of land-use changes in the northern mountains of Vietnam: the combined use of an agent-based model, a role-playing game, and a geographic information system*. Ecology and Society, 2005. **10**(1): p. 27.
88. Balcan, D., et al., *Modeling the spatial spread of infectious diseases: the GLocal Epidemic and Mobility computational model*. J Comput Sci., 2011. **1**(3): p. 132-145.
89. Rana, O.F. and K. Stout. *What is scalability in multi-agent systems?* in *Proceedings of the fourth international conference on Autonomous agents*. 2000. Barcelona, Spain: ACM New York, NY, USA.
90. GORDON, D.M., B.C. GOODWIN, and L.E.H. TRAINOR, *A parallel distributed model of the behaviour of ant colonies*. Journal of Theoretical Biology, 1992. **156**: p. 293-307.
91. Scheutz, M. and P. Schermerhorn, *Adaptive algorithms for dynamic distribution and parallel execution of agent-based models*. Journal of Parallel and Distributed Computing, 2006. **66**(8): p. 1037-1051.
92. Logan, B. and G. Theodoropoulos, *The distributed simulation of multi-agent systems*. Proceedings of the IEEE, 2001. **89**(2): p. 174-186.
93. Da-Hun, T., et al., *Parallel computing platform for agent-based modeling of multicellular biological systems*. LNCS, 2004. **2210**: p. 5-8.
94. McCune, J.M., *The dynamics of CD4+ T-cell depletion in HIV disease*. Nature, 2001. **410**: p.

- 974-979.
95. Roitt, I.M. and P.J. Delves, *Essential Immunology*. 2001: Blackwell Science.
 96. Stine, G.J., *AIDS Update*. 1999: Prentice Hall.
 97. Banchereau, J., *The long arm of the immune system*. Scientific American, 2002. **287**: p. 52-59.
 98. *HIV (wikipedia)*. 2010 [cited; Available from: <http://en.wikipedia.org/wiki/HIV>].
 99. Coffin, J.M., *Structure and Classification of Retroviruses*, in *The Retroviridae*, J.A. Levy, Editor. 1992, New York: Plenum Press.
 100. Selliah, N. and T.H. Finkel, *Biochemical mechanisms of HIV induced T cell apoptosis*. Cell Death and Differentiation, 2001. **8**: p. 127-136.
 101. Heinkelein, M., S. Sopper, and C. Jassoy, *Contact of human immunodeficiency virus type 1-infected and uninfected CD4+ T lymphocytes s highly cytolytic for both cells*. Journal of Virology, 1995. **69**(6925-6931).
 102. Ban, N., et al., *Crystal structure of an idiotype-anti-idiotype Fab complex*. PNAS, 1994. **91**: p. 1604-1608.
 103. *HIV Virus*. 2010 [cited; Available from: <http://www.hiv-info.org/hiv-virus.htm>].
 104. *The size of a T-cell?* 2010 [cited; Available from: http://wiki.answers.com/Q/The_size_of_a_t-cell].
 105. Poncelet, P., et al., *Surface CD4 density remains constant on lymphocytes of HIV-infected patients in the progression of disease*. Res Immunol., 1991. **142**(4): p. 291-298.
 106. Boyce, J.M., *Understanding and controlling Methicillin-resistant Staphylococcus aureus infections*. Infect Control Hosp Epidemiol, 2002. **23**: p. 485-487.
 107. Cookson, B.D., *Methicillin-Resistant Staphylococcus aureus in the Community: New Battlefronts, or Are the Battles Lost?* Infection Control and Hospital Epidemiology, 2000. **21**(6): p. 398-403.
 108. Coia, J.E., et al., *Guidelines for the control and prevention of meticillin-resistant Staphylococcus aureus (MRSA) in healthcare facilities*. Journal of Hospital Infection, 2006. **63**(Supplement 1): p. S1-S44.
 109. Blythe, D., et al., *Environmental contamination due to methicillin-resistant Staphylococcus aureus (MRSA)*. Journal of Hospital Infection, 1998. **38**(1): p. 67-69.
 110. Chaix, C., et al., *Control of Endemic Methicillin-Resistant Staphylococcus aureus A Cost-Benefit Analysis in an Intensive Care Unit*. Am Med Assoc, 1999. **282**: p. 1745-1751.
 111. Thompson, D.S., *Methicillin-resistant Staphylococcus aureus in a general intensive care unit*. Royal Soc Med, 2004. **97**: p. 521-526.
 112. Klevens, R.M., et al., *Estimating health care-associated infections and deaths in US hospitals, 2002.*, in *Public Health Report*. 2007. p. 160-166.
 113. Nulens, E., et al., *Cost of the methicillin-resistant Staphylococcus aureus search and destroy policy in a Dutch university hospital*. Journal of Hospital Infection, 2008. **68**: p. 301-307.
 114. Leth, R.A. and J.K. Moller, *Surveillance of hospital-acquired infections based on electronic hospital registries*. Journal of Hospital Infection, 2006. **62**: p. 71-79.
 115. Paule, S.M., et al., *Chromogenic Media vs Real-Time PCR for Nasal Surveillance of Methicillin-Resistant Staphylococcus aureus: Impact on Detection of MRSA-Positive Persons*. Am J Clin Pathol, 2009. **131**(532-539).
 116. Pettinger, A. and M.D. Nettleman, *Epidemiology of Isolation Precautions*. Infection Control and Hospital Epidemiology, 1991. **12**(5): p. 303-307.
 117. Jernigan, J.A., et al., *Effectiveness of Contact Isolation during a Hospital Outbreak of Methicillin resistant Staphylococcus aureus*. Am. J. Epidemiol., 1996. **143**(5): p. 496-504.
 118. D'Agata, E.M., G. Webb, and M. Horn, *A mathematical model quantifying the impact of antibiotic*

- exposure and other interventions on the endemic prevalence of vancomycin-resistant enterococci.* J Infect Dis, 2005. **192**: p. 2004-2011.
119. Hassan, Z.M., et al., *Hand Hygiene Compliance and Nurse-Patient Ratio Using Videotaping and Self Report.* Infectious Diseases in Clinical Practice, 2009. **17**(4): p. 243-247.
 120. Pittet, D., et al., *Hand Hygiene among Physicians: Performance, Beliefs, and Perceptions.* Ann Intern Med., 2004. **141**(1-8).
 121. Pittet, D., et al., *Effectiveness of a hospital-wide programme to improve compliance with hand hygiene.* The Lancet, 2000. **356**: p. 1307-1312.
 122. Pittet, D., *Compliance with hand disinfection and its impact on hospital-acquired infections.* Journal of Hospital Infection, 2001. **48**(Supplement A): p. S40-S46.
 123. Beggs, C., et al., *The influence of nurse cohorting on hand hygiene effectiveness.* Am J Infect Control., 2006. **34**(1): p. 621-626.
 124. Pittet, D., P. Mourouga, and T.V. Perneger, *Compliance with Handwashing in a Teaching Hospital.* Annals of Internal Medicine, 1999. **130**(2): p. 126-134.
 125. Nijssen, S., et al., *Relative risk of physicians and nurses to transmit pathogens in a medical intensive care unit.* Arch Intern Med, 2003. **163**: p. 2785-2786.
 126. *Staphylococcus.* 2010 [cited; Available from: <http://en.wikipedia.org/wiki/Staphylococcus>.
 127. Rohr, U., et al., *Methicillin-resistant Staphylococcus aureus whole-body decolonization among hospitalized patients with variable site colonization by using mupirocin in combination with octenidine dihydrochloride.* J Hosp Infect, 2003. **54**(4): p. 305-309.
 128. Hannon, B., M. Ruth, and S.A. Levin, *Modeling Dynamic Biological Systems.* 1997: Springer.
 129. Zhigilei, L.V., P.B.S. Kodali, and B.J. Garriso, *Molecular Dynamics Model for Laser Ablation and Desorption of Organic Solids.* The journal of physical chemistry B, 1997. **101**(11): p. 2028-2037.
 130. Li, H., et al., *Emergence of Preferred Structures in a Simple Model of Protein Folding.* Science, 1996. **273**(5275): p. 666-669.
 131. Phillips, A.N., *Reduction of HIV concentration during acute infection: independence from a specific immune response.* Science, 1996. **271**: p. 497-499.
 132. Wei, X., et al., *Viral dynamics in human immunodeficiency virus type 1 infection.* Nature, 1995. **373**: p. 117-122.
 133. Perelson, A.S., et al., *HIV-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time.* Science, 1996. **271**: p. 1582-1586.
 134. Stafford, M.A., et al., *Modeling plasma virus concentration during primary HIV infection.* Journal of Theoretical Biology, 2000. **203**: p. 285-301.
 135. Grilo, A., A. Caetano, and A. Rosa, *Immune system simulation through a complex adaptive system model.* Proceedings of the 3rd workshop on genetic algorithms and artificial life, 1999.
 136. Sieburg, H.B., et al., *Simulation of HIV infection in artificial immune systems.* Physica D, 1990. **45**: p. 208-227.
 137. Jacob, C., J. Litorco, and L. Lee, *Immunity through Swarms: agent-based simulations of the human immune system.* Proceedings of 3rd International Conference on Artificial Immune Systems, 2004: p. 400-412.
 138. Folcik, V.A., et al., *Using an agent-based model to analyze the dynamic communication network of the immune response.* Theoretical Biology and Medical Modelling, 2011. **8**(1).
 139. Macal, C.M. and M.J. North, *Tutorial on agent-based modelling and simulation.* Journal of Simulation, 2010. **4**: p. 151-162.
 140. Wein, L.M., R.M. D'Amato, and A.S. Perelson, *Mathematical analysis of antiretroviral therapy aimed at HIV-1 eradication or maintenance of low viral loads.* Journal of Theoretical Biology,

1998. **192**: p. 81-98.
141. Nowak, M.A., et al., *Antigenic diversity thresholds and the development of AIDS*. Science, 1991. **254**: p. 963-969.
 142. Nowak, M.A., R.M. May, and R.M. Anderson, *The evolutionary dynamics of HIV-1 quasispecies and the development of immunodeficiency disease*. AIDS, 1990. **4**: p. 1095-1103.
 143. Perelson, A.S., et al., *Decay characteristics of HIV-1-infected compartments during combination therapy*. Nature, 1997. **387**: p. 188-191.
 144. Essunger, P. and A.S. Perelson, *Modeling HIV infection of CD4+ T-cell subpopulations*. Journal of Theoretical Biology, 1994. **169**: p. 375-390.
 145. Guo, Z., H.K. Han, and J.C. Tay, *Sufficiency verification of HIV-1 pathogenesis based on multi-agent simulation*, in *GECCO 2005*. 2005: Washington D.C.
 146. Axtell, R.L., *Effects of interaction topology and activation regime in several multi-agent systems*, in *the Second International Workshop on Multi-Agent-Based Simulation-Revised and Additional Papers*. 2000: London, UK.
 147. Jaffry, S.W. and J. Treur, *Agent-based and Population-based simulation: A Comparative case study for epidemics*.
 148. Casal, A., et al., *Agent-based modeling of the context dependency in T cell recognition*. Journal of Theoretical Biology, 2005. **236**: p. 376-391.
 149. Holland, J.H., *Emergence from Chaos to Order*. 1998: Oxford University Press.
 150. Corning, P.A., *The Re-Emergence of Emergence: A Venerable Concept in Search of a Theory*. Complexity, 2002. **7**(6): p. 18-30.
 151. Clayton, P. and P. Davies, *The Re-Emergence of Emergence: The Emergentist Hypothesis from Science to Religion*. 2006: Oxford University Press.
 152. Kremers, E., et al., *Agent-Based Simulation of Wind Farm Generation at Multiple Time Scales*, in *Wind Farm - Impact in Power System and Alternatives to Improve the Integration*, G.O. Suvire, Editor. 2011, InTech.
 153. Evansa, T.P. and H. Kelleya, *Multi-scale analysis of a household level agent-based model of landcover change*. Journal of Environmental Management, 2004. **72**(1-2): p. 57-72.
 154. Baldazzi, V., F. Castiglione, and M. Bernaschi, *An enhanced agent based model of the immune system response*. Cellular Immunology, 2006. **244**(2): p. 77-79.
 155. Kirschner, D.E., et al., *Toward a multiscale model of antigen presentation in immunity*. Immunological Reviews, 2007. **216**(1): p. 93-118.
 156. Bernaschi, M. and F. Castiglione, *Computational features of agent-based models*. International Journal of Computational Methods, 2005. **2**(1).
 157. Lysenko, M. and R.M. D'Souza, *A Framework for Megascale Agent Based Model Simulations on Graphics Processing Units*. Journal of Artificial Societies and Social Simulation, 2008. **11**(4).
 158. Massaiolt, F., F. Castiglione, and M. Bernaschi, *OpenMP Parallelization of Agent-Based Models*. OpenMP, 2005. **31**(10): p. 1068-1081.
 159. Som, T.K. and R.G. Sargent, *Model structure and load balancing in optimistic parallel discrete event simulation*, in *Proceedings of the fourteenth workshop on parallel and distributed simulation*. 2000. p. 147-154.
 160. Berger, T., *Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis*. Agricultural Economics, 2001. **25**(2/3): p. 245-260.
 161. Grilo, C. and L. Correia, *The Influence of Asynchronous Dynamics in the Spatial Prisoner's Dilemma Game*. Lecture Notes in Computer Science, 2008. **5040/2008**: p. 362-371.

162. Ingerson, T.E. and R.L. Buvel, *Structure in asynchronous cellular automata*. Physica D, 1982. **10**: p. 59-68.
163. Banks, J., J.S. Carson II, and B.L. Nelson. *Discrete-Event System Simulation (2nd ed.)*. 1999: Prentice Hall.
164. Stauffer, D., *Computer simulations of cellular automata*. J. Phys. A: Math. Gen., 1991. **24**: p. 909-927.
165. Schiff, J.L., *Cellular Automata: A Discrete View of the World*. 2008: Wiley & Sons, Inc.
166. Harvey, I. and T. Bossomaier. *Time out of joint: Attractors in asynchronous random boolean networks*. in *the Fourth European conference on Artificial Life (ECAL97)*. 1997.
167. Gershenson, C. *Classification of Random Boolean Networks*. in *Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life*. 2002. Sydney, Australia: MIT Press.
168. Kaneko, K., *Overview of Coupled Map Lattices*. Chaos, 1992. **2**(3): p. 279.
169. Xu, J. and X.F. Wang, *Cascading failures in scale-free coupled map lattices*. IEEE International Symposium on Circuits and Systems, 2005. **4**: p. 3395 - 3398.
170. Delgado, J. and R.V. Sole, *Self-synchronization and task fulfillment in ant colonies*. Journal of Theoretical Biology, 2000. **205**: p. 433-441.
171. Bonabeau, E., G. Theraulaz, and J.-L. Deneubourg, *The synchronization of recruitment-based activities of ants*. BioSystems, 1998. **45**: p. 195-221.
172. Huberman, B.A. and N.S. Glance, *Evolutionary Games and Computer Simulations*. PNAS, 1993. **90**: p. 7716-7718.
173. Valentin, A., et al., *CXCR4 Mediates Entry and Productive Infection of Syncytia-Inducing (X4) HIV-1 Strains in Primary Macrophages*. Virology, 2000. **269**: p. 294-304.
174. Cornforth, D., D.G. Green, and D. Newth, *Ordered asynchronous processes in multi-agent systems*. Physica D, 2005. **204**: p. 70-82.
175. Schönfisch, B. and A. De Roos, *Synchronous and asynchronous updating in cellular automata*. BioSystems, 1999. **51**: p. 123-143.
176. Ball, P.D., *Introductoin to discrete event simulation*, in *Second DYCOMANS workshop on "Management and Control: Tools in Action"*. 1996: Algarve, Portugal.
177. Merieb, E.N., *Essentials of human anatomy and physiology*. 8th ed. 2006: Pearson.
178. Guyton, A.C. and J.E. Hall, *Textbook of medical physiology (10th ed.)*. 2000: Philadelphia: Saunders.
179. Leffell, M.S., A.D. Donnenberg, and N.R. Rose, *Handbook of human immunology*. 1997: Boca Raton, New York: CRC Press.
180. Tay, J.C., et al., *The Effect of Staffing Levels on Hand Hygiene Adherence: New Insights from an Evidence-based Behavioural Model of MRSA Transmission within an SICU (In preparation)*. 2010.
181. *Reaction-diffusion system*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Reaction-diffusion_system].
182. Cyster, J.G., *Leukocyte mirgration: scent of the T zone*. Curr. Biol., 2000. **10**(1): p. R30-R33.
183. Wei, S.H., et al., *A stochastic view of lymphocyte motility and trafficking within the lymph node*. Immunol. Rev., 2003. **195**: p. 136-159.
184. Sallusto, F., et al., *Rapid and coordinated swtich in chemokine receptor expression during dendritic cell maturation*. Eur. J. Immunol., 1998. **28**(9): p. 2760-2769.
185. Sozzani, S., et al., *Cutting edge: differential regulation of chemokine receptors during dendritic cell maturation: a model for their trafficking properties*. J. Immunol., 1998. **161**: p. 1083-1086.
186. *Law of mass action*. 2010 [cited; Available from:

- http://en.wikipedia.org/wiki/Law_of_mass_action.
187. Levere, T., H., *Affinity and Matter – Elements of Chemical Philosophy 1800–1865*. 1971: Gordon and Breach Science Publishers.
 188. Philibert, J., *One and a Half Century of Diffusion: Fick, Einstein, before and beyond*. Diffusion Fundamentals, 2005. **2**: p. 1.1-1.10.
 189. Heath, M.T., *Scientific computing: an introductory survey*. 2005: McGraw Hill.
 190. Klabunde, R.E. *Cardiovascular Physiology Concepts* 2006 [cited; Available from: <http://www.cvphysiology.com/Hemodynamics/H006.htm>.
 191. Mathews, C.K., K.E. van Holde, and K.G. Ahern, *Biochemistry*. 3rd ed. 2000: Addison Wesley Longman.
 192. Guyton, A.C. and J.E. Hall, *Textbook of medical physiology*. 10th ed. 2000, Philadelphia: Saunders. xxxii, 1064.
 193. Firtel, R.A. and C.Y. Chung, *The molecular genetics of chemotaxis: sensing and responding to chemoattractant gradients*. BioEssays, 2000. **22**(7): p. 603-615.
 194. Lin, F., et al., *Effective neutrophil chemotaxis is strongly influenced by mean IL-8 concentration*. Biochem. Biophys. Res. Commun., 2004. **319**(2): p. 576-581.
 195. Mitchison, T.J. and L.P. Cramer, *Actin-based cell motility and cell locomotion*. Cell, 1996. **84**(3): p. 371-379.
 196. Hoffman, R.D., et al., *Stereospecific chemoattraction of lymphoblastic cells by gradients of lysophosphatidylcholine*. PNAS, 1982. **79**: p. 3285-3289.
 197. *Influenza*. [cited; Available from: <http://en.wikipedia.org/wiki/Influenza>.
 198. WHO, *Consensus Document on the Epidemiology of Severe Acute Respiratory Syndrome (SARS)*. 2003.
 199. Yu, I.T.S., et al., *Evidence of airborne transmission of severe acute respiratory syndrome virus*. New Eng. J. Med., 2004. **350**: p. 1731-1739.
 200. Li, Y., et al., *Factors affecting ventilation effectiveness in SARS wards*. Hong Kong Med J., 2008. **14**(1 supplement 1): p. 33-36.
 201. Li, Y., et al., *Role of air distribution in SARS transmission during the largest nosocomial outbreak in Hong Kong*. Indoor Air, 2004. **15**: p. 83-95.
 202. Chow, T.T. and X.Y. Yang, *Ventilation performance in operating theatre against airborne infection: review of research activities and practical guidance*. Journal of Hospital Infection, 2004. **56**(2): p. 85-92.
 203. Ong, B.S., et al., *An Individual-Based Model of Influenza in Nosocomial Environments*, in *Computational Science - ICCS 2008*. 2008. p. 590-599.
 204. Gardner, M., *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"*. Scientific American, 1970. **223**(10): p. 120-123.
 205. Lee, J., et al., *Asynchronous game of life*. Physica D: Nonlinear Phenomena, 2004. **194**(3-4).
 206. *Hex Life*. 2010 [cited; Available from: <http://www.well.com/~dgb/hexlife.html>.
 207. *Cellular Automata and the Game of Life in the Hexagonal Grid*. 2010 [cited; Available from: <http://www.cse.sc.edu/~bays/h6h6h6/>.
 208. Giarratano, J.C. and G.D. Riley, *Expert Systems: Principles and Programming*. 2005: Thomson Learning, Canada.
 209. *Colony-forming unit*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Colony-forming_unit.
 210. Gershenson, C. *Classification of Random Boolean Networks*. in *Artificial Life VIII*. 2002: MIT Press.

211. Tay, J.C., H. Wibisono, and P.A. Tambyah, *Tracking and Analysis of Staff Movement for Simulating Transmission of MRSA within an SICU*. In preparation, 2009.
212. Kleijnen, J., et al., *A User's Guide to the Brave New World of Designing Simulation Experiments*, in *INFORMS Journal on Computing*. 2005. p. 263-289.
213. *Guidelines on Hand Hygiene in Healthcare*. 2009, World Health Organization.
214. Averill, M.L., *How to build valid and credible simulation models*, in *Proceedings of the 40th Conference on Winter Simulation*. 2008, Winter Simulation Conference: Miami, Florida.
215. *Does cohort nursing help control healthcare acquired infection?* Policy+, 2007(4).
216. Mach, E., *On Thought Experiments*, in *Knowledge and Error: Sketches on the Psychology of Enquiry*, E. Mach, Editor. 1976, D. Reidel Publishing Co., (Dordrecht). p. 134-147.
217. *Schrödinger's cat*. 2010 [cited; Available from: http://en.wikipedia.org/wiki/Schrödinger's_cat.
218. Cohen, M., *Wittgenstein's Beetle and Other Classic Thought Experiments*. 2005: Blackwell, (Oxford). 55-56.
219. Huth, M. and M. Ryan, *Logic in Computer Science: Modelling and Reasoning About Systems*. 2004: Cambridge University Press.
220. Cooper, B.S., et al., *Methicillin-resistant Staphylococcus aureus in hospitals and the community: Stealth dynamics and control catastrophes*. Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**: p. 10223-10228.
221. Buyya, R., et al., *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*. Future Generation Computer Systems, 2009. **25**(6): p. 599-616.
222. Nurmi, D., et al. *The Eucalyptus Open-Source Cloud-Computing System*. in *CCGRID '09. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. 2009. Shanghai.
223. Bousquet, F., et al., *Multi-agent systems and role games: collective learning processes for ecosystem management*, in *Complexity and ecosystem management: the theory and practice of multi-agent systems*. 2002, Edward Elgar Publishing, Cheltenham, UK. p. 248-285.
224. Janssen, M.A. and E. Ostrom, *Empirically based, agent-based models*. Ecology and Society, 2006. **11**(2): p. 37.
225. Étienne, M., *A multiple target role-playing game to assess negotiation processes in sylvopastoral management planning*. JASSS-The Journal of Artificial Societies and Social Simulation 2003. **6**(2): p. 5.
226. Cederman, L.K., *Modeling the size of wars: from billiard balls to sandpiles*. American Political Science Review, 2002. **97**(1): p. 19-59.
227. Brown, D. and D. Robinson, *Effects of heterogeneity in residential preferences on an agent-based model of urban sprawl*. Ecology and Society, 2006. **11**(1): p. 46.