A SPATIAL AGENT-BASED MODEL
AND A MULTI-DIMENSIONAL DATA WAREHOUSE
FOR MALARIA RESEARCH

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

S. M. Niaz Arifin

______________________________
Gregory R. Madey, Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana
September 2013
A SPATIAL AGENT-BASED MODEL
AND A MULTI-DIMENSIONAL DATA WAREHOUSE
FOR MALARIA RESEARCH

Abstract

by

S. M. Niaz Arifin

Human malaria is transmitted only by female Anopheles mosquitoes, which are regarded as the primary vector for transmission. Agent-Based Modeling (ABM) is a powerful tool for representing complex biological systems. Following a biological core model of An. gambiae for malaria entomology, we develop agent-based simulation models (ABMs). For the ABMs, we describe the verification & validation (V&V) processes, a spatial extension, and a landscape generator tool. Using the spatial ABM, we demonstrate the effects of spatial heterogeneity, investigate the impact of vector control interventions, and describe an example application by integrating the ABM with a geographic information system (GIS).

For malaria, there is a growing need to integrate the existing heterogeneous data sources. Data warehouses (DWs) and dimensional modeling (DM) can be viable alternatives to meet this challenge. We describe the design and implementation of a multi-dimensional DW, to be integrated into the Vector Ecology and Control Network (VECNet) Cyberinfrastructure (CI) analytical framework, which provides a robust way to store, access, and analyze malaria-related data.
DEDICATION

To my parents:

Engineer S. M. Golam Mostofa, B.Sc. Engg. (Civil), FIE (B), PGD (CS)
My Father and Guide

Professor Parvin Akhter Jahan, M.A. (Economics), B.A. (Honors)
My Mother and Best Friend

my wife:

Rumana Reaz Arifin, B.S., M.S.
My Soulmate

and my sister:

Mafruhatul Jannat, B.S., M.S.
We Grew up Together
5.2.2 Landscapes ............................................................. 90
5.2.3 Movement of Agents .................................................. 93
5.2.4 Resource-Seeking Events ............................................. 94
5.2.5 Simulations: Assumptions and Notation ......................... 96
5.3 Landscape Generator Tools ............................................. 96
5.4 Results .................................................................. 98
  5.4.1 Model Verification ................................................... 100
  5.4.2 Landscape Patterns .................................................. 101
  5.4.3 Relative Sizes of Resources ...................................... 102
  5.4.4 Resource Density .................................................... 104
  5.4.5 Combined Carrying Capacity (CCC) ......................... 106
5.5 Spatial Heterogeneity .................................................... 107
5.6 Summary ................................................................ 111

CHAPTER 6: VECTOR CONTROL INTERVENTIONS ...................... 113
  6.1 Overview ............................................................... 113
    6.1.1 Larval Source Management (LSM) ....................... 116
    6.1.2 Insecticide-Treated Nets (ITNs) ......................... 116
  6.2 Replicability of Simulations ......................................... 117
    6.2.1 Replicated Simulation Runs .............................. 118
    6.2.2 Boundary Type of Landscapes ......................... 119
  6.3 Multiple Definitions of ITN Coverage ............................ 120
    6.3.1 Partial Coverage with Single Chance ............... 122
    6.3.2 Partial Coverage with Multiple Chances ............ 122
    6.3.3 Complete Coverage ......................................... 125
    6.3.4 Population Profiles for ITNs ......................... 125
  6.4 Combined Interventions ............................................... 127
  6.5 Assumptions and Notations .......................................... 129
    6.5.1 Assumptions .................................................... 130
    6.5.2 Simulations ...................................................... 131
  6.6 Applying LSM and ITNs ............................................... 131
    6.6.1 Applying LSM in Isolation .............................. 132
    6.6.2 Applying ITNs in Isolation ............................... 133
    6.6.3 Applying LSM and ITNs in Combination ............. 135
  6.7 Results: Impact of LSM ............................................... 138
    6.7.1 Impact of Single vs. Multiple Runs .................... 143
    6.7.2 Impact of Boundary Types ................................. 143
  6.8 Results: Impact of ITNs ............................................... 145
  6.9 Results: Impact of Combining LSM and ITNs .................. 150
  6.10 Discussion ............................................................... 153
    6.10.1 Impact of LSM (in Isolation) ......................... 153
    6.10.2 Impact of ITN (in Isolation) ......................... 155
    6.10.3 Impact of LSM and ITNs (in Combination) ........ 157
    6.10.4 Replication of Earlier Models ....................... 158
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.11 Recommendations for Future Modelers</td>
<td>159</td>
</tr>
<tr>
<td>6.12 Summary</td>
<td>161</td>
</tr>
<tr>
<td>CHAPTER 7: INTEGRATING THE ABM WITH A GIS</td>
<td>163</td>
</tr>
<tr>
<td>7.1 Overview</td>
<td>163</td>
</tr>
<tr>
<td>7.2 Use of GIS in Malaria Research</td>
<td>163</td>
</tr>
<tr>
<td>7.3 The Study Area and the GIS-ABM Workflow</td>
<td>166</td>
</tr>
<tr>
<td>7.3.1 The Study Area</td>
<td>167</td>
</tr>
<tr>
<td>7.3.2 The GIS-ABM Workflow</td>
<td>167</td>
</tr>
<tr>
<td>7.4 GIS Data Layers</td>
<td>169</td>
</tr>
<tr>
<td>7.4.1 Processing the Data Layers with GIS</td>
<td>170</td>
</tr>
<tr>
<td>7.4.2 Selecting Aquatic Sites for the Study Area</td>
<td>171</td>
</tr>
<tr>
<td>7.4.3 Scaling Down to Village and Household Level</td>
<td>173</td>
</tr>
<tr>
<td>7.4.4 Conversion of Data Formats</td>
<td>176</td>
</tr>
<tr>
<td>7.4.5 Generating the Study Area for the ABM</td>
<td>176</td>
</tr>
<tr>
<td>7.4.6 Generating the Feature Counts for the ABM</td>
<td>178</td>
</tr>
<tr>
<td>7.5 Creating Scenarios for the Spatial ABM</td>
<td>179</td>
</tr>
<tr>
<td>7.6 Simulations</td>
<td>180</td>
</tr>
<tr>
<td>7.6.1 Assumptions</td>
<td>180</td>
</tr>
<tr>
<td>7.6.2 Spatial Variables</td>
<td>182</td>
</tr>
<tr>
<td>7.7 Results</td>
<td>183</td>
</tr>
<tr>
<td>7.7.1 Mosquito Abundance (Non-Spatial)</td>
<td>183</td>
</tr>
<tr>
<td>7.7.2 Adult Abundance by Location</td>
<td>184</td>
</tr>
<tr>
<td>7.7.3 Cumulative Biomass and Females Oviposited</td>
<td>186</td>
</tr>
<tr>
<td>7.7.4 Cumulative Number of Bloodmeals</td>
<td>188</td>
</tr>
<tr>
<td>7.8 Summary</td>
<td>188</td>
</tr>
<tr>
<td>CHAPTER 8: DATA WAREHOUSE AND DIMENSIONAL MODELING</td>
<td>192</td>
</tr>
<tr>
<td>8.1 Overview</td>
<td>192</td>
</tr>
<tr>
<td>8.2 Data Warehouse (DW)</td>
<td>192</td>
</tr>
<tr>
<td>8.2.1 Dimensional Modeling (DM)</td>
<td>193</td>
</tr>
<tr>
<td>8.2.2 Characteristics and Goals</td>
<td>195</td>
</tr>
<tr>
<td>8.2.3 DW vs. OSS</td>
<td>196</td>
</tr>
<tr>
<td>8.3 Basic Concepts and Vocabulary</td>
<td>197</td>
</tr>
<tr>
<td>8.3.1 Facts and Dimensions</td>
<td>197</td>
</tr>
<tr>
<td>8.3.2 Fact Tables and Dimension Tables</td>
<td>199</td>
</tr>
<tr>
<td>8.3.3 Dimension Hierarchies</td>
<td>200</td>
</tr>
<tr>
<td>8.3.4 Schemas</td>
<td>201</td>
</tr>
<tr>
<td>8.3.5 Data Marts and Cubes</td>
<td>202</td>
</tr>
<tr>
<td>8.3.6 Bus Architecture and Bus Matrix</td>
<td>203</td>
</tr>
<tr>
<td>8.4 Data Warehouse Components</td>
<td>204</td>
</tr>
<tr>
<td>8.4.1 Operational Source Systems (OSSs)</td>
<td>204</td>
</tr>
<tr>
<td>8.4.2 Data Staging Area</td>
<td>204</td>
</tr>
<tr>
<td>8.4.3 Data Presentation Area</td>
<td>205</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>8.4.4 Data Access Tools</td>
<td>207</td>
</tr>
<tr>
<td>8.5 Dimensional Design Process</td>
<td>207</td>
</tr>
<tr>
<td>8.6 Summary</td>
<td>208</td>
</tr>
<tr>
<td>CHAPTER 9: DATA WAREHOUSES IN BIOLOGY</td>
<td>209</td>
</tr>
<tr>
<td>9.1 Overview</td>
<td>209</td>
</tr>
<tr>
<td>9.2 Use of Databases and Data Warehouses in Biology</td>
<td>209</td>
</tr>
<tr>
<td>9.3 Data Integration Challenges in Biology &amp; Bioinformatics</td>
<td>212</td>
</tr>
<tr>
<td>9.4 Heterogeneous Data Storage Systems for Malaria</td>
<td>214</td>
</tr>
<tr>
<td>9.5 Summary</td>
<td>216</td>
</tr>
<tr>
<td>CHAPTER 10: A DATA WAREHOUSE FOR MALARIA DATA</td>
<td>217</td>
</tr>
<tr>
<td>10.1 Overview</td>
<td>217</td>
</tr>
<tr>
<td>10.2 MalOLAP: A Data Warehouse for Malaria</td>
<td>218</td>
</tr>
<tr>
<td>10.2.1 MalOLAP within VECNet</td>
<td>219</td>
</tr>
<tr>
<td>10.2.2 Historical and Predictive Data</td>
<td>220</td>
</tr>
<tr>
<td>10.2.3 User Groups</td>
<td>223</td>
</tr>
<tr>
<td>10.3 Vocabulary</td>
<td>223</td>
</tr>
<tr>
<td>10.4 External Data Storage Systems</td>
<td>226</td>
</tr>
<tr>
<td>10.4.1 Household Surveys</td>
<td>226</td>
</tr>
<tr>
<td>10.4.2 Vector Control Interventions</td>
<td>230</td>
</tr>
<tr>
<td>10.4.3 Rainfall Data from PACRAIN</td>
<td>230</td>
</tr>
<tr>
<td>10.4.4 Georeferenced EIR Data from MARA</td>
<td>231</td>
</tr>
<tr>
<td>10.4.5 Weather Data from GSOD</td>
<td>232</td>
</tr>
<tr>
<td>10.4.6 Insecticides</td>
<td>233</td>
</tr>
<tr>
<td>10.4.7 Lookup Tables</td>
<td>233</td>
</tr>
<tr>
<td>10.5 Design and Implementation</td>
<td>233</td>
</tr>
<tr>
<td>10.5.1 Design Methodology</td>
<td>233</td>
</tr>
<tr>
<td>10.5.2 Data Cubes and OLAP Operations</td>
<td>235</td>
</tr>
<tr>
<td>10.5.3 Operational Source Systems (OSSs)</td>
<td>239</td>
</tr>
<tr>
<td>10.5.4 Data Staging Area</td>
<td>240</td>
</tr>
<tr>
<td>10.5.5 Data Presentation Area</td>
<td>240</td>
</tr>
<tr>
<td>10.5.6 Data Access Tools</td>
<td>240</td>
</tr>
<tr>
<td>10.6 The cubes and Django Frameworks</td>
<td>241</td>
</tr>
<tr>
<td>10.6.1 The cubes Framework</td>
<td>241</td>
</tr>
<tr>
<td>10.6.2 The Django Web Framework</td>
<td>243</td>
</tr>
<tr>
<td>10.7 Results</td>
<td>244</td>
</tr>
<tr>
<td>10.8 Discussion</td>
<td>252</td>
</tr>
<tr>
<td>10.9 Future Work</td>
<td>253</td>
</tr>
<tr>
<td>10.10 Summary</td>
<td>253</td>
</tr>
<tr>
<td>CHAPTER 11: CONCLUSION</td>
<td>254</td>
</tr>
<tr>
<td>11.1 Overview</td>
<td>254</td>
</tr>
<tr>
<td>11.2 An ABM of An. gambiae (Part 1)</td>
<td>254</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>G.4 Post-Simulation Analysis and Visualization for LiNK</td>
<td>330</td>
</tr>
<tr>
<td>G.4.1 Infection Statistics</td>
<td>331</td>
</tr>
<tr>
<td>G.4.2 Roaming Infection Statistics</td>
<td>331</td>
</tr>
<tr>
<td>G.4.3 Birth and Death Statistics</td>
<td>332</td>
</tr>
<tr>
<td>G.4.4 Pathogen Transmission Graphs</td>
<td>332</td>
</tr>
<tr>
<td>G.4.5 Summary Statistics</td>
<td>335</td>
</tr>
<tr>
<td>G.5 P-SAM Performance</td>
<td>338</td>
</tr>
<tr>
<td>G.5.1 Profiling</td>
<td>338</td>
</tr>
<tr>
<td>G.5.2 Code Optimization</td>
<td>340</td>
</tr>
<tr>
<td>G.6 Future Work</td>
<td>340</td>
</tr>
<tr>
<td>G.7 Conclusion</td>
<td>341</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>343</td>
</tr>
</tbody>
</table>
FIGURES

1.1 Dissertation Components .................................................. 2
3.1 Life-Cycle Stages of Mosquito Agents .................................. 38
3.2 Life-Cycle of Malaria .......................................................... 40
3.3 Class Diagram for the ABM ................................................. 50
3.4 Event-Action-List (EAL) Diagram for the ABM ....................... 55
3.5 Order of Major Processing Steps in the ABM ......................... 58
3.6 Dependency Relationships in the Processing Order .................. 59
4.1 V&V Workflow for Section 4.3 .............................................. 66
4.2 V&V Results for Phase 1 ...................................................... 68
4.3 V&V Results for Phase 3 ..................................................... 71
4.4 Simplified Core Model for V&V ........................................... 73
4.5 V&V Workflow for Section 4.4 .............................................. 75
4.6 Docking Abundance in Phase 1 ............................................. 81
4.7 Docking Aquatic Biomass in Phase 1 .................................... 82
4.8 Docking Abundance in Phase 2 ............................................. 82
4.9 Docking Aquatic Biomass in Phase 2 .................................... 83
4.10 Docking Abundance & Biomass in Phase 3 ............................. 85
5.1 Class Diagram for the Spatial ABM ....................................... 91
5.2 Landscape Types .............................................................. 93
5.3 Mosquito Movement during Resource-Seeking ....................... 95
5.4 Screenshot of AnophGUI ..................................................... 97
5.5 Screenshot of VectorLand ................................................... 99
5.6 Model Verification Results .................................................. 101
5.7 Results for Landscape Patterns .......................................... 102
5.8 Results for Resource Size Variation, Part 1 ......................... 103
5.9 Results for Resource Size Variation, Part 2 ......................... 104
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.10 Results for Resource Density Variation</td>
<td>105</td>
</tr>
<tr>
<td>5.11 Results for System Capacity Variation</td>
<td>106</td>
</tr>
<tr>
<td>5.12 Sample Landscape with 50 AHs and 20 BMLs</td>
<td>109</td>
</tr>
<tr>
<td>5.13 Sample Landscape with 100 AHs and 100 BMLs</td>
<td>109</td>
</tr>
<tr>
<td>5.14 Results for Resource Density below Critical Level</td>
<td>110</td>
</tr>
<tr>
<td>5.15 Results for Resource Density above Critical Level</td>
<td>110</td>
</tr>
<tr>
<td>6.1 Flowchart for Household-Level Coverage with Single Chance</td>
<td>123</td>
</tr>
<tr>
<td>6.2 Flowchart for Household-Level Coverage with Multiple Chances</td>
<td>124</td>
</tr>
<tr>
<td>6.3 Flowchart for Household-Level Complete Coverage</td>
<td>126</td>
</tr>
<tr>
<td>6.4 Sample Landscapes for LSM and ITNs</td>
<td>137</td>
</tr>
<tr>
<td>6.5 Impact of LSM with Absorbing Boundary</td>
<td>139</td>
</tr>
<tr>
<td>6.6 Impact of LSM with Non-Absorbing Boundary</td>
<td>140</td>
</tr>
<tr>
<td>6.7 Importance of Performing Multiple Simulation Runs</td>
<td>144</td>
</tr>
<tr>
<td>6.8 Impact of ITNs with Partial Coverage, Multiple Chances</td>
<td>146</td>
</tr>
<tr>
<td>6.9 Impact of ITNs with Complete Coverage</td>
<td>147</td>
</tr>
<tr>
<td>6.10 Percent Reductions in Abundance by ITNs</td>
<td>148</td>
</tr>
<tr>
<td>6.11 Percent Reductions in Abundance by LSM and ITNs</td>
<td>151</td>
</tr>
<tr>
<td>7.1 The Study Area of Asembo, Kenya</td>
<td>168</td>
</tr>
<tr>
<td>7.2 The GIS-ABM Workflow</td>
<td>169</td>
</tr>
<tr>
<td>7.3 Water Sources and Villages Projections for Kenya</td>
<td>171</td>
</tr>
<tr>
<td>7.4 Selecting Subset of Data for Kenya Water Sources</td>
<td>172</td>
</tr>
<tr>
<td>7.5 Selected Village Cluster Area in Asembo, Kenya</td>
<td>173</td>
</tr>
<tr>
<td>7.6 Selected Polygon within the Village Clusters</td>
<td>175</td>
</tr>
<tr>
<td>7.7 Study Area for the Spatial ABM</td>
<td>177</td>
</tr>
<tr>
<td>7.8 Study Area with Selected Data Layers and Grid-Lines</td>
<td>178</td>
</tr>
<tr>
<td>7.9 Mosquito Abundance (Non-Spatial)</td>
<td>183</td>
</tr>
<tr>
<td>7.10 Adult Abundance by Location</td>
<td>185</td>
</tr>
<tr>
<td>7.11 Cumulative Biomass</td>
<td>186</td>
</tr>
<tr>
<td>7.12 Cumulative Number of Females Oviposited</td>
<td>187</td>
</tr>
<tr>
<td>7.13 Cumulative Number of Bloodmeals</td>
<td>189</td>
</tr>
<tr>
<td>10.1 MalOLAP within the Context of VECNet CI</td>
<td>221</td>
</tr>
<tr>
<td>10.2 MalOLAP Ontology</td>
<td>224</td>
</tr>
<tr>
<td>10.3 MalOLAP Constellation Schema for Data Marts</td>
<td>227</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>E.9 Dimensions for GSOD Weather</td>
<td>305</td>
</tr>
<tr>
<td>E.10 Fact Table GSOD Weather</td>
<td>306</td>
</tr>
<tr>
<td>E.11 Dimensions for Insecticides</td>
<td>307</td>
</tr>
<tr>
<td>E.12 Fact Table Insecticides</td>
<td>307</td>
</tr>
<tr>
<td>E.13 Lookup Table Species</td>
<td>308</td>
</tr>
<tr>
<td>E.14 Lookup Table Entomological Endpoints</td>
<td>308</td>
</tr>
<tr>
<td>F.1 Greenstone Librarian Interface</td>
<td>311</td>
</tr>
<tr>
<td>F.2 Format Tab in the Greenstone Librarian Interface</td>
<td>312</td>
</tr>
<tr>
<td>F.3 Homepage for the Prototype Digital Library Portal</td>
<td>313</td>
</tr>
<tr>
<td>F.4 Search with Keyword Malaria</td>
<td>314</td>
</tr>
<tr>
<td>F.5 Sample Document from VECNet DL Collection</td>
<td>315</td>
</tr>
<tr>
<td>F.6 Search with Keyword Garki</td>
<td>316</td>
</tr>
<tr>
<td>F.7 Search with Keyword WHOPES</td>
<td>317</td>
</tr>
<tr>
<td>F.8 Sample Document from WHOPES DL Collection</td>
<td>318</td>
</tr>
<tr>
<td>G.1 P-SAM Architecture</td>
<td>327</td>
</tr>
<tr>
<td>G.2 P-SAM Infection Statistics Tab</td>
<td>332</td>
</tr>
<tr>
<td>G.3 P-SAM Roaming Infection Statistics Tab</td>
<td>333</td>
</tr>
<tr>
<td>G.4 P-SAM Birth and Death Statistics Tab</td>
<td>334</td>
</tr>
<tr>
<td>G.5 Pathogen Transmission Graph</td>
<td>336</td>
</tr>
<tr>
<td>G.6 P-SAM Summary Statistics Tab</td>
<td>337</td>
</tr>
<tr>
<td>G.7 P-SAM Performance</td>
<td>339</td>
</tr>
</tbody>
</table>
TABLES

2.1 FEATURE COMPARISONS OF MALARIA MODELS . . . . . . . . . . 28
3.1 STAGE TRANSITION TIMES FOR MOSQUITO AGENTS . . . . . . . . 37
3.2 DEFAULT MODEL PARAMETERS . . . . . . . . . . . . . . . . . . . . 53
4.1 STAGE TRANSITION TIMES (V&V) . . . . . . . . . . . . . . . . . . . 74
4.2 V&V ISSUES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
6.1 POPULATION PROFILES FOR PARTIAL COVERAGE . . . . . . . . 127
6.2 POPULATION PROFILES FOR COMPLETE COVERAGE . . . . . . . 128
6.3 PARAMETER SPACE FOR ITNS . . . . . . . . . . . . . . . . . . . . 135
6.4 PARAMETER SPACE FOR LSM AND ITNS . . . . . . . . . . . . . . 136
6.5 PR IN ABUNDANCE WITH LSM . . . . . . . . . . . . . . . . . . . . 141
7.1 GIS FEATURE COUNTS . . . . . . . . . . . . . . . . . . . . . . . . . 179
7.2 HYPOTHETICAL SCENARIO 21K . . . . . . . . . . . . . . . . . . . . 181
7.3 HYPOTHETICAL SCENARIO 150K . . . . . . . . . . . . . . . . . . . 181
B.1 ENTROPY AND ENTHALPY OF ACTIVATION . . . . . . . . . . . 279
B.2 ENZYMES, REACTION AND RATE CONSTANT . . . . . . . . . . . 280
G.1 PERL EXTENSION MODULES USED IN P-SAM . . . . . . . . . . . 328
LIST OF ABBREVIATIONS (IN ORDER OF FIRST APPEARANCE)

ABM  Agent-Based Model, Agent-Based Modeling
V&V  Verification & Validation
GIS  Geographic Information System
DW  Data Warehouse

VECNet  Vector Ecology and Control Network
CI  Cyberinfrastructure
BI  Business Intelligence
DSS  Decision Support System

OLAP  Online Analytical Processing
DDT  Dichloro-Diphenyl-Trichloroethane
WHO  World Health Organization
MDA  Mass Drug Administration
ITN  Insecticide-Treated Net
IRS  Indoor Residual Spraying
IVM  Integrated Vector Management
LSM  Larval Source Management
LLIN  Long-Lasting Insecticide-treated Net

ACT  Artemisinin-Combination Therapy
MSAT  Mass Screening And Treatment
EIR  Entomological Inoculation Rate

ASMR  Age-Specific Mortality Rate
CC  Carrying Capacity

EAL  Event-Action-List

DAG  Directed Acyclic Graph

DGPS  Differential Global Positioning System

CCC  Combined Carrying Capacity

AH  Aquatic Habitat

ATT  Average Travel Time

VA  Vector Abundance

BML  Bloodmeal Location

GUI  Graphical User Interface

ABMS  Agent-Based Modeling and Simulation

RBM  Roll Back Malaria

PR  Percent Reduction

DMR  Daily Mortality Rate

ESD  Effective Shortest Distance

API  Application Programming Interface

GPS  Global Positioning System

3NF  Third Normal Form

POS  Point of Sale

ER  Entity Relationship (Model)

OSS  Operational Source System

EAI  Enterprise Application Integration

ETL  Extract-Transformation-Load

SQL  Structured Query Language

RDBMS  Relational Database Management System

IT  Information Technology

WMR  World Malaria Report
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACRAIN</td>
<td>Pacific Rainfall Database</td>
</tr>
<tr>
<td>GSOD</td>
<td>Global Surface Summary of Day</td>
</tr>
<tr>
<td>NCDC</td>
<td>National Climatic Data Center</td>
</tr>
<tr>
<td>DVS</td>
<td>Dominant Vector Species</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Gregory R. Madey, for his guidance, patience, encouragement, and support. Special thanks go to Dr. Frank H. Collins for guidance and funding of the conducted research of this dissertation. Thanks to Dr. Neil F. Lobo for his feedback on the modeling process.

Thanks to my other committee members: Dr. Patrick J. Flynn and Dr. Jessica J. Hellmann, for their valuable comments and contributions on this dissertation.

Thanks also to my colleagues, Dr. Ryan C. Kennedy, for sharing his expertise and for proofreading, Ying Zhou, for sharing her initial model, and Gregory J. Davis, for the V&V work. Special thanks to my brother-in-law, Mohammad Shahed Sorower from Oregon State University, for his valuable comments and a very helpful proofreading. Thanks to my colleagues in the VECNet CI [241] team at CRC [36]. Thanks to our administrative assistant for graduate studies, Ms. Joyce Yeats, for her invaluable assistance over the years.

Finally, I would like to thank my family, particularly my parents, Engineer S. M. Golam Mostofa and Professor Parvin Akhter Jahan, and my sister, Mafruhatul Jannat, for their continuous encouragement and support. Last but not least, sincere gratitude to my wonderful wife, Rumana Reaz Arifin, for her tremendous support and care over the past few years. This work would not have been possible without their everlasting inspiration and motivation.
CHAPTER 1
INTRODUCTION

1.1 Overview

The work presented in this dissertation consists of two related parts. Part 1, which consists of Chapters 2–7, concerns agent-based modeling of malaria, a deadly vector-borne disease. We describe the detailed design, implementation, and verification & validation (V&V) issues for agent-based models (ABMs)\(^1\) of the dominant malaria vector *Anopheles gambiae* (*An. gambiae* for short). We discuss a spatial extension of the ABM, and its evaluation with vector control interventions. We also demonstrate an example application of the ABM by integrating data from a geographic information system (GIS). Part 2, which consists of Chapters 8–10, involves a multi-dimensional data warehouse (DW) for malaria-related data, that we design and implement as part of the Vector Ecology and Control Network (VECNet) Cyberinfrastructure (CI) Project [241]. It lies within the realms of Business Intelligence (BI) and Decision Support Systems (DSSs). Each of these parts, having shared implications for global health, may be categorized under the computational biology realm, connected to the interdisciplinary fields of bioinformatics and epidemiology, as shown in Figure 1.1.

The following sections provide a brief introduction to both parts. We conclude

\(^1\)Several language-specific implementations of the biological core model have been developed by individual researchers within our research group over the recent years. For this dissertation, the plural term ‘ABMs’ refers to these implementations (which are compared solely for V&V purposes in Chapter 4). The singular term ‘ABM’, where applicable, refers to the particular implementation developed by the author.
Figure 1.1. Dissertation Components. This dissertation consists of two related parts. **Part 1** (Chapters 2–7) concerns agent-based modeling of malaria. Chapters in Part 1 commonly focus on the dominant mosquito vector *An. gambiae* for malaria epidemiology, which, in turn, fall under the realm of bioinformatics. **Part 2** (Chapters 8–10) describes a multi-dimensional data warehouse for malaria-related data, as an integral part of the VECNet CI Project [241]. Chapters that describe our contribution for the different categories are listed under the category name (where applicable). Note that some chapters may overlap into multiple categories. The dashed lines indicate connections which are more conceptual in nature, while the solid lines indicate connections which are more implementation- and/or application-oriented. Both parts of the dissertation share global/public health implications.
this chapter by listing our specific contributions, and by providing a roadmap for the remainder of this dissertation.

1.2 Agent-Based Modeling of Malaria

Agent-based modeling (ABM) is a powerful simulation modeling technique that has applications to diverse real-world problems. An ABM consists of a collection of autonomous decision-making entities called agents [27]. By employing a set of rules, agents can individually assess the environment, and may make decisions. Appropriate behaviors for the system may then be executed by the agents.

Malaria, as a deadly disease, contributes substantially to the global burden of diseases. The population of sub-Saharan Africa, which accounts for 80% of the globally-estimated 2 million annual deaths [174], experiences the highest burden of the disease, with a child dying every 30 seconds [251]. As highland malaria re-emerged in several African countries in the recent decades [191], the control of malaria represents one of the greatest global public health challenges of the 21st century.

Caused by protozoan parasites of the genus Plasmodium, malaria is one of the oldest and deadliest infectious diseases in humans. According to the World Health Organization, half of the world’s population (about 3.3 billion) are at risk of malaria [251]. In 2008, it caused 247 million cases and nearly one million deaths, mostly among African children [144]. Human malaria is transmitted only by females of the genus Anopheles, which are regarded as the primary vector for transmission. An. gambiae, which refers to a complex in the genus Anopheles, contains the most important malaria vectors in sub-Saharan Africa, and the most efficient malaria vectors in the world. We develop an ABM that simulates the life cycle of An. gambiae, by tracking attributes relevant to the vector population dynamics for each individual mosquito.
1.3 A Data Warehouse for Malaria-Related Data

Biological data require specialized expertise for their collection, analysis and interpretation [259]. The problem of integrating heterogeneous biological databases and data storage systems arises in various scenarios, primarily due to the differences in data definition, type, and content [32, 146, 128]. In many of these cases, information from autonomous (i.e., independently developed and operated) data sources need to be combined under a common platform for storage, access, and analysis.

For malaria, due to the complex nature of the disease, and its numerous aspects on which data are stored, there is a growing need to integrate these heterogeneous data sources. Data warehouses can be a viable solution to traditional databases. We describe a multi-dimensional, online analytical processing (OLAP) data warehouse (DW), designed specifically to store, access, and analyze malaria-related data scattered in heterogeneous forms and contents across many data storage systems. The DW is designed to fit and operate within an analytical framework named as the Vector Ecology and Control Network (VECNet).

1.4 Contributions

This dissertation makes the following contributions:

- design and implement an ABM of *An. gambiae* from a biological core model
- perform the V&V processes (of various ABMs developed within our research group)
- model a spatial extension of the ABM
- develop a landscape generator tool (*VectorLand*) for the ABM
- examine the impact of several vector control interventions
- demonstrate an example application of the ABM by integrating it with a geographic information system (GIS)
- design and implement a multi-dimensional data warehouse for malaria-related data
1.5 Organization

The remainder of this dissertation is organized as follows. In Part 1, Chapter 2 provides a general background about ABMs, malaria, and the applicability of ABMs in malaria research, and describes some early and recent malaria models. Chapter 3 thoroughly describes the biological core model of An. gambiae, and the ABM we develop from the core model. Chapter 4 describes the V&V processes that we perform on several language-specific implementations of the core model. Chapter 5 describes a spatial extension of the ABM aided by a custom-built landscape generator tool. Chapter 6 presents results of examining the impact of vector control interventions (larval source management and insecticide-treated nets), applied both in isolation and in combination. Chapter 7 shows an example application of the ABM that integrates a GIS.

In Part 2, Chapter 8 presents a general background of dimensional modeling and data warehouses. Chapter 9 discusses, in general, the use of data warehouses in biology, including various heterogeneous data storage systems for malaria-related data. Chapter 10 describes a multi-dimensional data warehouse that we develop for malaria-related data.

Chapter 11 summarizes the contributions of this dissertation. At the end, we conclude this document with supplementary information: Appendix A depicts the detailed flowchart for the Java implementation (described in Chapter 3) of the ABM; Appendix B describes the theoretical background on the temperature-regulated rules and equations used in the models; Appendices C, D, and E present some additional materials for Chapters 6, 7, and 10 respectively; Appendix F describes a prototype digital library portal for the VECNet project [241]; and Appendix G describes a software module, called P-SAM (Post-Simulation Analysis Module), that we develop to analyze and visualize the post-simulation outputs of ABMs.
CHAPTER 2

MALARIA MODELING: A BACKGROUND

2.1 Overview

This chapter provides general background about Agent-Based Modeling (ABM), malaria, and the applicability of ABMs in malaria modeling. We also describe early and recent malaria models, and broadly categorize these as mathematical and agent-based. Section 2.2 introduces the major components of an ABM, and mentions some differences between ABMs and traditional mathematical modeling techniques. The major advantages and disadvantages of ABMs are discussed next. In Section 2.3 we provide a brief history of malaria. In Section 2.4 we discuss, in general, why ABMs are suitable for malaria modeling. Finally, in Section 2.5 we present a literature review of relevant malaria models that fall under both categories (mathematical and agent-based). The spatial dimension of malaria models is also highlighted.

2.2 Agent-Based Modeling (ABM)

Agent-Based Modeling (ABM) is a powerful simulation modeling technique that has applications to diverse real-world problems. An ABM models a system as a collection of autonomous decision-making entities called agents [27]. Employing a set of rules, agents individually assess the environment and make decisions. Appropriate behaviors for the system may then be executed by the agents. An ABM can exhibit complex behavior patterns and provide valuable information about the dynamics of the real-world system that it emulates [27].
2.2.1 Agents, Environment, and Rules

According to Epstein and Axtell [62], the three major components of an ABM are agents, environment, and rules. At the simplest level, an ABM consists of a system of agents and the relationships between them. Agents are autonomous decision-making entities that can have internal states to store relevant attributes and data. These internal states can be represented by discrete or continuous variables. Some states (e.g., sex, id, vision) may remain fixed throughout the agent’s lifespan in the simulation, while others (e.g., health, wealth, individual preferences) may change through interaction with other agents and/or the environment. Once an agent’s state variables are defined, its behavior can be represented as a state-determined automata or finite state machine, in which a state transition occurs whenever the agent interacts with another agent. Thus, all the interactions and changes of states are regulated by rules of behavior for the agents and the environment. Examples of agent behaviors include production or consumption of resources, communication to other agents and to the environment, etc. In addition, agents may be capable of evolving, allowing unanticipated behaviors to emerge. Sophisticated ABMs sometimes incorporate neural networks, evolutionary algorithms, or other learning techniques to allow realistic learning and adaptation.

Agents’ life in an ABM evolves in an artificial, well-defined environment. The environment can be thought of as a medium on which agents operate and with which they interact. For example, a landscape can be naturally modeled as a lattice environment of resource-bearing sites for the agents. More abstract examples of environment may include a communication network whose topology and connectivity may change over time.

Rules couple the agents with their environment. Both the agents and the environment can have their own set of rules, which can be operated between agent-environment level, environment-environment level or agent-agent level. For example,
an agent-environment level rule for movement may instruct an agent to look around as far as it can, find the site richest in resource, go there and grab the resource. An environment-environment level rule for site-resource update may treat the rate of resource change at a particular site as a function of resource levels at neighboring sites, where the neighborhood is pre-defined according to some criteria. Finally, agent-agent level rules may include how a male agent mates with a female agent, how one type of agents (e.g., predator) interact with another type (e.g., prey), etc.

2.2.2 Differences between ABMs and Mathematical Models

ABMs rely on the power of computers to explore system dynamics that is otherwise out of the reach of traditional mathematical models. Traditional mathematical models usually include differential equation-based models. In the past, majority of theories in ecology evolved from simple differential equations in which a single variable represents population densities. Solutions of these are analyzed mathematically, and may be compared to population abundance estimates from field or laboratory observations.

Although these models greatly influenced on theories in ecology, their aggregated form is particularly difficult to relate to observational biology. Their application to complex natural systems with spatially and temporally varying environmental factors leads to models which are not analytically tractable, and must be investigated numerically [84].

According to Gros [84], to design these mathematical models, one of the first factors to be determined is whether the key features of the problem under consideration require a discrete or continuous formulation. For example, in population biology, a discrete model may use matrices or systems of non-linear difference equations in order to compare populations with discrete life stages, whereas a continuous model may use systems of differential equations with overlapping generations. A second di-
chotomy concerns whether stochastic factors should be included or ignored. Another major issue lies in dealing with spatial aspects: whether to ignore them by assuming homogeneous mixing, or include them (e.g., through discrete patches, discrete lattices, partial differential equations, etc.).

However, to describe the model, all of the above approaches involve state variables, which represent a form of aggregation. As a result, some factors of the real-world problem which affect individual actions can only be dealt with in a highly aggregated manner in traditional mathematical models. ABMs offer the powerful capability to analyze these aggregated variables and to observe them as a function of the actions and interactions of the agents which make up the aggregation. Thus, as well as describing the basic processes which control the actions of agents, ABMs can also aggregate these up to determine the resultant macro-descriptors which arise at higher levels of model organization.

2.2.3 Advantages of ABMs

ABMs offer three primary advantages over other modeling techniques: capturing emergent phenomena, providing more natural description of a system, and flexibility. These are briefly described below.

2.2.3.1 Capturing Emergent Phenomena

ABMs can capture emergent phenomena resulting from the interactions of individual entities. Because of the interactions between the parts, the whole is more than the sum of its parts. So, an emergent phenomenon can have properties that are decoupled from the properties of the part. However, the characteristics of emergent phenomena sometimes make them difficult to understand and predict because they can be counter-intuitive. For example, a traffic jam resulting from the interactions between individual vehicle drivers, may be moving in the direction opposite that of
the vehicles that cause it. ABMs capture emergence from bottom up, and simulate
the behavior of the system’s constituent units (the agents) and their interactions.
Thus, they offer a canonical approach to model emergent phenomena. They can
show coherent group behavior emerging from simple individual rules, and how small
changes in those rules can have a dramatic impact on the group behavior, treating
the group’s collective behavior as an emergent phenomenon.

This is particularly useful in the following situations [27]:

- Individual behavior is nonlinear and can be characterized by thresholds, if-then
  rules, or nonlinear coupling: Describing discontinuity in individual behavior is
difficult with differential equations.

- Individual behavior exhibits memory, path-dependence, or temporal correlations,
  including learning and adaptation.

- Agent interactions are heterogeneous and can generate network effects: Aggre-
gate flow equations usually assume global homogeneous mixing, but the topology
of the interaction network can lead to significant deviations from predicted
aggregate behavior.

- Averages are not suitable: Aggregate differential equations tend to smooth out
  fluctuations. Under certain conditions, fluctuations can be amplified, making
  the system linearly stable but unstable to larger perturbations.

Thus, ABMs provide a new way of approaching biological phenomena.

2.2.3.2 Providing Natural Description of a System

In many cases, ABMs can offer the most natural way for describing and simulating
a system composed of behavioral entities, and makes the model seem closer to reality.
For example, movement of shoppers in a supermarket can be more naturally described
by an ABM than by density equations that govern the dynamics of shoppers density.
Since the density equations result from the behavior of shoppers, the ABM approach
will also enable the user to study aggregate properties.

According to [27], ABMs can model a complex system more naturally in the
following situations:
• Individual behavior is complex, and cannot be clearly defined through aggregate transition rates. The complexity of differential equations increases exponentially as the complexity of behavior increases, and describing complex individual behavior with equations becomes intractable.

• Activities, rather than processes, are a more natural way of describing the system.

• Verification and validation (V&V) as well as calibration of the model through expert judgment is crucial. ABMs are often the most appropriate way of describing what is actually happening in the real world, enabling the experts to easily connect to the model.

• Stochasticity applies to the agents’ behavior. With ABMs, sources of randomness are applied to the appropriate places as opposed to a noise term added arbitrarily to an aggregate equation.

2.2.3.3 Flexibility

The flexibility of ABMs can be observed along multiple dimensions. It is straightforward to scale up the number of agents in an agent-based model. ABMs also provides a natural framework for tuning the agents complexity (e.g., agent behavior, degree of rationality, ability to learn and evolve, and rules of interactions). Another dimension of flexibility is the ability to change levels of description and aggregation. It offers flexible ways to play with aggregate agents, subgroups of agents, and single agents, with different levels of description coexisting in a given model. Thus, it can be used when the appropriate level of description or complexity is not known ahead of time.

2.2.4 Disadvantages of ABMs

There is no single method or tool without having flaws, or being the best to use in all cases. ABMs also have drawbacks, some of which are listed below:

• Existence of closed-form solution: A closed-form solution (e.g., finding the roots of a polynomial) can be expressed analytically in terms of a bounded number of certain functions. Usually, it refers to an equation-based solution that solves a
given problem in terms of functions and mathematical operations from a given generally accepted set \[44\]. If the model under consideration is too simple, and can be solved in closed-form, then the closed-form solution is preferable over an ABM \[13\].

- **Speed**: ABMs are usually slower to run than other modeling techniques.
- **Cost**: Due to the finer granularity of information, relatively high costs (in both time and effort) are involved, compared to equation-based models \[171\]. Also, ABMs tend to require more data than many other approaches \[27\].
- **Errors in Coding**: A complex ABM, requiring many lines of code, creates the possibility for more coding errors, and/or programmatic misinterpretations, to creep in.
- **Result Analysis**: As a consequence of the finer granularity and massive data requirement, it is often difficult to detect whether the results produced by an ABM are the cause of a programming error or a groundbreaking insight \[171\].
- **Data Analysis**: The search space is usually much bigger for parameter analysis, calibration etc.
- **Verification & Validation (V&V)**: V&V activities can sometimes be very resource demanding \[171\].
- **Programming Language Requirement**: Development of an ABM demands knowledge of programming languages. Also, once developed, it may still be difficult to comprehend by many people.
- **Re-usability**: Usually, an ABM is often customized, and is specific to the modeled context, thus having limited scope for re-use \[171\].

2.3 A Brief History of Malaria

Despite tremendous efforts to control malaria, it remains a public health problem in more than 90 countries inhabited by 40% of the global population. More than 90% of all malaria cases occur in sub-Saharan Africa, where in some areas over 70% of residents are continuously infected with the most deadly form of the parasite, \textit{Plasmodium falciparum} (\textit{P. falciparum} for short). Children that survive the early years of life develop varying levels of natural immunity, but this does not protect them from repeated infections and illnesses throughout life \[20\]. Although numerous malaria
models have been developed during the recent past, malaria remains uncontrolled and is increasing in many areas, as are vector and parasite resistance to insecticides and drugs [51].

This section briefly describes the history of malaria as an ancient disease, its naming, and some early malaria models.

2.3.1 Naming of Malaria

The term malaria originated from the Italian mala (meaning “bad”) and aria (meaning “air”). Since ancient times, malaria, also known as jungle fever, marsh fever, paludal fever, or swamp fever, was linked with poisonous vapors of swamps or stagnant water on the ground. This probable relationship gave the two most frequently used names to the disease mal’aria (later shortened to one word malaria), and paludisme [99].

2.3.2 Malaria as an Ancient Disease

Malaria[1] or a disease resembling malaria, has been noted for more than 4,000 years, and thus is one of the oldest and deadliest documented infectious diseases in humans. The symptoms of malaria were described in ancient Chinese medical writings (around 2700 BC).

Advent of quinine as an antimalarial drug occurred in the early 17th century. Following their arrival in the new world, the Spanish learned of a medicine used for the treatment of fevers from indigenous Indian tribes. It was prepared from the bark of trees known as the Peruvian bark. In 1880, Charles Louis Alphonse Laveran, a French army surgeon stationed in Algeria, was the first to discover the malaria parasites in the blood of a patient suffering from malaria.

[1]The summary presented in this section is adopted from [230].
On August 1897, Ronald Ross, a British officer in the Indian Medical Service, was the first to demonstrate that mosquitoes transmit malaria parasites. He showed that malaria parasites could be transmitted from infected patients to mosquitoes. In further work with bird malaria, Ross showed that mosquitoes could transmit malaria parasites from bird to bird.

In 1898, a team of Italian investigators discovered the transmission of the human malaria parasites *Plasmodium* by collecting *Anopheles* mosquitoes and feeding them on malarial patients. They demonstrated the complete sporogonic cycle (the time interval during which the parasite develops in the mosquito) of *P. falciparum*, *P. vivax*, and *P. malariae*. In 1934, *chloroquine* was discovered by a German, Hans Andersag, and was finally recognized and established as an effective and safe antimalarial drug in 1946 by British and U.S. scientists. In 1939, Paul Müller in Switzerland discovered the insecticidal property of Dichloro-Diphenyl-Trichloroethane (DDT). DDT was used for malaria control at the end of WWII after it had proven effective against malaria-carrying mosquitoes by British, Italian, and American scientists.

2.4 Applicability of ABMs in Malaria Modeling

ABMs has increasingly become a useful tool for representing complex systems as it allows the construction of model frameworks to include substantial details and reality. In modeling malaria (and epidemiology in general), this is particularly useful for a variety of reasons [62]:

**Treatment of space** Treatment of space differs fundamentally from that in traditional mathematical models. Usually, ordinary differential equation models have no spatial component, and all agents interact in time but not in space. Some partial differential equation models include a physical space *x*, but the state variables that represent agent populations are continuous in *x*. In ABMs, agents are usually distributed over a two-dimensional space lattice. As we show in Chapter 5, our ABM provides a flexible framework to incorporate a spatial extension that can capture the spatial properties of both the mosquito agents and the environments (landscapes).
Compartmentalization In mathematical epidemiology, the society being modeled is often divided into homogeneous sub-populations called compartments. Each compartment represents a sub-population in which members are not distinguishable from each other. Homogeneity is assumed within each sub-population and/or species. For example, to model malaria infectivity, McKenzie et al. [159, 158] divided the host and vector populations into infected, infectious and susceptible compartments. In these models, as the compartments are homogeneous, and individuals within each compartment can have no variation. By contrast, in ABMs there is no such aggregation. ABMs allow substantial variation within a compartment by allowing agents to possess and display heterogeneous behavior. As a result, important phenomena such as immunological memory, persistence of genetic traits etc. can emerge naturally within the system dynamics.

Adaptation In ABM, individual traits of an agent can adapt as a result of interaction with other agents and components of the environment. In evolutionary time (‘elapsed’ time on computers), selection pressures operate to alter the distribution of traits in populations. This can be especially helpful for malaria modeling, for example, to model insecticide resistance and genetic variations within the mosquito population.

To model certain characteristics of a system, sometimes it may be advantageous, and even necessary, to use ABMs. Some of these characteristics [27], and how they relate to our malaria ABM, are discussed below:

- Interactions between the agents are complex, nonlinear, discontinuous, or discrete: for example, when the behavior of an agent can be altered dramatically, even discontinuously, by other agents; in our malaria ABM, agents interact in a complex, nonlinear, and discrete fashion.

- Space is crucial and the agents’ positions are not fixed: in our malaria ABM, space is classified as the mosquito world, aquatic habitat, etc., for the mosquito agents; eventually, when humans are added as agents, more diverse types of spaces may be incorporated.

- The population is heterogeneous, and each individual is (potentially) different: the mosquito population can be heterogeneous in terms of species, traits, development time, etc.

- The topology of the interactions is heterogeneous and complex.

- The agents exhibit complex behavior, including learning and adaptation: as biological complexities are added to the core model, it is iteratively refined; thus, the mosquito agents exhibit increasingly complex behavior; the possibilities of learning and adaptation by agents also exist.
2.5 Malaria Models: A Literature Review

In this section, we describe some early and recent malaria models. We broadly categorize these models as \textit{mathematical} and \textit{agent-based}, and briefly introduce some of these models. Then, we also classify the models into non-spatial and spatial classes, based on the representation of the \textit{spatial} dimension in the models. Note that the criteria for this classification do not include other model-specific features (e.g., use of geographic information systems), and hence several models falling under different categories may share those features.

Malaria modeling, both mathematical and agent-based, can play important roles in quantifying the effects of malaria-control interventions and in answering other interesting research questions. Models can play key roles in selecting appropriate combinations of interventions to interrupt transmission and in setting response timelines and expectations of impact, and can help to elucidate whether different interventions are likely to be synergistic, and when they can be deployed to best effect. Models are thus an indispensable tool for thinking carefully and quantitatively about the dynamics of malaria control and elimination. Although computer-based simulation studies are not a substitute for reality, they do provide a highly refined and structured way of synthesizing information and testing ideas. In particular, they provide a useful tool for testing how differences in transmission can lead to different results when the same interventions are applied in two different populations.

Starting with the pioneering models of Ross [198] and Macdonald [140], numerous models of malaria have been developed during the past century, and applied to potential interventions. In the history of malaria research, several examples show how vector control measures like indoor spraying of insecticides, larval control, and environmental management have helped control or eradicate malaria in areas of marginal or unstable transmission. However, a valuable lesson gained in that is the strategies to effectively control malaria in one ecological setting may not be appropriate
in another [20]. It is difficult to predict where successful approaches will emerge, and there is every indication that integrated approaches will be needed for effective and sustainable control. Thus, simply having a new antimalarial drug, an effective malaria vaccine, or a new way to kill mosquitoes is still a long way and many years from the achievement of effective malaria control.

The complex interactions that occur between malaria parasites and vector species of mosquitoes have been studied by many models. For example, Beier [20] discusses how the development of appropriate strategies to control malaria by blocking parasite transmission in nature depends both on an understanding of malaria parasite-vector interactions and how these interactions serve to regulate the dynamics of transmission by populations of vector mosquitoes, and reviews malaria parasite development in mosquitoes from the perspective of transmission dynamics. Highlighting drug development, vaccine development, and vector control as some of the areas in which recent progress in malaria control has occurred, he lists four categories relating to the challenges of vector control research:

- vector control to reduce transmission,
- antimalarial drugs that kill gametocyte-stage parasites,
- genetic approaches to reduce transmission, and
- transmission-blocking immunological approaches.

2.5.1 Mathematical Models of Malaria

In this section, we provide a brief description of mathematical models of malaria, which have been used to provide an explicit framework for understanding malaria transmission dynamics in human population for over 100 years [148]. A thorough literature survey, describing the prominent mathematical models, their underlying features, and their specific contributions in the understanding of spread and transmission of malaria, can be found in [148].
Mathematical modeling of malaria transmission dates back to the early models of Ronald Ross, who developed the first mathematical model for understanding malaria transmission [197]. In the 1950’s, Macdonald developed a reformulated model that identified mosquito vector longevity as the single most important variable in the force of transmission, and combined Ross’s more famous differential equation model with epidemiological and entomological field data [140, 141]. Since then, the Ross-Macdonald theory has played a central role in the development of research on mosquito-borne pathogen transmission and the development of strategies for mosquito-borne disease prevention [219].

Molineaux and Gramiccia [164] present a well-known malaria model developed during the Garki project for the planning and management of malaria control. The study was carried out in the Garki district of northern Nigeria in 1969-76 by a joint research team of the World Health Organization (WHO) and the government of Nigeria. With a focus on the epidemiology and control of malaria in the African savanna, the model comprehensively studied the effects of a residual spraying campaign and mass drug administration (MDA) on malaria transmission. The specific objectives of the project includes an epidemiological study of malaria in the lowland savanna (concentrating on the measurements of entomological, parasitological and seroimmunological variables and on their relationships), to measure the effects of house-spraying with propoxur (to control the vectors An. gambiae and An. arabiensis) alone or in combination with MDA, and to construct and test a mathematical model of the transmission of malaria to compare various control strategies in terms of their expected effects.

Martens et al. [152, 153] describe an integrated linked-system mathematical model to study the effects of projected changes in temperature and precipitation on mosquito and parasite characteristics and their potential impact on malaria risk. Using the Integrated Model to Assess the Greenhouse Effect (IMAGE, 200), they asses the sensi-
tivity of the biological activity and geographic distribution of the malaria mosquitoes and parasite to climatic influences (e.g., temperature, precipitation, etc.). IMAGE is a climate assessment model designed to simulate the cause-effect chain with respect to climate change, and consists of a number of independent, interlinked and integrated sub-models, each of which represents a separate component of the climate system (e.g., a world energy/economy model, land-use change model, atmospheric chemistry model, etc.). They conclude that the simulated changes in malaria risk must be interpreted on the basis of local environmental conditions, the effects of socioeconomic developments, and malaria control programs or capabilities, and increased risk of malaria due to climate change may seriously affect human health in the next century. The study shows that the transmission potential of malaria as a vector-borne diseases is very sensitive to climate changes on the periphery of the present endemic areas and at higher altitudes within these areas, and the health impact will be most pronounced in populations living in the less economically developed temperate areas in which endemicity is low or absent.

In recent years, climate models of malaria transmission have also been developed to improve our understanding of the likely impact of climate change on malaria transmission. Martens et al. [151] describe a malaria model in which the assessment of the impact of climate change on potential malaria transmission is made using the MIASMA (Modelling framework for the health Impact ASsessment of Man-induced Atmospheric changes) model. Main features of the model include continental-scale estimates regarding the distribution of 18 main malaria vectors, species-specific relationships between temperature and transmission dynamics, and a more realistic approach regarding malaria endemicity to explore changes in populations at various degrees of malaria risk (e.g., risk of epidemics vs. year-round transmission). In this model, global estimates of the potential impact of climate change on malaria transmission are calculated based on future climate scenarios produced by the HadCM2
and HadCM3 global climate models [102]. The model shows that at the borders of malaria transmission, the changes in average length of the transmission season may be important. Also, the simulations show that in most current high endemic regions, an increase in seasonal transmission occurs at the expense of year-round transmission. Although strongly varying between the climate change scenarios, this implies that the malaria situation in these regions moves towards unstable malaria. They conclude that estimates of future populations at risk of malaria differ significantly between regions and between climate scenarios. The results, though not to be treated as predictions of the future, show trajectories of possible changes in malaria risk with the given assumptions.

Janssen and Martens [107] describe a malaria-epidemiological dynamics model that simulates the adaptation of mosquitoes and parasites to available pesticides and drugs. To address the evolutionary character of the development of resistance, they couple genetic algorithms with the model to simulate the evolving processes within the mosquito and parasite populations. Results suggest that adequate use of insecticides and drugs may reduce malaria occurrence in low endemicity areas, with increased efforts in the event of a climate change. In high endemicity areas, the use of insecticides and drugs may lead to an increase in malaria incidence due to enhanced resistance development in the mosquito and parasite populations.

Craig et al. [47] describe a simple numerical, fuzzy logic-based modeling approach of the distribution of stable malaria transmission in sub-Saharan Africa, based upon biological constraints of climatic suitability on parasite and vector development. Using geographic information systems (GIS) software and large global data sets including climate, population, satellite imagery and topography, the model defines geographic regions as perennial (where conditions are always suitable for transmission), seasonal (where conditions become suitable for a short season every year), epidemic (where long-term variation in climate renders conditions suitable for transmission on
an irregular basis with a potential of epidemic malaria), and malaria-free. To examine the pattern of mean climate as it relates to different epidemiological settings, the model extracts and uses monthly rainfall and temperature data from the climate data surfaces for 20 different sites. The model compares well with contemporary field data and historical ‘expert opinion’ maps, excepting small-scale ecological anomalies. It provides a numerical basis for further refinement and prediction of the impact of climate change on transmission. Together with population, morbidity and mortality data, the model provides a fundamental tool for strategic control of malaria.

Hay et al. [95, 96] highlight the sensitivity of malaria epidemics to climate change in the highlands of East Africa. They use a regression approach to analyze the potential effects of climate change on highland malaria. Using a first-order autoregressive model and meteorological data obtained from a global gridded data set of monthly terrestrial surface climate for the 1901–95 period, they investigate long-term meteorological trends in four high-altitude sites in East Africa, where increases in malaria have been reported in the past two decades. They show that temperature, rainfall, vapor pressure and the number of months suitable for *P. falciparum* transmission have not changed significantly during the past century or during the period of reported malaria resurgence. They conclude that the claimed associations between local malaria resurgences and regional changes in climate are overly simplistic, due to the presence of a high degree of temporal and spatial variation in the climate of East Africa.

Killeen et al. [121] present a kinetic model of mosquito foraging for aquatic habitats and vertebrate hosts that allows estimation of malaria transmission intensity by defining the availability of these resources as the rate at which individual mosquitoes encounter and use them. The model analyzes the individual and combined effects (predicted proportional impact) of four environmental interventions at 80% coverage levels: water management, larvicide application, physical domestic protection, and
zooprophylaxis (the diversion of host-seeking mosquitoes from humans to animals), and an integrated program combining all of these interventions.

Hoshen and Morse [100] describe a weather-driven dynamic mathematical malaria model that outputs new infections in the human host, and is able to capture the gross spatial dynamics of malaria transmission across the African continent. The model uses the ERA-40 weather reanalysis-climate data set, which can provide daily estimates of a range of potentially significant weather variables for the whole globe, and is the reference data for the DEMETER multi-model system (Development of a European Multimodel Ensemble system for seasonal to inTERannual prediction) [53]. By numerical evaluations of the model in both time and space, they show that the model qualitatively reconstructs the prevalence of infection across Africa.

Menach et al. [160] describe a malaria epidemiology model on heterogeneous landscapes that incorporates a more detailed description of the gonotrophic cycle into models for mosquito infection dynamics. In this model, spatial heterogeneity is abstractly incorporated by subdividing the landscape into finite number of patches in an array (grid), where the location and state (e.g., fed, unfed, gravid, infective, etc.) of the mosquito can be identified in each patch. They argue that since mosquitoes commute between bloodmeal hosts and water bodies (aquatic habitats), heterogeneity in human biting reflects the underlying spatial heterogeneity in the distribution and suitability of larval habitat as well as inherent differences in the attractiveness, suitability and distribution of bloodmeal hosts. The model demonstrates that oviposition is an important factor explaining heterogeneous biting and vector distribution in a landscape with a heterogeneous distribution of larval habitat. Since adult female mosquitoes tend to aggregate around places where they oviposit, thereby increasing the risk of malaria (regardless of the suitability of the habitat for larval development), an aquatic habitat may be unsuitable for adult mosquito emergence, but simultaneously can be a source for human malaria.
Chitnis et al. present a linear difference equation model [39] and a deterministic dynamical systems model [40] to describe the dynamics of malaria in a mosquito population interacting with a heterogeneous population of humans, and use them for exploring the impact of combinations of insecticide-treated nets (ITNs) and indoor residual spraying (IRS). They also describe a periodically-forced difference equation model that captures the effects of seasonality and allows the mosquitoes to feed on a heterogeneous population of hosts [41]. An open-source version of the full model, named OpenMalaria, is available online [177].

Yakob and Yan [257] use a mathematical model to analyze the effects of integrating larval habitat source reduction and ITNs on reducing malaria transmission, applied separately and in combination. In order to accelerate the comprehensive effects of integrated vector management (IVM), they argue that attacking multiple points in the transmission cycle may yield synergistic benefits and improve upon current single-tool interventions based on the use of ITNs.

2.5.2 Agent-Based Models (ABMs) of Malaria

In this section, we provide a brief description of ABMs of malaria. In the recent years, they have been extensively used to model the behavior of individual mosquitoes, including interactions within agents and to their environment. We also highlight the spatial dimension of malaria ABMs.

McKenzie et al. [159] develop a discrete-event simulation model using a single time-line variable to represent the P. falciparum life-cycle in individual hosts and vectors within interacting host and vector populations. This work is further advanced in [158] by embedding a differential-equation model of parasite-immune system interactions within each of the individual humans represented in the discrete-event model, and by examining the effects of human population turnover, parasite diversity, recombination, and gametocyte production on the dynamics of malaria. The
integrated approach provides a framework for investigating relationships between pathogen dynamics within an individual host as well as within interacting host and vector populations \[158\].

Depinay et al. \[51\] present an individual-based simulation model of African malaria vectors that incorporates knowledge of the mechanisms underlying *Anopheles* population dynamics and their relations to the environment. The model, being the first to integrate both biological and environmental factors of malaria vector population dynamics, incorporates basic biological requirements for *Anopheles* development. It represents the life cycle of each individual mosquito through four stages: three immature stages which occur in a water body (egg, larva, and pupa), and the mature stage, a flying adult. Using local environmental data as input, it considers five basic factors: temperature, moisture (in the form of precipitation and relative humidity), nutrient competition, predation or death by disease, and dispersal. Results show that the model can reproduce some broad, diverse patterns found in the field, allowing detailed analyses and explanations of vector population dynamics.

Gu and Novak use an agent-based model to report the impact of Larval Source Management (LSM) under various intervention scenarios \[86\], and the impact of ITNs under various levels of diversion (repellence) and coverage of the bednets \[85\]. They represent mosquito foraging as a two-stage process: *random flight* when the resource is not within the mosquito’s perception range, and *directional flight* to the resource when it is detected. They design three scenarios of targeted source reduction to eliminate all aquatic habitats within certain distances of human habitations, and three non-targeted source reductions (for comparison). The LSM study \[86\] shows that the elimination of habitats within 100 m, 200 m, and 300 m of surrounding houses cause 13%, 91%, and 94% reductions in malaria incidence, respectively, compared with

\[2\]In Chapter 6 using our spatial ABM, we investigate the effects of LSM and ITNs (both in isolation and in combination), and compare our results to those reported by Gu and Novak \[86\].
3%, 19%, and 44%, respectively, for the corresponding conventional interventions. They report two major findings: first, source reduction may not require coverage of extensive areas: coverage of up to 300 m surrounding houses could lead to interruption of the gonotrophic cycle and significant reductions in malaria transmission. Therefore, mosquito foraging might be a promising target for malaria control using source reduction. Second, distance to the nearest houses can be the primary measure for habitat targeting, and may serve as an operational indicator in the field. In the ITN study [85], responses of individual mosquitoes to ITNs are modeled by a series of parameters: coverage, repellence (diversion), mortality (insecticide effect of bed-nets), and personal protection (of net users from being bitten). Results show that the application of ITNs could give rise to varying impacts on population-level metrics, which depends on parameter values governing interactions of mosquitoes and treated nets at the individual level. In their simulation results, the most significant factor in determining effectiveness is the killing capability (mortality) of the ITNs. They also show that strong excito-repellent effect of impregnated nets may lead to higher risk exposure to non-bednet users.

Griffin et al. [83] describe an individual-based simulation model for *P. falciparum* transmission in an African context incorporating the three major vector species (*An. gambiae* sensu stricto, *An. arabiensis*, and *An. funestus*) with parameters obtained by fitting to parasite prevalence data from 34 transmission settings across Africa. The model investigates the effect of applying different combinations of long-lasting insecticide-treated nets (LLINs), IRS, artemisinin-combination therapy (ACT), mass screening and treatment (MSAT), and vaccines, in six representative settings with varying transmission intensities (low, medium, and high, as summarized by the annual entomological inoculation rates (EIRs)), vectorspecies combinations, and patterns of seasonality. The results show that interventions using current tools can result in major reductions in *P. falciparum* malaria transmission and the associ-
ated disease burden in Africa, and the combined interventions can result in substantial declines in malaria prevalence across a wide range of transmission settings.

Eckhoff [57] presents a cohort-based vector simulation model that includes mosquito population dynamics, effects of weather, and impacts of multiple simultaneous interventions. The model demonstrates the effects of increasing coverage of interventions with perfect IRS, combining IRS and ITNs, and combining larval control (using larvicides) and space spraying, with a focus on local elimination of malaria. In addition, mosquito population behaviors (e.g., anthropophily, indoor feeding, etc.), are included to study their effect upon the efficacy of vector control-based elimination campaigns.

2.5.2.1 The Spatial Dimension of Malaria Models

We emphasize that the models of malaria can also be classified into non-spatial and spatial models, considering the form of the spatial representation embedded in the models. This classification may overlap with instances from both categories of mathematical and agent-based models (as described above in this section).

The non-spatial malaria models do not model space explicitly. These models usually represent space abstractly (e.g., a lattice of point spaces, patch-based landscapes, etc.), and assume various statistical distributions to model various spatial features (e.g., 25% probability of successful completion of a specific event). Although some models in this category may include space in various (non-trivial) forms, agents (in these models) do not possess explicit spatial attributes, and/or the choices made by the agents, i.e. the effects of actions performed by the agents, do not reflect the use of any spatial feature. These models include the model by Janssen and Martens [107] that simulates the adaptation of mosquitoes and parasites to available pesticides and drugs, the discrete-event simulation model by McKenzie et al. [159, 158], the individual-based simulation model by Depinay et al. [51], the patch-based trans-
mission dynamics model for malaria by Menach et al. [160], etc.

On the other hand, the spatial malaria models explicitly model space. In these models, the agents, as well as their environments, have explicit spatial coordinates. These models include the ABMs of Gu and Novak [86, 85], the cohort-based model of Eckhoff [57], the spatial ABM we present (see Chapters 3 and 5), etc.

As we describe the design and implementation of our spatial ABM, comparison of various characteristics of some models (as described above) with our model is worth noting. We present a summary comparing relevant model features from two mathematical models [257, 41] and two ABMs [85, 57] in Table 2.1. In the table, each row represents a specific model feature, and each column represents a specific malaria model. Features marked with * are either modeled with improvements/extensions, or may be treated as new (not modeled earlier by other studies) in our spatial ABM. Text in the cells represent whether the feature is implemented/available in the model, including simple yes/no, or other comments. N/A means not applicable or not available. Time-step resolution indicates the most fine-grained time resolution (for example, models reported as hourly are also capable of reporting events on a daily basis). For some features, we specify whether the model implements variability of the feature (e.g., variability in daily temperature), and note the default value used in the model in parentheses. For fecundity, \( N \) indicates a normal distribution with mean and standard deviation. Daily mortality and age-dependent mortality refer to mortality of mosquitoes.
### TABLE 2.1

SUMMARY OF FEATURE COMPARISONS FROM SEVERAL MALARIA MODELS

<table>
<thead>
<tr>
<th>Model Feature</th>
<th>Gu &amp; Novak</th>
<th>Yakob &amp; Yan</th>
<th>Eckhoff</th>
<th>Chitnis et al.</th>
<th>Our Spatial ABM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Type</td>
<td>Agent-based</td>
<td>Mathematical</td>
<td>Agent-based</td>
<td>Mathematical</td>
<td>Agent-based</td>
</tr>
<tr>
<td>Spatial representation</td>
<td>Landscape-based</td>
<td>N/A</td>
<td>Abstract</td>
<td>N/A</td>
<td>Landscape-based</td>
</tr>
<tr>
<td>Automation of landscape generation (e.g., using separate tools)</td>
<td>No</td>
<td>N/A</td>
<td>No</td>
<td>N/A</td>
<td>Yes (VectorLand)</td>
</tr>
<tr>
<td>Boundary type of landscape *</td>
<td>Absorbing</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Non-absorbing</td>
</tr>
<tr>
<td>Average of multiple simulations *</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Time-step resolution</td>
<td>Daily</td>
<td>Daily</td>
<td>Hourly</td>
<td>Daily</td>
<td>Hourly</td>
</tr>
<tr>
<td>Age-specific mortality</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>No</td>
<td>Yes (see [9])</td>
</tr>
<tr>
<td>Model Feature</td>
<td>Gu &amp; Novak</td>
<td>Yakob &amp; Yan</td>
<td>Eckhoff</td>
<td>Chitnis et al.</td>
<td>Our Spatial ABM</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td>----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Daily mortality rate (imma-</td>
<td>Fixed (0.2)</td>
<td>Fixed (0.15)</td>
<td>Temperature-</td>
<td>N/A</td>
<td>Age-specific (for larvae)</td>
</tr>
<tr>
<td>ture stages)</td>
<td></td>
<td></td>
<td>dependent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily mortality rate (adult</td>
<td>Fixed (0.2)</td>
<td>Fixed (0.15)</td>
<td>Adult life expectancy of 10</td>
<td>N/A</td>
<td>Age-</td>
</tr>
<tr>
<td>stages)</td>
<td></td>
<td></td>
<td>days</td>
<td></td>
<td>specific</td>
</tr>
<tr>
<td>Fecundity (eggs/oviposition)</td>
<td>Fixed (80)</td>
<td>N/A</td>
<td>Fixed (100)</td>
<td>N/A</td>
<td>$N(170, 30)$</td>
</tr>
<tr>
<td>Variability in daily tempera-</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (25°C)</td>
</tr>
<tr>
<td>ture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of individual simulation run</td>
<td>200 days for</td>
<td>N/A</td>
<td>&gt; 6 years</td>
<td>N/A</td>
<td>1 year</td>
</tr>
<tr>
<td></td>
<td>LSM, 300 days</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for ITNs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interventions modeled</td>
<td>LSM, ITNs</td>
<td>LSM, ITNs</td>
<td>IRS, ITNs, larvicides,</td>
<td>ITNs, IRS</td>
<td>LSM, ITNs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>space spraying</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Feature</td>
<td>Gu &amp; Novak</td>
<td>Yakob &amp; Yan</td>
<td>Eckhoff</td>
<td>Chitnis et al.</td>
<td>Our Spatial ABM</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------------</td>
<td>-------------</td>
<td>---------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Time-step of intervention application</td>
<td>Day 100 for LSM, day 150 for ITNs</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Day 100</td>
</tr>
<tr>
<td>Explores combined interventions</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Variability in human population</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Coverage scheme used for ITNs *</td>
<td>Proportion of households with bednets</td>
<td>Proportion of populations sleeping under bednets</td>
<td>Proportion of populations sleeping under bednets</td>
<td>Proportion of populations sleeping under bednets</td>
<td>Partial and complete coverage</td>
</tr>
<tr>
<td>Comparison of coverage schemes for ITNs *</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
2.6 Summary

In this chapter, we presented a general background about ABM, malaria, and the applicability of ABMs in malaria modeling. We also described some mathematical and agent-based malaria models. We find it particularly relevant to compare some model characteristics of our ABM with some of the malaria models discussed in this chapter (see Table 2.1).

Among the benefits of ABMs, the ability to handle emergent phenomena is what drives the other benefits. To summarize, using an ABM is more convenient when the interactions between the agents are complex and nonlinear, the space is crucial and agents are non-stationary, and the population and topology are heterogeneous, all of which, in general, hold true for malaria modeling. We discussed the advantages of ABM over other modeling techniques, and some of its disadvantages.

In the following chapters (Chapters 3–5), we describe the design, implementation, and verification & validation (V&V) issues of the ABMs, and a spatial extension. We also evaluate the ABM with vector control interventions (Chapter 6), and demonstrate an example application by integrating a GIS (Chapter 7). As we progress, we show that our ABM inherently contains many of the characteristics mentioned above, and hence an ABM is a convenient choice to model the vector and/or host populations in malaria research.
CHAPTER 3

THE CORE MODEL AND THE SIMULATION MODEL

3.1 Overview

In this chapter, we describe the biological core model, and the agent-based implementation derived from the core model\(^1\). The core model, essentially conceptual in nature, is governed by the biology underlying An. gambiae. Evolution of the core model has been guided by relevant biological concepts concerning An. gambiae. We iteratively refine existing biological concepts, and incrementally add new concepts to the core model to incorporate real-world complexities. Subsequently, we update the ABM to reflect the changes. In the evolutionary process, we have several versions of the core model. In this chapter, we describe one fixed version.

The biological core model primarily involves the vector population dynamics, which is regulated by the distinct stages of An. gambiae mosquito life-cycle, both for the immature and adult stages. Special attention is given on mortality for each specific stage, as well as on habitat-regulated oviposition mechanism that eventually drives the vector population.

The organization of this chapter is as follows: Sections 3.2–3.6 describe the biological core model. In Section 3.2, we define some relevant terms, discuss the life-cycle stages of An. gambiae mosquitoes and the malaria parasite. Section 3.3 describes temperature-dependent larval development. In Section 3.4, we discuss the need to

\(^1\)The biological core model has been developed through discussions and results of simulation projects performed in the course *Computer Simulation*, taught over Fall 2009 and Spring 2010 at the University of Notre Dame. We would like to thank the instructors Frank H. Collins and Gregory R. Madey, and all students who actively contributed to the development process.
incorporate senescence for vector mortality, and discuss the mathematical formulation of basic mortality models. Based on the age-dependent mortality discussed in Section 3.4, Section 3.5 describes the concept of Age-Specific Mortality Rate (ASMR) for all the adult stages and the larvae stage. In Section 3.6 we describe the aquatic habitats and the oviposition mechanism, as adopted in the core model.

Section 3.7 describes the non-spatial ABM\(^2\) which is built according to the specification of the core model. We describe the implementation details based on one specific version of the core model (as described in Sections 3.2–3.6). We also discuss the agents, their environments, and the features, initialization parameters, and assumptions of the ABM. Section 3.8 concludes this chapter.

3.2 Life-Cycles

Before discussing the life-cycle stages and other major concepts of the biological model, we define some terms of interest.

Parasite The term parasite refers to any organism that lives in or on another organism without benefiting the host organism. Parasites are commonly referred to as pathogens, which are organisms that can cause disease.

Malaria parasite Malaria parasites are micro-organisms that belong to the genus Plasmodium. There are more than 100 species of Plasmodium, which can infect many animal species such as reptiles, birds, and various mammals. The malaria parasite requires specific human and mosquito tissues to complete its life-cycle. Once inside a human, the parasite develops and multiplies, causing periodic bouts of flu-like symptoms, including fever, headache, and chills. The developing parasites destroy red blood cells, which may cause death by severe anemia as well as by the clogging of capillaries that supply the brain or other vital organs with blood.

Only four species of Plasmodium infect humans in nature. They are known as *P. falciparum*, *P. malariae*, *P. ovale*, *P. vivax*. The deadliest of these is *P. falciparum*, which is responsible for about 80% of all malaria cases, and is also responsible for about 90% of the deaths from malaria [161].

\(^2\)A spatial extension of the ABM is presented in Chapter 5.
Vector  The term *vector* refers to the carrier of a pathogen from one host to another. In epidemiology, a vector is an insect or any living carrier that transmits an infectious agent. Vectors are vehicles by which infections are transmitted from one host to another. A vector is not only required for part of the parasite’s developmental cycle, but it also transmits the parasite directly to subsequent hosts [245]. For this reason, a mosquito is treated as *vector* as well as a *host* for malaria transmission [155].

*An. gambiae* Human malaria is transmitted only by females of the genus *Anopheles*. Female *Anopheles* mosquitoes are the primary (definitive) hosts and transmission vectors of the malaria parasite *P. falciparum*. *An. gambiae* is one of the best known malaria vectors because of its predominant role in the transmission of the parasite [60].

Like all other mosquito species, anophelines go through four stages in their life-cycle: egg, larva, pupa, and adult. The first three stages are aquatic and last 5-14 days, depending on the particular species and the ambient temperature [3], and they are collectively termed as the immature stages. The adult stage is when the female *Anopheles* mosquito acts as a malaria vector. The adult females can live up to a month (or more in captivity), but most do not live more than 1-2 weeks in nature [3]. In the following, we describe the *An. gambiae* life-cycle and the *Plasmodium* life-cycle in detail.

3.2.1 Life-Cycle of Immature *An. gambiae* Mosquitoes

Although the immature stages do not themselves transmit malaria, they play a major role in determining the form which diseases assume [155]. The course of development during the immature stages, from a fertilized egg to a mature adult, is broadly the same in all mosquitoes: an oviposition is followed by maturation of the embryo, hatching, three successive larval moults, pupation, and finally the emergence of the adult. The three aquatic stages are described below:

- **Egg**: Adult females lay 50-200 eggs per oviposition. Eggs are laid singly and directly on water, and are unique in that they have floats on either side. Eggs are not resistant to drying; they hatch within 2–3 days in tropical temperatures, although hatching may take up to 2–3 weeks in colder climates [3].
• **Larva**: Larvae (larva - singular) have a well-developed head with mouth brushes used for feeding, a large thorax, a segmented abdomen, and no legs. In contrast to other mosquitoes, *Anopheles* larvae lack a respiratory siphon and for this reason position themselves so that their body is parallel to the surface of the water [3]. They are also commonly called *wigglers* or *wrigglers*. They feed on algae, bacteria, and other microorganisms in the water surface micro-layer. They dive below the surface only when disturbed. Larvae swim either by jerky movements of the entire body or through propulsion with the mouth brushes. During growth, the larvae molt shedding their exoskeleton, or skin, to allow for further growth [3]. The stages between molts are called *instars*. Larvae develop through four instars, after which they metamorphose into pupae [3].

The larvae occur in a wide range of habitats but most species prefer clean, unpolluted water. Larvae of *Anopheles* mosquitoes have been found in fresh- or salt-water marshes, mangrove swamps, rice fields, grassy ditches, the edges of streams and rivers, and small, temporary rain pools. Many species prefer habitats with vegetation. Others prefer habitats that have none. Some breed in open, sun-lit pools while others are found only in shaded breeding sites in forests. A few species breed in tree holes or the leaf axils of some plants [3].

• **Pupa**: The pupal stage is a resting, non-feeding stage. This is the time the mosquito turns into an adult. Pupae (pupa - singular) are comma-shaped when viewed from the side. Pupae are lighter than water and therefore can float at the surface. As with the larvae, pupae must come to the surface frequently to breathe, which they do through a pair of respiratory trumpets on the cephalothorax. After a few days as a pupa, the dorsal surface of the cephalothorax splits and the adult mosquito emerges [3]. They are also commonly called *tumblers*.

The metamorphosis of the mosquito into an adult is completed within the pupal case. The adult mosquito splits the pupal case and emerges to the surface of the water where it rests until its body can dry and harden. The duration from egg to adult varies considerably among species and is strongly influenced by ambient temperature. Mosquitoes can develop from egg to adult in as little as 5 days, but usually take 10–14 days in tropical conditions [3].

3.2.2 Life-Cycle of Adult *An. gambiae* Mosquitoes

Adult mosquitoes primarily have two objectives, to mate and to feed. They usually mate within a few days after emerging from the pupal stage. In most species, the males form large swarms, usually around dusk, and the females fly into the swarms to mate [60]. The males, after mating, seek out for nectar to feed on from plants and fruits. Females also feed on sugar sources for energy but usually require a bloodmeal
for the development of eggs. After obtaining a full bloodmeal, the female will rest for a few days while the blood is digested and the eggs are developed. Once it hatches and oviposits the eggs, it searches for another bloodmeal. This process of blood-feeding, egg maturation, and oviposition is repeated several times throughout the life-cycle until the female dies, and is referred to as the gonotrophic cycle. Length of an adult’s life-cycle usually depends on several factors, including sex of the mosquito, temperature, rainfall, humidity, and season of the year. We incorporate the following distinct stages of the adult life-cycle into the simulation model:

- **Immature Adult**: In this stage, a mosquito emerges as an adult when its aquatic development is complete.

- **Mate Seeking**: In this stage, a pair of male and female mate. Once mating is done, a female immediately goes out in search of a human bloodmeal, while the male just rests until it dies.

- **Bloodmeal Seeking**: In this stage, a female seeks out for a bloodmeal. In general, mosquitoes seek their host in response to a combination of chemical and physical stimuli, including carbon dioxide plumes, body odors, warmth, movement, etc. [173]. Anophelines feed most frequently at night and occasionally in the evening, or in shaded areas during the early morning. During a bloodmeal, the mosquito injects a minute amount of salivary fluid into the host to increase blood flow to the area. Thus, if the mosquito is infective, sporozoites are transmitted to the host via the salivary fluid.

- **Bloodmeal Digesting**: After obtaining a full bloodmeal, the female goes to the Bloodmeal Digesting stage. This is also termed as the resting stage, because the female simply rests for a few days while the blood is being digested and eggs are being developed. Some females seek out cool, humid areas of a house (e.g., walls, undersides of furniture, etc.), while others find dark spots outdoors near the ground.

- **Gravid**: In this stage, the eggs being developed, and the female is ready to lay them to some aquatic habitat. The maximum number of eggs a female may oviposit, $E_{max}$, is drawn from a normal distribution with some predefined mean and standard deviation (e.g., mean = 170, s.d. =30). The parent female then resumes host seeking, and this process is repeated until the female dies.

Figure 3.1 shows the stages of mosquito agents’ life-cycle. As a convention, we italicize the names of the stages. Table 3.1 lists the stage transition times for mosquito
TABLE 3.1

STAGE TRANSITION TIMES FOR MOSQUITO AGENTS

<table>
<thead>
<tr>
<th>Stage</th>
<th>Duration (in Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egg</td>
<td>1</td>
</tr>
<tr>
<td>Larva</td>
<td>Temperature-dependent; see Equation (3.1)</td>
</tr>
<tr>
<td>Pupa</td>
<td>1</td>
</tr>
<tr>
<td>Immature Adult</td>
<td>1, 2, or 3 days $[p(1) = 0.1; p(2) = 0.8; p(3) = 0.1]$</td>
</tr>
<tr>
<td>Mate Seeking</td>
<td>0 (immediate transition)</td>
</tr>
<tr>
<td>Bloodmeal Seeking</td>
<td>2 or 3 days $[p(2) = 0.7; p(3) = 0.3]$</td>
</tr>
<tr>
<td>Bloodmeal Digesting</td>
<td>0 (immediate transition)</td>
</tr>
<tr>
<td>Gravid</td>
<td>0 (immediate transition when no eggs remaining)</td>
</tr>
</tbody>
</table>

agents in the core model. The probability-based transition times, where applicable, are denoted by the $p(x)$ notation, where $x$ indicates the number of days required for the transition (for example, a mosquito may require two days to be in the Immature Adult stage with an 80% probability).

3.2.3 Life-Cycle of the Malaria Parasite

The malaria parasite, *Plasmodium*, relies on two hosts (the female *Anopheles* mosquito and humans) to complete its life-cycle. The parasite is transmitted through the bite of an infected *Anopheles* mosquito to a human host. The parasites undergo many asexual replicative stages within the human host. One blood stage form (gametocyte) is transferred to the mosquito when the mosquito bites an infected human host. Once inside the mosquito, the parasite replication cycle is completed. The life-cycle of the malaria parasite (see Figure 3.2) is briefly outlined below:
Figure 3.1. Life-Cycle Stages of Mosquito Agents. Each oval represents a stage in the model. As a convention, we italicize the names of the stages. The rectangles represent durations (e.g., 24h) for the fixed-duration stages. The symbol $h$ denotes hour. Stages with temperature-dependent duration are described in Table 3.1. Permissible time transition windows (from one stage to another) are shown next to the corresponding stage transition arrows as rounded rectangles (e.g., 6pm-6am). We assume that death sinks capture the agents that die out in each stage, and omit these sinks from the figure for simplicity. Note that adult males, once reaching the Mate Seeking stage, remain forever in that stage until they die; adult females, on the other hand, cycle through obtaining bloodmeals (in Bloodmeal Seeking stage), developing eggs (in Bloodmeal Digesting stage), and ovipositing these eggs (in Gravid stage) until they die. This is referred to as the gonotrophic cycle.
• an *Anopheles* mosquito, infected with malaria sporozoites, bites a human

• *Plasmodium* sporozoites enter the bloodstream and travel to the liver, where they invade liver cells and divide

• new morphological stages of the parasite called merozoites emerge from the liver and infect red blood cells

• merozoites infect red blood cells and rapidly replicate

• some merozoites in red blood cells develop into a sexual form known as gametocytes

• when an *Anopheles* mosquito bites the infected person, it ingests these gametocytes

• in the mosquito’s digestive system, the gametes combine to form a zygote

• the zygote goes through several developmental stages to form sporozoites

• the sporozoites migrate to the salivary glands of the mosquito

• when the insect bites another person, the cycle repeats

3.3 Temperature-Dependent Larval Development

The growth and development of larvae, however, is much slower and depends on the daily temperature. Details about the formulation of models that analyze the effect of temperature on the growth and development of larvae, along with the model parameters, are described in Appendix B. For the core model, we adopt the Depinay et al. model (see Section B.6). To define the daily development of larva, $Development_{day}$, we use the equation:

$$Development_{day} = (Temperature_{day} \times 0.000305 - 0.003285) \times 24$$  \hspace{1cm} (3.1)

where $Temperature_{day}$ is the average temperature (in °C) of the day.

The threshold for larval development, $DevelopmentThreshold_{larva}$, represents the threshold for the Larva stage (i.e., when the development is complete, and the pupation begins). To allow 10% variability, it is chosen from a normal random variable $G$.
Figure 3.2. Life-Cycle of Malaria (Adapted from [37]).
mean = 1.0, s.d. = 0.1:

\[
DevelopmentThreshold_{\text{larva}} = 1 + G(0, 0.1) \tag{3.2}
\]

Each day, each larva approaches its threshold by developing at a rate characterized by Equation (3.1), and Development\text{day} is accumulated into the cumulative larval development time, CumulativeDevelopment\text{larva}. The larva transforms to a pupa when the following condition is satisfied:

\[
CumulativeDevelopment_{\text{larva}} \geq DevelopmentThreshold_{\text{larva}} \tag{3.3}
\]

3.4 Vector Mortality Models

In most epidemiology models, the mortality of the organisms being modeled plays a crucial role in shaping the model’s characteristics. Daily mortality is the most important determinant of a mosquito’s ability to transmit pathogens [225]. In this section, we discuss the importance of incorporating senescence (biological aging) into vector mortality. Then, we discuss the mathematical formulation of the basic mortality models.

3.4.1 Incorporating Senescence

Traditionally, most malaria transmission models assume age-independent (i.e., non-senescent) vector mortality. Non-senescence of the vector could only lead to approximate estimates and misleading predictions since they obscure the age-dependent aspects of the mosquito biology. It assumes an unrealistic, simplified view that the vector potential of all mosquitoes, regardless of their age, is the same. This, in turn, also affects other determinants of pathogen transmission (e.g., biting rate, host preference, vector competence, dispersal, resistance to insecticides, etc.).
Several studies have shown the impact of vector senescence on malaria transmission. The longer a mosquito lives, the more likely it is to encounter an infectious host, survive the incubation period, and transmit an infectious agent during subsequent feeding attempts. Thus, small changes in daily mortality can result in relatively large changes in the pathogen transmission cycle. Using large-scale laboratory life-table techniques ($N > 100,000$), Styer et al. [225] show that mosquito mortality was low at young ages (less than 10 days old), steadily increased at middle ages, and decelerated at older ages. Clements and Paterson analyze the mortality and survival rates in wild populations of mosquitoes of eleven tropical species [43]. Their results indicate that in most of these populations the adult female mortality rates are age-dependent.

Bellan [22] develops a simple mathematical model of vector-borne disease transmission to assess how relaxing the classical assumption of constant mortality affects the predicted effectiveness of anti-vectorial interventions. He shows that with constant mortality assumption, control of survival reduces the life expectancy of all mosquitoes by a large amount. With age dependence, however, the reduction in life expectancy is less dramatic, and skewed towards younger age classes. By defining $C^*$ as the scaled vectorial capacity, he shows that while reducing survival dramatically reduced the $C^*$ contribution of older mosquitoes in a constant mortality model, this effect is less important in age dependent models. Thus, the constant mortality model overestimates the effectiveness of reducing survival in controlling transmission. The effectiveness of mosquito control when mosquitoes die at age-dependent rates is compared across different extrinsic incubation periods. He concludes that future transmission models that examine anti-vectorial interventions should incorporate realistic age-dependent mortality rates.
3.4.2 Mathematical Formulation

In this section, we briefly discuss the mathematical formulation of the basic mortality models. Standard age-dependent mortality models are based on age-specific mortality rate, \( u(x) \), which is an instantaneous measure of mortality that can be estimated from empirical data with the approximation:

\[
u(x) = -\ln p(x)\]

where \( p(x) \) is the probability that an individual alive at age \( x \) survives to age \( x + 1 \).

The mortality patterns are well-described by the Gompertz mortality function [81], according to which the mortality rate increases with age in such a manner that its logarithm is directly proportional to age. Gompertz suggests that a law of geometric progression dominates the mortality after a certain age. Gompertz mortality model can be represented as:

\[
u(x) = a \cdot e^{bx}\]

where \( u(x) \) is the age-specific mortality rate, \( a \) is the baseline mortality rate, \( b \) is the senescent/aging component (the exponential mortality increase with age), and \( x \) is the age of the age-cohort considered. In 1860, Makeham [143] extended the Gompertz model by adding an age-independent constant \( c \):

\[
u(x) = a \cdot e^{bx} + c\]

The logistic mortality model is defined as [225]:

\[
u(x) = \frac{a \cdot e^{bx}}{1 + \left( \frac{a \cdot e^{bx}}{b} \right)(e^{bx} - 1)}\]

where \( s \) is the degree of mortality deceleration.
3.5 Mortality in the Core Model

Based on the age-dependent mortality models discussed in Section 3.4, we use the concept of *Age-Specific Mortality Rate (ASMR)* for all the adult stages and the larvae stage. All mortality rates, as adopted in the core model, are discussed below.

3.5.1 Adult Stages

For the adult stages, ASMR refers to the total number of deaths per day per 1000 mosquitoes of a given age (throughout all adult stages). We use a modified version of the logistic mortality model in which the age-dependent component of mortality increases exponentially with age \([225]\). Newly emergent adults begin with a daily mortality rate of \(\alpha\). As the mosquito ages, the age-specific mortality rate \(ASMR_{Age(\text{adults})}\) for that age-cohort changes based on the following equation:

\[
ASMR_{Age(\text{adults})} = \frac{\alpha \cdot e^{\frac{Age}{\beta}}}{1 + \alpha \cdot s \cdot \beta(e^{\frac{Age}{\beta}} - 1)}
\]  

(3.4)

where \(\alpha\) is the baseline daily mortality rate, \(\beta\) is the inverse of exponential mortality increase with age, \(s\) is the degree of mortality deceleration, and \(Age\) is the common age of the age-cohort. The number of adults that are removed from the system, \(ToKill_{Age(\text{adults})}\), is then computed by:

\[
ToKill_{Age(\text{adults})} = ASMR_{Age(\text{adults})} \cdot Adults_{Age}
\]  

(3.5)

where \(Adults_{Age}\) is the size of the adult age-cohort.

3.5.2 Immature Stages

Immature mosquitoes live in the aquatic habitats, and have different mortality rates. We use a constant mortality rate \(\alpha\) of 10% for both *Egg* and *Pupa* stages. Mor-
tality in the Larva stage, however, requires consideration of several factors. Larvae mortality, unlike adults mortality, depends on:

- age of larvae,
- effects of predation and cannibalism in the larval population,
- carrying capacity of the aquatic habitat, and
- rainfall

The effects of predation and cannibalism on larvae mortality have been investigated by several studies. For example, Koenraadt and Takken [126] explore the occurrence of predation and cannibalism within and between members of the *An. gambiae* complex, assessing the effects of larval food availability and the presence of older instars on the development of younger instars. This study shows that older larvae of the *An. gambiae* complex are able to prey on younger larvae of conspecifics (cannibalism) as well as on larvae of closely related species (predation). Biotic factors in and around these aquatic habitats, such as the presence of predators, parasites and pathogens affect the survival and growth of *An. gambiae* s.l. larvae.

Koenraadt et al. [127] analyze the impact of the presence of a fourth-instar larva (*An. gambiae* Giles s.s. or *An. arabiensis* Patton), the quantity of food, and the available space on the survival and development of freshly hatched larvae by constructing two proportional hazard models. Results suggest that cannibalism and predation occurred readily. Limitation in space significantly increased mortality of larvae, whereas a limitation of food reduced larval development rate but did not affect mortality. They conclude that inter- and intra-specific interactions among larvae of the *An. gambiae* complex strongly affect survival and development, and that the quantity of food and the available space are important determinants of the outcome of these interactions. Studies on inter- and intra-specific competition show that competition may result in reduced larval development rate, lower larval survival, smaller adult size, lower fecundity, and a distorted sex-ratio [127]. Over time, these effects
may even lead to the replacement of one species by another. Increasing densities of *An. gambiae* negatively affects larval survival, development rate, and adult size. Some aquatic habitats of *An. gambiae* may dry up within a few days. This transient nature leads to inter- and intra-specific competition for food and space \[127\].

In the core model, we incorporate the effects of predation and cannibalism in an aquatic habitat by using the notion of *one-day old equivalent larval population*, $N_e$. Within a given aquatic habitat, we compute $N_e$ by first multiplying the number of larvae in each age-cohort by the age, and then summing up the values for all age-cohorts:

$$
N_e = \sum_{Age=0}^{MaxAge} Age \times Larvae_{Age}
$$

where $Age$ and $Larvae_{Age}$ are the common age and size of the larvae age-cohort, respectively, and $MaxAge$ is the age of the oldest larvae age-cohort in the system.

As mentioned before, larvae mortality is also affected by rainfall (precipitation). Paaijmans et al. \[180\] explore the effect of natural rainfall as a density-independent factor on flushing, ejection and mortality of larvae of *An. gambiae* under ambient conditions in western Kenya. Their results show that precipitation flushed, ejected and killed a significant proportion of larvae of *An. gambiae* in different stages of development. Young larvae (L1 stage) experience the highest flushing, ejection and mortality, while the oldest larvae (L4 stage) are better able to withstand the effects of precipitation. This study demonstrates that immature populations of malaria mosquitoes suffer high losses during rainfall events. Thus, rainfall has a profound effect on the productivity of mosquito breeding sites and, as a result, on malaria transmission.

Based on the factors discussed above, we use $N_e$ to calculate the age-specific mortality rate $ASMR_{Age(larvae)}$ for each larvae age-cohort using the following equation:

$$
ASMR_{Age(larvae)} = \alpha \times e^{N_e \times CC \times R}
$$

(3.7)
where \( C \) is the carrying capacity of the aquatic habitat and \( R \) is the rainfall coefficient (for now, we assume a rainfall coefficient of 1.0). The number of larvae with a given age that are removed from the system, \( ToKill_{Age(larvae)} \), is then computed by:

\[
ToKill_{Age(larvae)} = ASMR_{Age(larvae)} \times Larvae_{Age}
\]  (3.8)

3.6 Aquatic Habitats and Oviposition

The immature stages of \textit{An. gambiae} require aquatic environments to develop. These habitats are often found in transient, sunlit and generally small pools \[180\]. The availability of these transient aquatic habitats primarily depends on rainfall.

We associate each aquatic habitat with a finite \textit{carrying capacity} (\( CC \)), which is treated as a soft limit on the aquatic mosquito population that the aquatic habitat can sustain. In the core model, \( CC \) serves two purposes: 1) it limits the number of eggs a female mosquito may oviposit in an aquatic habitat (thus determining soft limits on larval density of the habitat); and 2) is used to model the \textit{Gravid} female’s inclination to avoid less suitable (e.g., over-crowded) habitats. \( CC \) can also be defined as the environments maximal load, and represents the population size of a biological species that the environment can sustain indefinitely, given the available resources (food, water, etc.) in the environment. Unlike other studies \[86 \[85\], we do not treat \( CC \) as a hard limit in our model.

The \textit{biomass}, which poses a theoretical limit on the population size of the aquatic habitat, is defined as the sum of the eggs, pupae, and the one-day old equivalent larval population \( N_e \) (see Equation (3.6)), in the selected aquatic habitat:

\[
\text{biomass} = Eggs_{habitat} + N_e + Pupae_{habitat}
\]  (3.9)

where \( Eggs_{habitat} \) and \( Pupae_{habitat} \) are the number of eggs and pupae, respectively, in
the selected habitat.

*Oviposition* is the process by which gravid females lay new eggs. According to Koenraadt and Takken [126], *An. gambiae* s.l. females tend to avoid oviposition sites containing older instar larvae, and thus reduce the risk of predation of offspring. It is not fully understood, however, which physical or chemical cues female mosquitoes use for selection of sites optimal for the development of their offspring.

In the core model, new mosquito agents are created (in the form of eggs) when females in the *Gravid* stage visit an aquatic habitat and oviposit. Each day, *Gravid* females make a maximum of three attempts to lay the entirety of eggs. However, the potential number of eggs, $E_{potential}$, that a female can actually lay in a given habitat is limited by the biomass already present in the habitat:

$$E_{potential} = E_{max} \times \left(1 - \frac{biomass}{i \times CC}\right) \quad (3.10)$$

where $E_{max}$ is the maximum number of eggs available to lay, $N_e$ is the one-day old equivalent larval population of the habitat (see Equation (3.6)), $i$ is the oviposition attempt number (i.e., 1, 2, or 3), and $CC$ is the carrying capacity of the habitat.

Another quantity, $E_{remaining}$, keeps track of how many eggs the female still has to lay, and is updated every time the female actually lays eggs. Whenever $E_{remaining}$ becomes 0, the female is equipped with a new bunch of eggs by setting its $E_{max}$ to a value pulled from the normal distribution (once it arrives in the *Gravid* stage).

In an oviposition attempt, if $E_{remaining} \geq E_{potential}$, the female lays $E_{potential}$ number of eggs, and $E_{remaining}$ is reduced by $E_{potential}$. Otherwise, the female has room to lay all of the remaining eggs ($E_{remaining}$), and it lays them. If the female still has $E_{remaining} \geq 0$, another oviposition attempt occurs. If, after three oviposition attempts, $E_{remaining} \neq 0$, the female stays another day in the *Gravid* stage, attempting to lay the remaining eggs. In each attempt, the actual number
of eggs laid, $Eggs_{laid}$, is recorded. Once all remaining eggs are laid, the female transitions back to the Bloodmeal Seeking stage. These rules are depicted as logical flowcharts in Appendix A.

3.7 The Agent-Based Model (ABM)

In this section, we describe the ABM, which is built according to the specification of the biological core model (as described in Sections 3.2–3.6).

The ABM simulates the life cycle of the vector $An.\ gambiae$ by tracking attributes relevant to the vector population dynamics for each individual mosquito. It is developed in Java [109], using the Eclipse Software Development Kit (SDK, Version: 3.5.2), which is freely available from [220]. Some of the model features (e.g., random number generators, probability distributions, etc.) are implemented using Repast (Recursive Porous Agent Simulation Toolkit, [196]), which is a software framework for agent-based simulations created by the Social Sciences Computing Services at the University of Chicago [221]. Repast provides an integrated library of classes for creating, running, displaying, and collecting data from an agent-based simulation. The Java implementation, being highly portable, provides some key advantages. First, Repast provides built-in graphical visualization tools allowing the agents to be readily inspected. This aids in the debugging process, and results can be shared rapidly in graphical form. Second, programming in Java is more efficient and results in less error-prone code.

Figure 3.3 shows a class diagram for the model, listing its four main classes: AnophModel, AquaticHabitat, HumanHabitat, and MosquitoAgent. It depicts the major attributes (member variables), operations (member functions) and relationships within the classes. In the Java implementation, these classes are organized in different packages. A detailed flowchart of the ABM is included in Appendix A.

An instance of the AnophModel class creates all other required class instances,
Figure 3.3. Class Diagram for the ABM. The model consists of four major classes: AnophModel, AquaticHabitat, HumanHabitat, and MosquitoAgent. An instance of the AnophModel class creates all other required class instances, initiates and runs the simulation, and logs outputs to time-stamped output text files. It can have one or more instances of the AquaticHabitat class, which represent an aquatic habitat for the mosquito agents. It can also have one or more instances of the HumanHabitat class, which represent a human habitat with the number of persons living in the habitat specified. The aquatic and human habitats instances are accessed through aquaticHabitatList and humanHabitatList, respectively, by the AnophModel instance. The mosquito agents are created as instances of the MosquitoAgent class, and can be accessed through agentLists by both the AnophModel and AquaticHabitat instances. A detailed flowchart of the model is included in Appendix [A]. In this figure, only the major attributes and operations of the classes are shown.
initiates and runs the simulation, keeps track of the simulation time, instantiates initial mosquito agents, and logs outputs to time-stamped output text files. It can have one or more instances of the `AquaticHabitat` class, representing aquatic habitats with specific carrying capacities for the mosquito agents. It can also have one or more instances of the `HumanHabitat` class, representing human habitats (houses) with the number of persons living in the habitats specified. The aquatic and human habitats instances are accessed through `aquaticHabitatList` and `humanHabitatList`, respectively, by the `AnophModel` instance. The mosquito agents are created as instances of the `MosquitoAgent` class, which represent individual mosquito agents. Agents can be accessed through `agentLists` by both the `AnophModel` and `AquaticHabitat` instances. When (a subset of) the aquatic agents, residing in a specific aquatic habitat, turn into adult agents (via emergence, as shown in Figure 3.1), they are transferred from the respective aquatic `agentList` to the adult `agentList` (of the `AnophModel` instance).

Note that a `HumanHabitat` instance does not have an explicit `agentLists`, since at a given point in time, we do not (yet) need to keep track of mosquitoes per house (although this information can still be generated by probing the `MosquitoAgent` instances).

3.7.1 Agents & Environments

*An. gambiae* mosquitoes are the only agents in the simulation model. The lifecycle stages, as shown before in Figure 3.1, are modeled in the ABM with duration in each stage, as defined in Table 3.1. Female adult mosquitoes go through all five adult stages: Immature Adult, Mate Seeking, Bloodmeal Seeking, Bloodmeal Digesting, and Gravid. Male adult mosquitoes, unlike females, go through only the first two stages: Immature Adult and Mate Seeking. Aquatic mosquitoes reside in the aquatic habitats, and go through three aquatic stages: Egg, Larva, and Pupa. Being autonomous agents, mosquitoes begin their life as eggs (except the initial adults created at the
start of the simulation), and progress through various life-cycle stages.

For the non-spatial ABM, the environment for all adult mosquito agents is modeled as an \textit{abstract point in space} (in Chapter 5, we describe a spatial extension, in which environments can have spatial information). The aquatic agents reside in aquatic habitats which are also modeled as \textit{point} spaces (see Section 3.6). Each aquatic habitat has its own characteristics (e.g., carrying capacity, biomass etc.), and serves as a \textit{sub-environment} in the ABM. In addition, some human habitats (houses, also modeled as \textit{point} spaces), with the number of persons residing in each house, are also specified. Houses serve as bloodmeal locations for host-seeking female adult mosquito agents.

3.7.2 Model Features

In the beginning, although we started the ABM with a \textit{daily} time-step resolution, we later modify it to incorporate an \textit{hourly} time-step resolution. The \textit{hourly} time-step, in general, offers finer granularity of control over the agents’ actions, and over other model-specific events. It also allows more flexibility in modeling certain agent behaviors (e.g., host-seeking to start at a particular hour at night). The ABM currently includes the following major features:

- Adult mosquito agents are killed according to the age-specific mortality rate, \textit{ASMR_{Age(adults)}}; the calculated number of adult mosquitoes \textit{die out} from each age-cohort (see Equations (3.4) and (3.5)).

- Eggs and pupae are killed with a constant daily mortality rate of 10%.

- Larvae are killed according to the age-specific mortality rate, \textit{ASMR_{Age(larvae)}}, which depends on age, \textit{one-day old equivalent larval population} \((N_o)\), carrying capacity of the aquatic habitat \((CC)\), and rainfall coefficient \((R)\); the calculated number of larvae \textit{die out} from each age-cohort (see Equations (3.7) and (3.8)).

- As the simulation goes, the abundance graphs (e.g., total number of male and female adults versus time) are dynamically generated and relevant information is written to files for further analysis.

- Dead mosquito agents are immediately removed from the system.
### Table 3.2

#### DEFAULT MODEL PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation runtime</td>
<td><em>stopTick</em></td>
<td>365 days</td>
</tr>
<tr>
<td>Daily temperature</td>
<td><em>dayTemperature</em></td>
<td>25°C</td>
</tr>
<tr>
<td>Number of aquatic habitats</td>
<td><em>numAquaticHabitats</em></td>
<td>Variable</td>
</tr>
<tr>
<td>Number of human habitats</td>
<td><em>numAquaticHabitats</em></td>
<td>Variable</td>
</tr>
<tr>
<td>Carrying capacity of habitat</td>
<td><em>carryingCapacity</em></td>
<td>1000</td>
</tr>
<tr>
<td>Fecundity (first egg-batch)</td>
<td><em>fecundity</em></td>
<td>mean = 170, s.d. = 30</td>
</tr>
</tbody>
</table>

### 3.7.3 Model Initialization

The ABM is initialized as follows:

- 1000 adult mosquitoes, with equal male-female distribution
- A given number of aquatic habitats, all initially empty
- A given number of human habitats (houses), with the number of persons specified in each house

Other default model parameters are described in Table 3.2.

### 3.7.4 Model Assumptions

The ABM represents a theoretical (as opposed to field-based) model. We concentrate on the vector life-cycle dynamics and do not yet include the parasite life-cycle and the malaria transmission cycle. Mosquitoes senesce, and their probability of death increases with age, which is modeled via age-specific mortality rates. The human population is modeled as static, i.e., humans do not move in space. All humans
are assumed to be identical. For host-seeking, we do not model alternative hosts for blood-feeding (e.g., cattle), and the only bloodmeal-sources are humans in the houses. Daily temperature, variations in which can affect our model’s output, is fixed at 25°C. We do not account for seasonality and other weather/climate parameters. Each aquatic habitat is set with a carrying capacity (CC) of 1000. Time is modeled with hourly (instead of daily) time-steps. Female adult mosquito abundance is treated as the primary output of the model. When no vector control intervention is in action, the mosquito population is governed by the combined carrying capacities of all aquatic habitats, and the density-dependent oviposition mechanism, which limits the potential number of eggs that a female mosquito may preferentially lay in an aquatic habitat, considering both the associated CC and the biomass already present in the habitat.

In addition, the model makes the following assumptions:

- The male-female ratio of newly-laid eggs is assumed to be 1:1
- In the Mate Seeking stage, a female always finds a male mosquito to mate (within the first time-step it seeks to mate)
- In the Bloodmeal Seeking stage, a female always finds a bloodmeal (either immediately after mating, or after completing oviposition of an egg-batch)

3.7.5 Event-Action-List Diagram

We propose a new type of diagram, called the **Event-Action-List (EAL) Diagram**, to depict the simulation events (occurring on a daily basis), the corresponding actions triggered by those events, and the list(s) of agents (data structures) affected by them. The goal of an EAL Diagram is to capture the major daily events of a simulation in a standard, canonical manner. An EAL Diagram for our simulation model is shown in Figure 3.4.

Each event represents a biological phenomenon, and the corresponding action represents the programmatic task(s) performed by the simulation. Optionally, some
Figure 3.4. Event-Action-List (EAL) Diagram for the ABM. Each *squashed rectangle* represents an event-action pair, with the *event* at upper-half, and the *action* at lower-half of the *squashed rectangle*. Each *rectangle* represents the *list(s) (data structures)* of agents affected by the event-action pair.
list(s) of agents may be modified as a direct result of the performed action. Thus, an EAL Diagram can summarize the daily events of the simulation model by listing all major events, actions, and lists. For example, when the simulation is started, it needs to create some initial adult agents. The biological phenomenon “create initial adults”, termed as an event, is realized by the (simulation) action “add agents”; this event-action pair affects the list of adult agents, AL, in the simulation.

3.7.6 Vector Population Dynamics

Mosquito (vector) populations are dynamic in nature. The vector population dynamics is governed by three key operations performed at each simulation day, which are described in the following:

- Killing Adult Mosquitoes: To kill adult agents, we use a modified version of the logistic mortality model in which the age-dependent component of mortality increases exponentially with age (see Section 3.5). Newly emergent adults have a baseline daily mortality rate of $\alpha$. For each (adult) age-cohort, agents are killed with the rate of $ASMR_{Age(\text{adults})}$, which is calculated by Equation (3.4).

- Killing Immature Mosquitoes: Eggs and pupae are killed with a constant mortality rate $\alpha$ of 10%. Larvae mortality rate depends on both the age and the one-day old equivalent larval population, $N_e$. Within a given aquatic habitat, $N_e$ is first computed by Equation (3.6). For each (larva) age-cohort, agents are killed with the rate of $ASMR_{Age(\text{larvae})}$, which is calculated by Equation (3.7) using $N_e$ as well as the carrying capacity of the aquatic habitat ($CC$) and the rainfall coefficient ($R = 1.0$).

- Creating New Mosquitoes: When a Gravid female agent visits an aquatic habitat, it makes an attempt to oviposit. At most three such attempts can be made by an individual female (in a one-day period). The potential number of eggs a female can lay, $Eggs_{\text{potential}}$, depends on the current biomass in the habitat, and is calculated by Equation (3.10). The biomass of the habitat is calculated by Equation (3.9). Using $Eggs_{\text{potential}}$, the actual number of eggs allowed to lay ($Eggs_{\text{allowed}}$) is calculated (see Section 3.6). New mosquito agents are then created into the simulation in the form of eggs.

The order of major processing steps of the ABM captures the dependency relationships between the processing steps. These steps, performed on a daily basis, are
shown in Figure 3.5. Preservation of the dependency relationships is important to ensure the correctness of the model’s output, since alteration of these may drastically change the simulation results. To illustrate these relationships, the order of processing steps can also be viewed as a directed acyclic graph (DAG), and hence, a topological sort can be performed on it. In graph theory, a topological sort (or topological ordering) of a directed acyclic graph $G = (V, E)$ is a linear ordering of all its vertices such that if $G$ contains an edge $(u, v)$, then $u$ appears before $v$ in the ordering [46, p. 549]. A topological sort can be viewed as an ordering of vertices along a horizontal line so that all directed edges go from left to right. Every DAG will have one and possibly more topological sorts. Figure 3.6 depicts two possible topological sorts of the model’s daily processing steps.
Figure 3.5. The Order of Major Processing Steps in the ABM. During each simulated day, the ABM collects, saves, and prints output data (vector abundance, model parameters, etc.) to time-stamped output files, kills adult mosquitoes according to age-specific mortality rate $ASMR_{Age(\text{adults})}$, kills larvae according to age-specific mortality rate $ASMR_{Age(\text{larvae})}$, and kills eggs and pupae with fixed mortality rate of 0.1; it then updates the age, stage duration and stage transition for all agents (adults and aquatic). The dependency relationships between the processing steps are crucial to ensure the correctness of the model’s output.
Figure 3.6. Dependency Relationships in Ordering the Daily Processing Steps. The steps are depicted as directed dashed lines in (a). Two possible ordering that preserve the dependency relationships, are shown in (b) and (c). By topological sort, (c) can be arranged so that all directed edges go from left to right, as in (b).
A crucial issue in modeling the oviposition event arises regarding the *timing of update* of biomass of an aquatic habitat. As noted before, the actual number of eggs that a *Gravid* female agent can lay depends on the biomass. In a *static update approach*, the biomass is updated *once* in a one-day period, and *all* females that try to lay eggs the same day would sense (perceive) the same value of biomass. On the contrary, in a *dynamic update approach*, it is updated *every* time a female has laid eggs. While the latter has more intuitive appeal since it reflects the simulation world more accurately, the former has the advantage of eliminating any bias introduced to the female agents that appear later than others in the same habitat. For this reason, we choose to implement the *static* update approach.

3.8 Summary

In this chapter, we described the biological core model and the ABM. The core model primarily involves the vector population dynamics. We described the life-cycles of the vector *An. gambiae* and the parasite *Plasmodium*. We discussed vector mortality models and their basic mathematical formulation, showed the importance of incorporating senescence of vectors into the mortality models, and described the *age-specific mortality rates (ASMRs)* for all the adult stages and the larvae stage, as adopted in the core model. We also described temperature-dependent larval development, the aquatic habitats and houses, and the oviposition mechanism.

For the ABM, we described the class diagram, the model features, initialization, assumptions, and default model parameters. We also demonstrated the use of a new type of diagram, called the Event-Action-List (EAL) Diagram.

As mentioned before, the ABM described in this chapter is *non-spatial*. After discussing the verification & validation (V&V) processes for several non-spatial, language-specific ABMs (developed by individual researchers within our research group) in Chapter 4, we present a *spatial* extension in Chapter 5.
4.1 Overview

Assessing the credibility of complex simulation models can be challenging. In ABM, V&V techniques are used to determine that the model is an accurate representation of the real system. In order to improve the model, verification & validation (V&V) processes are iteratively performed by comparing the model to actual system behavior until model accuracy is judged to be acceptable.

In Chapter 3, we described the biological core model of the vector population dynamics, which is governed by the biology underlying the malaria vector *An. gambiae*. We also described the implementation details of the ABM, which is based on the described version of the core model. In our research group, several versions of two language-specific (Java and C++) implementations of the core model evolved over the past few years. V&V of these ABMs seemed necessary to ensure the correct outputs from the models. In this chapter, we describe the V&V processes performed on the ABMs, primarily by means of docking (a V&V mechanism).

*Verification* involves transformational accuracy of the model artifacts in its development, which can be performed by using a variety of testing techniques. The goal of verification is to ensure that the implementation is a correct realization of the conceptual model. It is often performed by debugging the model to ensure that

---

1 We acknowledge our colleagues Gregory J. Davis, Steve Kurtz, Ying Zhou, and James E. Gentile, who contributed to the docking process of the ABMs.

2 Results of this chapter appeared in Arifin et al. [6] and Arifin et al. [5].
it works correctly. In other words, verification involves building the model right \[18\]. Validation, on the other hand, involves substantiation that a model, within its domain of applicability, possesses a satisfactory range of accuracy consistent with the intended application of the model. A model is considered valid for a set of experimental conditions if its accuracy is within its acceptable range, which is the amount of accuracy required for the model’s intended purpose \[205\]. In other words, validation involves building the right model \[18\].

Docking, also known as alignment or replication, cross-model validation, or model-to-model comparison, is a V&V method that tries to align multiple models in order to investigate whether they yield similar results. Docking is useful to confirm whether the claimed results of a given simulation are reliable, and can be reproduced by someone starting from scratch. As noted in \[14\], without this confirmation, some published results may be simply mistaken due to programming errors, misrepresentation of what was actually simulated, or errors in analyzing or reporting the results. Docking can also be useful for testing the robustness of inferences from models, and to determine if one model can subsume another.

In this chapter, we describe the docking process between two language-specific implementations of the ABMs (both of which follow a fixed version of the core model, as described in Chapter \[3\]): the Java-based ABMs, which have different versions developed in different phases (hereafter referred to as J1, J2, J3), and the C++-based ABM. Both implementations simulate the life-cycle of \textit{An. gambiae} by tracking attributes relevant to the vector population dynamics for each individual mosquito. Details of the Java implementation is discussed in Chapter \[3\]. The C++ implementation, described in detail in \[263\], utilizes the GNU Scientific Library (GSL) \[69\]. This implementation is written to be portable, compiling and running on any platform for which an ANSI compliant C++ compiler is available. A major drawback of this implementation is the lack of built-in visualization tools. On the other hand,
a C++ implementation may execute faster and is better suited for parallelization in large-scale simulations.

We report how we obtained partial agreement between the agent-based implementations by describing results that dock the adult mosquito populations. Then, we describe further experiments in order to achieve a complete dock for both the adult and the aquatic populations between the ABMs.

The organization of this chapter is as follows: Section 4.2 discusses some of the previous works involving docking, replication, alignment or model-to-model comparison, explaining the context of our V&V work. Section 4.3 describes the partial docking process and its results, performed in different phases. Section 4.4 describes the complete docking process, which follows the Divide and Conquer paradigm, and the results. Section 4.5 concludes.

4.2 Literature Review

In this section, we mention some of the previous works involving V&V, paying special attention to docking, replication, alignment, or model-to-model comparison.

Ören et al. describe the V&V process performed on a simulation model of nuclear fuel waste management systems, providing a list comprising over 80 most common computational errors, and software tools and techniques used to detect and to correct those [178]. Axtell et al. [15] describe the docking process of Axelrod and Sugarscape models, and conclude that by comparing independently-built simulations using different tools, docking (or alignment) may discover bugs, misinterpretation of model specification, and inherent differences in toolkit implementations. Balci [17, 18] presents fifteen guiding principles for conducting verification, validation and accreditation (VV&A) of simulation models. Axelrod [14] discusses the importance of replicating simulations, and provides examples of the procedures and difficulties involved in the process of replication. Sargent [205] describes different approaches to
decide validity by two different paradigms that relate V&V to the model development process, with the use of graphical data statistical references for operational validity.

Carley [33] describes the importance of docking in computational social and organizational science. North and Macal [172] implement the Beer game using Mathematica programming, Repast and Swarm ABMS, reproducing (i.e., docking) all published results with these new implementations. Burton [31] argues that docking provides a guide in use of different laboratories to address organization questions, and computational and non-computational models can be docked to broaden the understanding of organization science. Xu et al. [256] discuss the results of docking a Repast simulation and a Java/Swarm simulation of four social network models of the Open Source Software (OSS) community.

Edmonds and Hales [58] replicate a published model involving co-operation between self-interested agents in two independent implementations to align the results and the conceptual design. The replication process reveals a host of minor bugs and ill-defined implementation issues that otherwise appeared unnoticed. They conclude that aligning models can be very difficult, but very revealing, because simply implementing simulations with respect to a conceptual model and then analyzing outputs for consistency with the conceptual model and data series is insufficient to ensure the correctness of an implementation.

By using model-to-model comparison (docking), Xiang et al. [255] demonstrate the V&V processes for a Natural Organic Matter (NOM) simulation model. Rand and Wilensky [194] present a case study that replicate the Axelrod-Hammond model involving the evolution of ethnocentrism, showing that aligning the order of events lowers the variance of the Wilensky-Rand model, causing the replicated model to be in statistical agreement with the original model. They further describe the challenges in recreating the model and in determining whether the replication is successful [247].

Yilmaz [258] presents a process-centric perspective for the V&V of agent-based
computational organization models. He emphasizes the importance of V&V in assessing not only the predictive capability, but also the explanation accuracy of formal models in terms of the degree of realism. Introducing the notion of social contracts to facilitate V&V of interaction constraints among agents, he presents a framework for the V&V of multi-agent organizations, and a set of formal validation metrics to substantiate the operational validity of emergent macro-level behavior.

Will and Hegselmann [249] show the importance of replication by reporting a failure to replicate the results presented on a published model. Obtaining the source code from the original authors, Will [248] reports that the simulation unintentionally implements an assumption that was never mentioned in the original model, and shows that the model crucially depended on that dubious assumption, and its removal leads to dramatically different results.

Pavón et al. [182] propose the use of agent-based graphical modelling languages to specify social systems as multi-agent systems, which allow replication of simulations on different platforms. Rouchier et al. [201] describe advancements in model-to-model analysis, and categorized comparative modelling research into a number of areas. Olaru et al. [176] report the docking experience and validation stages performed when replicating a fuzzy logic model with an agent-based model in innovation business networks.

This chapter serves as a case study for docking of our malaria entomology models, similar to the works done in other fields [15 172 176 255 256]. The novelty of this work, however, lies in demonstrating the importance of docking by showing that it can increase confidence to the biological core model, reveal conceptual and/or programming errors and eliminate dubious assumptions, reinforcing the findings in [33 58 194 248 249].
Figure 4.1. The Verification & Validation (V&V) Workflow for a Partial Dock (see Section 4.3). The unidirectional solid arrows indicate immediate successorship between the implementations, and successive iterations for $J3$ (the curved self-loop). The bidirectional solid arrow indicates Verification relationships between $J1$ & $C++$. The unidirectional dashed arrows indicate Validation relationships between the Core Model and the four implementations.

4.3 Obtaining a Partial Dock

In this section, we describe the partial docking process, performed in different phases during which the separate Java implementations are built, and the results.

4.3.1 Docking Phases

The partial docking process is performed in three different phases, as shown in Figure 4.1. The phases are outlined below:

1. Phase 1: The first Java implementation, $J1$, is built from the core model independently of the $C++$ implementation. Verification starts with comparing the $J1$ output against the $C++$ output. When compared (see Section 4.3.2), results show substantial differences between the two models ($J1$ & $C++$), necessitating further investigation between these two, and also validation against the core model.
2. **Phase 2**: Before delving too deep into seeking the source of differences, we develop J2, a simple clone of the C++ implementation. When compared, results show close agreement between the two models (J2 & C++). This phase, though seeming trivial, eliminates the source of any potential language-specific discrepancies and/or programming errors.

3. **Phase 3**: The goal of the final Java implementation, J3, is to ‘dock’ with the C++ implementation, and to validate these against the core model. This phase, being the most time-consuming, is iterative in nature.

4.3.2 Results

In this section, we compare output abundance graphs from different implementations, and discuss the findings obtained throughout the three different phases. Figure 4.2 shows the abundance graphs from J1 & C++. It points to the following major issues:

**Difference 4.3.1** *Both adult & aquatic populations differ dramatically (Figure 4.2)*

**Difference 4.3.2** *Proportion of female adults with an age of 12 or greater is significantly larger in J1 than C++ (Figure 4.2 (c))*

**Difference 4.3.3** *Aquatic population sizes are consistently less in J1 than C++ (Figure 4.2 (d), (e) & (f))*

Before analyzing these, we address some other minor issues. First, we verify that initial parameter settings, along with the constants used for the age-specific mortality functions, are identical for both models. Next, to check that the random number generators used by both simulations reliably generate numbers following their specified distributions (uniform & normal), we run a series of simulations on both models. To generate random numbers, Repast and GSL both implement the Mersenne Twister [154]. We log all randomly generated numbers with the specified distribution parameters used, and observe that proper distributions are being generated, ruling out any potential difference due to the random number generator libraries. Next, we ensure
Figure 4.2. Abundance Graphs for V&V Phase 1. The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance.
that floating point arithmetic is correctly computed in both models. For example, integer division may result in type coercion and rounding, causing cascaded loss of precision through the models leading to divergent results. To avoid this, we explicitly typecast all arithmetic involving a mix of decimals and integers to decimals.

To address Difference 4.3.1 we verify the age-specific mortality functions for the adults (as described by Equation (3.4)). Comparisons of the specific routines calculating $ASMR_{age(adults)}$ reveal that once calculated, the C++ model places an artificial bound to kill only 80% when it is to eliminate all members of a given age group. However, this modification, applied to J3, seems to have little impact on the results.

This step reveals another critical error. In J1, an agent enters the simulation with an age of 0. At each simulation day, the age of the surviving agent is increased by 1. This continues without any resetting of age when the agent enters the Immature Adult stage from its aquatic stage (see Figure 3.1). In C++, the age is reset to 0 once this transition occurs. This partially explains Difference 4.3.2, i.e., why a larger proportion (approximately half) of the female adult population is over 12 days old in J1 (see Figure 4.2 (c)). While resetting the age reduces the oscillation in the proportion of older females, it does not significantly raise the mean number of adult mosquitoes close to the C++ model.

To address Difference 4.3.3 we first note that the number of eggs in J1 is consistently less than C++, causing J1 to inject less larvae and pupae, and eventually less adults into the simulation. This, in conjunction with Difference 4.3.2, suggests that younger mosquitoes are being killed more rapidly than the older ones, a counterintuitive indication given the death rate equations, which are designed to kill the older mosquitoes at a higher rate.

In an attempt to eliminate Difference 4.3.3 we verify the age-specific mortality functions for the aquatic stages. This reveals that J1 & C++ implement the concept of $N_e$ (see Equation (3.6)) differently. The C++ model computes $N_e$ once, at the
start of each day, using the previous day’s aquatic populations. J1, on the other hand, recomputes $N_e$ for every oviposition event, creating a selection bias for female mosquitoes trying to lay eggs ahead of others within the same simulation day, and thus causing more repulsive force to the females trying later. This, when implemented, has some impact to increase the amount of eggs being laid in J3. This step also reveals a transition logic variance in C++ for egg development time: eggs take two days instead of one (see Table 3.1). When implemented, this reduces the difference in number of adults by approximately 500.

As evident from Figure 4.3, J3 & C++ are not completely docked, especially in terms of the aquatic populations. To achieve a complete dock, we perform further research by investigating other issues, which is described in the next section.

4.4 Obtaining a Complete Dock

In the previous section, we described the verification performed between two agent-based implementations (Java-based and C++-based), and the validation performed against the biological core model with respect to these implementations. The importance of rigorous docking was illustrated by the discovery of some incorrect assumptions and programming errors, which, being unnoticed, led to erroneous results. However, as evident from the results, the two implementations are not yet completely docked, especially in terms of the aquatic mosquito populations. In this section, we analyze the reasons, and perform further experiments to achieve a complete docking.

In particular, this section addresses the following issues:

- How the results from four agent-based implementations compare to each other and to the corresponding results from theoretical models
- How to verify the age-structure of mosquito populations and the age-specific mortality rates (for both adults and aquatic populations)
- How to verify the oviposition mechanism (see Section 3.6), and the calculation of one-day old equivalent larval population $N_e$ at each step of the simulation
Figure 4.3. Abundance Graphs for V&V Phase 3. The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance.
(see Equation (3.6)), by removing all randomness (e.g., with fixed number of eggs to lay, single aquatic habitat, etc.)

We show how to achieve a complete dock between multiple implementations, all of which are built on a sufficiently complex core model. It also highlights the importance of docking by showing that a successful dock may reveal conceptual and/or programming errors, and may eliminate dubious assumptions, reinforcing the findings in [33], [58], [194], [248], and [249].

4.4.1 The Divide and Conquer Paradigm

In most complex ABMs, synergies arising from seemingly insignificant differences in separate implementations may lead to significant mismatches in overall model outputs. Hence, it is crucial, and sometimes necessary, to compartmentalize the artificial simulation world, i.e., to separate it into isolated compartments so that errors in one specific compartment are not propagated and thus cannot influence the discovery (and correction) of errors in other compartments.

We compartmentalize the mosquito world with respect to the adult and aquatic populations, and perform some experiments to identify and fix the sources of output mismatches in order to achieve a complete dock. Following the Divide and Conquer paradigm, we ensure that the pieces are working as intended, and then combine them to perform some other, more complex experiments. Using four separate implementations that sprung from the simplified core malaria entomology model, we describe the design of these experiments, analyze their results, and show how they lead to a successful dock between all four implementations, and hence achieve a complete four-fold docking of the core model. The complete four-fold docking procedure encompasses verification between the four implementations, as well as validation against the core model with respect to these implementations.
Figure 4.4. The Core Model, Simplified from Figure 3.1, has a Single Aquatic Habitat. It is used in experiments $E1$, $E2$, and $E3$ (see Section 4.4).

4.4.2 The Simplified Core Model

For different experiments performed in this section, we use a simplified version of the core model (described in Chapter 3). For the purpose of docking, we make some simplifying assumptions, which may deviate from biological plausibility. The major difference, as shown in Figure 4.4, is to use a single (as opposed to multiple) aquatic habitat for the experiments.

The stage durations for mosquito agents, slightly re-defined from Table 3.1 are shown in Table 4.1. One important, though biologically unnatural, assumption is illustrated in the stage duration for the Gravid stage: the duration (in days) that the female mosquito is forced to remain as Gravid equals the number of oviposition attempts taken to successfully lay all the remaining eggs.
TABLE 4.1

STAGE TRANSITION TIMES (V&V)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Duration (in Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egg</td>
<td>1</td>
</tr>
<tr>
<td>Larva</td>
<td>Temperature-dependent; see Equation (3.1)</td>
</tr>
<tr>
<td>Pupa</td>
<td>1</td>
</tr>
<tr>
<td>Immature Adult</td>
<td>2</td>
</tr>
<tr>
<td>Mate Seeking</td>
<td>0 (immediate transition)</td>
</tr>
<tr>
<td>Bloodmeal Seeking</td>
<td>1</td>
</tr>
<tr>
<td>Bloodmeal Digesting</td>
<td>2</td>
</tr>
<tr>
<td>Gravid</td>
<td>Until all eggs are laid</td>
</tr>
</tbody>
</table>

The present model discretizes time such that every agent is updated during a time step representing a single day (24-hour period). For convenience, we show the major daily processing steps from Chapter 3 in Figure 4.5 (left).

4.4.3 Implementations of the ABMs

The four different implementations along with the V&V workflow, illustrating logical connections between the implementations and the core model, are shown in Figure 4.5 (on right). CPP1, described in detail in [203], is the earliest implementation of the core model. It is developed in the C++ programming language utilizing the GNU Scientific Library (GSL) [69]. This implementation is written to be portable, compiling and running on any platform for which an ANSI compliant C++ compiler is available. A major drawback of this implementation is the lack of built-in visualization tools. The first Java implementation, Java1, is built over three phases using
Figure 4.5. Verification & Validation (V&V) Workflow for Section 4.4.

The daily processing order of the simulation (on left; reproduced from Chapter 3), and the V&V workflow for a complete dock (on right). In the workflow, bidirectional arrows indicate verification relationships between the four implementations. The dashed arrow, between the re-factored implementations Java2 and CPP2, indicates internal verification, since these two are docked with respect to each other. Unidirectional, dashed arrows indicate validation relationships between the core model and the four implementations.
Java, as described in Section 4.3.

The re-factored implementations, CPP2 and Java2, reflect a unified architecture, and encode a mosquito’s life-cycle and behavior in a structure called a strategy. The strategy is flexible and can adapt to characterize new genus, species or variation within one species. The architecture is well-suited for parallelization across many cores or computers (see [71] for details). For the purpose of docking, CPP2 and Java2 are verified with respect to each other (internal verification), and a single output is compared with those of CPP1 and Java1.

4.4.4 Experiments and Results

In this section, we describe three docking experiments (labeled as E1, E2, and E3) and their results to achieve a complete docking. We assume the following:

- All randomness are removed from the ABMs while performing E1, E2, and E3. This allows direct model-to-model docking, and to compare the results against those of theoretical models.
- In all experiments, carrying capacity of the aquatic habitat is reduced to 3000.
- The Mate Seeking stage is omitted altogether for simplicity.
- The oviposition mechanism has been simplified.
- All males are omitted from all experiments to allow uniform killing of agents from separate age-groups.

In E1 and E2, following the Divide and Conquer paradigm, we compartmentalize the core model with respect to the adult and aquatic populations. E3 combines the adult and aquatic populations.

4.4.4.1 Isolating Adult Populations

Experiment E1 deals with the adult mosquito populations by isolating it from the aquatic mosquito populations. We use 2, 2, and 0 days as the stage durations
for Immature Adult, Bloodmeal Seeking, and Bloodmeal Digesting, respectively. The female stays in the Gravid stage until all eggs are laid, and transitions back to Bloodmeal Seeking the following day (see Figure 4.4). Starting with 100 female adults, $E1$ ensures that the age-structure and age-specific mortality rates (for the adult stages) match to the theoretical values. It also verifies the simplified oviposition mechanism.

4.4.4.2 Isolating Aquatic Populations

Experiment $E2$ deals with the aquatic mosquito populations by isolating it from the adult mosquito populations. Starting with 1000 female eggs that initially reside in the single aquatic habitat (with 3000 carrying capacity), it ensures that the age-structure of all aquatic stages, the age-specific mortality rates of larvae, and the base mortality rates of eggs and pupae populations match to the theoretical values. It also verifies the temperature-dependent larval development rate (see Equation (3.1)).

4.4.4.3 Combining Both Populations

Experiment $E3$ combines the adult and aquatic mosquito populations in a single aquatic habitat. Starting with 100 female adults and 1000 female eggs (initially residing in the single aquatic habitat with 3000 carrying capacity), it verifies the transitions from aquatic to adult (i.e. from Pupa stage to Immature Adult stage) population. It also checks the oviposition mechanism, including actual number of eggs laid in the aquatic habitat, and the stage durations (see Table 4.1).

$E3$ is performed in three separate phases as described below. After each phase, we compare the four-fold outputs, analyze and fix potential misinterpretations, and proceed to the next phase. All the issues discovered after analyzing the results in each phase are outlined in Table 4.2.
### TABLE 4.2

**DOCKING (V&V) ISSUES FOR SECTION 4.4**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Docking Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In <em>CPP1</em> and <em>Java1</em>, the adult populations are slightly less in number than those in <em>Java2</em> and <em>CPP2</em>. Also, the aquatic populations in <em>CPP1</em> and <em>Java1</em> are killed at a higher rate than suggested by the theoretical numbers.</td>
<td>Resolved after all implementations use the same carrying capacities for the aquatic habitat.</td>
</tr>
<tr>
<td>1</td>
<td><em>CPP1</em> and <em>Java1</em> have <em>Gravid</em> mosquitoes lay all of their eggs on the first oviposition attempt. This creates an egg-laying pattern where bursts of eggs are laid on the same day, followed by a few days when no eggs are laid, then another burst, and so on. In <em>Java2</em> and <em>CPP2</em>, however, the eggs are laid over a period of successive days.</td>
<td>This issue suggests a difference in oviposition code. It is partially resolved after all implementations ensure to use the same formulas for calculating how many eggs a female can lay in different oviposition attempts.</td>
</tr>
<tr>
<td>1</td>
<td>On the first oviposition attempt when eggs are laid, <em>Java1</em> and <em>CPP1</em> lay eggs one day sooner than <em>Java2</em> and <em>CPP2</em>.</td>
<td><em>CPP1</em> and <em>Java1</em> lay eggs at the end of the <em>Bloodmeal Digesting</em> stage, rather than waiting until the first day of being <em>Gravid</em>. They are updated to ensure that eggs are only laid while in the <em>Gravid</em> stage.</td>
</tr>
</tbody>
</table>
TABLE 4.2

Continued

<table>
<thead>
<tr>
<th>Phase</th>
<th>Docking Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In <em>Java1</em>, after all female mosquitoes lay all their (80) eggs in the <em>Gravid</em> stage, they do not transition back to the <em>Bloodmeal Seeking</em> stage on the same day.</td>
<td><em>Java1</em> updates to ensure that the females transition back to the <em>Bloodmeal Seeking</em> stage from the <em>Gravid</em> stage on the same day once all eggs are laid.</td>
</tr>
<tr>
<td>2</td>
<td><em>CPP1</em> and <em>Java1</em> enable the female mosquitoes to lay all (80) of their eggs on the first oviposition attempt at the beginning part of the simulation (on day 5 and 6). However, as suggested by Equation (3.10), laying all the eggs in a single oviposition attempt is possible only if the aquatic habitat is empty.</td>
<td><em>CPP1</em> and <em>Java1</em> find that the egg-laying is indeed complete after three attempts, and performs a simple adjustment in the delay before the female mosquitoes can leave the <em>Gravid</em> stage.</td>
</tr>
<tr>
<td>2</td>
<td><em>CPP2</em> and <em>Java2</em> place an upper bound of 80% on the larval mortality rate.</td>
<td><em>CPP1</em> and <em>Java1</em> ensure the same.</td>
</tr>
<tr>
<td>2</td>
<td>In the calculations of Equation (3.10), <em>CPP2</em> and <em>Java2</em> use <em>Eggs</em>(<em>{\text{remaining}}) (the number of eggs remaining to lay), instead of <em>Eggs</em>(</em>{\text{max}}).</td>
<td><em>CPP2</em> and <em>Java2</em> ensure to use <em>Eggs</em>(_{\text{max}}).</td>
</tr>
</tbody>
</table>
In *Java1*, *Gravid* females incorrectly lay all (80) eggs, and the biomass of the aquatic habitat does not affect the number of eggs actually laid. In the calculation of $Eggs_{laid}$ (the number of eggs allowed to lay), some double values are coerced to integer values, making the expression evaluating to 0 in these instances. This, in turn, affects the related variables ($Eggs_{potential}$, $Eggs_{laid}$, and $Eggs_{remaining}$). *Java1* updates this by using explicit typecasting to double values.

After 14 simulation days, *CPP1* and *Java1* kill a different number of adult mosquitoes, suggesting a rounding error. *Java1* uses an extra ‘$a$’ in the adult age-specific mortality rate function (Equation 3.4), and omits it to match the correct equation.
Figure 4.6. Four-Fold Docking Results of Vector Abundance, for $E_1$, $E_2$, and $E_3$, Performed in Phase 1. The graph on left shows the total number of (female) adults, and the graph on right shows the total number of agents (adults and aquatic). The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance. For other details, see Section [4.4].

Figures 4.6–4.10 show the results. After the issues found in Phase 2 are addressed, we analyze the results again and discover a single issue. Once it is addressed, the results produce a complete four-fold dock, as shown in Figure 4.10.

4.5 Summary

In this chapter, we described the docking process of several independently-developed ABMs derived from one particular version of the biological core model. Though we employed other V&V methods (e.g., face-validation, input-output transformations, internal consistency checks, etc.), in this chapter, we discussed only the docking method for V&V. We showed how to achieve a partial dock in terms of the adult mosquito populations, and then performed further experiments to achieve a complete dock in terms of the adult and aquatic populations. We also described how the major findings (docking issues and their resolutions) helped to clarify concepts, eliminate ambiguities, and revealing semantic errors, by identifying differences in different phases of the ABMs.

Docking helped in the verification process between the ABMs, and in the valida-
Figure 4.7. Four-Fold Docking Results of Aquatic Biomass, for E1, E2, and E3, Performed in Phase 1. The graph on left shows the One-day Old Equivalent Larval Population, $N_e$ (see Section 3.7.6). The graph on right shows the total biomass in the aquatic habitat. The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance. For other details, see Section 4.4.

Figure 4.8. Four-Fold Docking Results of Abundance for E3 in Phase 2. The graph on left shows the total number of (female) adults, and the graph on right shows the total number of agents (adults and aquatic). The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance. For other details, see Section 4.4.
Figure 4.9. Four-Fold Docking Results of Aquatic Biomass for E3 in Phase 2. The graph on left shows the One-day Old Equivalent Larval Population, $N_e$ (see Section 3.7.6). The graph on right shows the total biomass in the aquatic habitat. The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance. For other details, see Section 4.4.

Figure 4.10. Four-Fold Docking Results for E3 in Phase 3. The graph on left shows the total number of agents (adults and aquatic). The graph on right shows the biomass in the aquatic habitat. The $x$-axis denotes simulation days, and the $y$-axis denotes mosquito abundance. The complete overlap between the results of four implementations indicates a complete four-fold dock. For other details, see Section 4.4.
tion process against the core model, by identifying differences in model specification, interpretation, implementation and enhancement phases, and revealing semantic errors. The importance of rigorous docking was illustrated by the discovery of some incorrect assumptions and programming errors, which, being unnoticed, led to erroneous results. Our results also showed that synergies arising from seemingly insignificant differences in separate implementations may lead to significant mismatches in overall model outputs, suggesting that the entire docking process should be iterative in nature, and should involve well-planned feedback from earlier implementations. As the core model is continually being refined, and new features are added to it, the need to verify all implementations, and to validate these against the core model, warrants a continuous V&V process to be carried on. The entire docking process serves the dual purpose of increasing confidence to the core model, and revealing conceptual errors in implementations of the ABMs.

This chapter may also serve as a case study to illustrate the importance of docking complex simulations, since even the best simulation programs may imply dubious assumptions, leading to erroneous results [248]. To be specific, our docking exercise demonstrates a detailed case study for verifying and validating multiple simulation models, all of which originated from the same biological core model.

The next chapter (Chapter 5) describes a spatial extension of the ABM.
CHAPTER 5

A SPATIAL EXTENSION OF THE ABM

5.1 Overview

In Chapter 3, we described the core model which is governed by the biology underlying An. gambiae vectors, and the agent-based implementation derived from the core model. The ABM, as mentioned before, was non-spatial: none of the agents and/or the environments possessed any spatial attributes. In this chapter, we describe a spatial extension of the ABM, aided by a landscape generator tool.

An ABM can be applied to a domain with or without an explicit representation of space. In some cases, however, an explicit spatial representation may be required for certain aspects of the ABM to be modeled more realistically. For example, in a spatial ABM of malaria, events like obtaining a successful bloodmeal (host-seeking), or finding an aquatic habitat to lay eggs (oviposition), can be modeled by utilizing the distribution of corresponding resources in the landscape.

In general, analysis of spatial relationships is fundamental to epidemiology research, as demonstrated by several studies. For example, in a large-scale longitudinal study of malaria in a 70 km² area in Siaya district in western Kenya, Hightower et al. perform spatial analysis and examine spatial hypotheses by using differential global positioning system (DGPS) to produce maps with highly accurate locational information for all of the geographic features of interest [98]. In the discipline of spatial epidemiology (also known as landscape epidemiology), the notion of spatial

\footnote{Major portions of this chapter appeared in Arifin et al. [8] and Arifin et al. [9], and as a standalone chapter in Zhang [260, Chapter 14].}
heterogeneity is considered as one of the most important factors for an effective representation of the environment being modeled. In most cases, the probability of disease transmission significantly declines with distance from an infected host. Thus, the spatial locations of pathogens, hosts and vectors are fundamentally important to disease dynamics [179]. For example, Benavides et al. show that the spread of infectious diseases in wildlife populations is influenced by patterns of between-host contacts: the habitat hotspots (places attracting a large numbers of individuals or social groups) can significantly alter contact patterns and, hence, disease propagation [23].

In modeling malaria with ABMs, representation of space may be crucial [86, 85, 160]. The prevailing large within-site spatial variations in the abundance and temporal dynamics of malaria vector populations indicates that the risk of parasite transmission differs among sites [105]. Several studies have reported large spatio-temporal variations in commonly used assessment metrics such as mosquito density, sporozoite rate, and entomological inoculation rate (EIR) [220, 138, 55, 52, 2]. Examples of local variations may include locations of aquatic habitats and bloodmeal events, characteristics of mosquitoes, etc. For malaria models, space can be represented as mosquito world, aquatic habitats, etc. for the mosquito agents; and as houses, huts, etc. for the human agents.

In our malaria ABM, some events (e.g., host-seeking, oviposition) by nature require spatial attributes. The underlying spatial heterogeneity defines the spatial distribution of resources, and controls how easily adult female mosquitoes may find resources that are necessary to complete their gonotrophic cycle (the cycle of obtaining bloodmeals and ovipositing eggs). This, in turn, directly affects the mosquito population in the ABM [8, 9]. In the previous non-spatial model, the resource-seeking events are modeled with separate probability distributions (to account for travel and search times incurred by adult female mosquitoes), simply because the ABM does
not have any explicit spatial representation [263, 71, 9]. For example, host-seeking and oviposition events are modeled with 25% probability of success in each hour of searching (the value of 25% is chosen as a baseline and not meant to be absolute). A spatial model, however, provides opportunities to model these events by coupling them with the corresponding locations of resources in the landscape. This concept can be generalized whenever an agent needs to seek for a resource.

As we mentioned in Section 2.5.2.1, malaria models can also be classified into non-spatial and spatial models, considering the form of the spatial representation embedded in the models, and this classification may overlap with both categories of mathematical and agent-based models. In spatial models, space can be represented in a variety of ways. For example, Bian categorizes grid and patch as the two fundamentally different data models to represent space [24]. A grid consists of a finite number of regular cells. In the patch model, space is partitioned according to landscape features (e.g., patches, corridors, and nodes). In our spatial extension, we represent space with a discrete, finite-sized grid model.

In this chapter, we describe a spatial extension of the previous ABM. Though in both the non-spatial and the spatial models, all mosquito agents are represented individually, in the new spatial model, the agents also possess explicit spatial information. We show how the previous model and the current spatial model yield consistent results with identical parameter settings (whenever applicable), and hence are docked. We also show how spatial heterogeneity affects some results in the spatial model. Another primary goal of this chapter is model verification, which is achieved in part by docking of the non-spatial and spatial models. In Chapter 4, we described the verification & validation (V&V) processes performed on the previous non-spatial ABMs, primarily by means of docking, which is a form of V&V that tries to align multiple models. In this chapter, we show that docking significantly helps in model verification between both model types (non-spatial and spatial).
The organization of this chapter is as follows: In Section 5.2, we describe the spatial ABM in detail, including the mosquito agents and their spatial movement, the landscapes in which the agents move, the modeling aspects of the resource-seeking events, etc. Section 5.3 describes two versions of our landscape generator tools. Section 5.4 discusses the results of model verification, the effects of varying the landscape patterns, the relative size and density of the aquatic habitats, and the overall capacity of the system. In Section 5.5, we demonstrate the effects of spatial heterogeneity of the landscapes, by considering the resource density.

5.2 The Spatial ABM

We begin by describing some relevant terms. Then, we briefly mention the non-spatial ABM, and discuss the spatial ABM in detail.

5.2.1 Definition of Terms

In the following, we describe some terms that are relevant to the spatial ABM, and are used throughout the rest of this chapter. Some of these terms were first introduced in Chapter 3, and reproduced here for easy reference.

- **Biomass**: the biomass of an aquatic habitat (see Section 3.6) is the sum of the number of eggs $Egshabitat$, the one-day old equivalent larval population $Ne$, and the number of pupae $Pupae_{habitat}$, as we describe in Equation (3.9):

  $$\text{biomass} = Egshabitat + Ne + Pupae_{habitat}$$

- **Combined Carrying Capacity (CCC)**: for a given landscape with one or more aquatic habitats, the combined carrying capacity $CCC$ is the sum of the $CC$s of all aquatic habitats (recall from Section 3.6 that the carrying capacity $CC$ is a soft limit on the aquatic mosquito population that the aquatic habitat can sustain); since the combined aquatic population, controlled by $CC$ in each aquatic habitat, eventually limits the overall mosquito abundance, $CCC$ effectively represents the overall capacity of the landscape (in terms of the mosquito population).
• **Resource Density**: for a given landscape with multiple resources, resource density of a particular resource-type is defined as the percentage of total area in the landscape occupied by objects of that resource-type. For example, considering aquatic habitats as the resource-type, a $5 \times 10$ landscape with 25 aquatic habitats (AHs) have 50% AH-Density.

• **Average Travel Time (ATT)**: for a given landscape, the average travel time (ATT) is defined as the average time (in hours) taken by an adult female mosquito to successfully find a resource (of a particular resource-type). In this study, though we do not explicitly measure ATT, it reflects a statistically-expected measure in our spatial analyses (as described later).

• **Oviposition**: as described in Section 3.6, oviposition is the process by which gravid females lay new eggs; referring to Equation (3.10), when an adult female finds an aquatic habitat to lay eggs, the potential number of eggs, $E_{\text{potential}}$, that it may preferentially lay, is regulated by both the carrying capacity $CC$ and the biomass already present in the habitat:

$$E_{\text{potential}} = E_{\text{max}} \left( 1 - \frac{\text{biomass}}{i \times CC} \right)$$

where $E_{\text{max}}$ is the maximum number of eggs available to lay, $N_e$ is the one-day old equivalent larval population of the habitat (see Equation (3.6)), and $i$ is the oviposition attempt number (i.e., 1, 2, or 3).

• **Vector Abundance (VA)**: at any specific point in simulated time, VA denotes the total number of adult female mosquito (vector) agents in the ABM; for all simulations in this chapter, we consider VA as the primary output of interest; for each simulation time-step (measured in hours), we keep track of VA by counting the aggregate total of adult female agents across every cell in the landscape.

In the non-spatial ABM (described in detail in Section 3.7), the resources (human and aquatic habitats) and the agents (mosquitoes) had no spatial locations. Adult female mosquitoes found resources at random. To model this random behavior, we used uniform statistical distributions generated by Repast pseudo-random numbers generator library [196]. All simulation events, including emergence, host-seeking, and oviposition, occurred without any spatial context. For example, oviposition was probability-based: during each (simulation) hour, a Gravid female tried to find an aquatic habitat with 25% probability of success. As long as the female had remaining
eggs to lay, it was allowed to make at most 3 attempts per 12 hours (i.e., each night). This, in turn, translated to 25% chance of finding an aquatic habitat per hour.

As mentioned before, the spatial ABM provides an extension in space of the previous non-spatial ABM. Figure 5.1 depicts a simplified class diagram for the spatial ABM, extended from the previous class diagram (see Figure 3.3) for the non-spatial ABM, illustrating its four major classes and attributes. The main differences between the two diagrams, as marked in bold in Figure 5.1, are the addition of spatial locations to all habitats and agents. Each location is encoded as a 2D point, representing a location in the \((x, y)\) coordinate space.

As before, an instance of the AnophModel class creates all other required class instances, initiates and runs the simulation, keeps track of the simulation time, instantiates initial mosquito agents, and logs outputs to time-stamped output text files. In addition, it keeps track of the global environment (which, in turn, keeps track of all spatial habitats), and assigns the initial mosquito agents to different spatial locations. The mosquito agents are created as instances of the MosquitoAgent class, supplied with their initial spatial coordinates. Instances of the AquaticHabitat and HumanHabitat classes represent both resource-types: aquatic habitats and bloodmeal locations (houses), respectively. Each instance has its spatial location and other attributes (e.g., carrying capacity of an aquatic habitat, number of persons living in a house, etc.).

5.2.2 Landscapes

In our spatial ABM, a landscape is used to represent the coordinate space necessary for the spatial locations and movement of adult female mosquito agents. Resources, in the forms of human habitats (houses) and aquatic habitats, are contained within a landscape. The density and spatial distribution of both types of resources inherently define the spatial heterogeneity of resources within the landscape.
Figure 5.1. Class Diagram for the Spatial ABM. The spatial ABM, extended from the previous non-spatial ABM (see Figure 3.3) consists of four major classes: AnophModel, AquaticHabitat, HumanHabitat, and MosquitoAgent. An instance of the AnophModel class creates all other required class instances, initiates and runs the simulation, and logs outputs to time-stamped output text files. In addition, it keeps track of the global environment (which, in turn, keeps track of all spatial habitats), and assigns the initial mosquito agents to different spatial locations. It can have one or more instances of AquaticHabitat and HumanHabitat classes. Each habitat instance has its spatial location and other attributes. The aquatic and human habitats instances are accessed through aquaticHabitatList and humanHabitatList, respectively. The mosquito agents are created as instances of the MosquitoAgent class, and can be accessed through agentLists. In this figure, only the major attributes and operations of the classes are shown. The addition of the location attribute, as instance of a Point2D class, allows the ABM to implement the spatial properties for the AquaticHabitat, HumanHabitat, and MosquitoAgent classes.
Denoted by its dimensions \( m \times n \), a landscape is defined as a collection of \( m \) horizontal rows and \( n \) vertical columns, and has \( m \times n \) cells. Each cell, with its spatial attributes, may represent a specific habitat (human or aquatic), or be part of the (adult) mosquito environment. Each landscapes is modeled with a non-absorbing boundary, modeled topologically as 2D torus spaces (in Chapter 6, we describe and use other types of boundaries). With a non-absorbing boundary, when mosquitoes hit an edge, they re-enter the landscape from the edge directly opposite of the exiting edge (and thus are not killed due to hitting the edge). For simplicity, we assume that each resource can occupy exactly one cell in a landscape. Resource density of a landscape is regulated by its dimensions and the number of resources it contains. For example, considering aquatic habitats (AHs), placing 25 AHs in a 10 \( \times \) 10 landscape produces 25\% AH-Density; placing the same number of AHs in a 5 \( \times \) 10 landscape produces 50\% AH-Density. In this chapter, we report results that use three types of landscapes:

- **Regular**: the spatial distribution of resources within the entire landscape follows a regular, well-defined pattern; every (non-empty) row and column, which has resources, contains the same number of them; horizontal and vertical distances between any two neighboring resources always remain the same.

- **Random**: resources are placed randomly following a uniform random distribution.

- **Hybrid**: a blend of the previous two; every (non-empty) row contains the same number of resources; within a (non-empty) row, however, they are placed randomly, thus allowing control to both density and randomness.

Examples of the three types of landscapes are depicted in Figure 5.2.

\(^2\)The landscapes reported in this chapter are generated using our landscape generator tool, which is described in Section 5.3.
Figure 5.2. Examples of Three Types of Landscapes used for the Spatial ABM. Each blue rectangle represents a spatial resource (aquatic or human habitat), and each white (empty) rectangle represents a cell in the mosquito environment. Sub-figures (a), (b), and (c) depict examples of regular, random, and hybrid landscapes, respectively. Each landscape has a dimension of $4 \times 8$. (a) and (b) has 25% resource density each; (c) has 37.5% resource density.

5.2.3 Movement of Agents

The primary reason for movement of adult female mosquito agents is to seek for resources (human habitats or aquatic habitats) in order to complete the gonotrophic cycle. As empirical data indicates limited flight ranges and sensory perception of mosquitoes [75, 73, 76, 74, 162], we limit the speed of movement to one cell per hour, and the perception range to be the eight adjacent (neighboring) cells of the focal/current cell (in effect, implementing a Moore neighborhood).

In some studies, the mosquito foraging behavior (i.e., the search for hosts and oviposition sites) has been modeled using a combination of methods (rather than using a pure random walk). For example, Gu and Novak modeled foraging as a two-stage process: a random flight when the resource site is not within the mosquitoes perception range, and a directional flight to the resource site once it is detected [86].

In our spatial ABM, movement of adult female mosquitoes is restricted: to seek for resources, they move only when in Bloodmeal Seeking or Gravid stages (these stages were marked in red in Figure 3.1). New agents, in the form of eggs, possess the same spatial location as that of the aquatic habitat in which they are oviposited.
We model the resource-seeking process with random, non-directional flights with limited flight ability and perceptual ranges until they can perceive resources at close proximity, at which point, the flight becomes directional. At any point in the resource-seeking process, a mosquito’s neighborhood is modeled as an eight-directional Moore neighborhood. If the current cell and its neighborhood do not contain any resource, the mosquito starts a random flight and moves randomly into one of the adjacent eight cells (like [86], we also set the probability of moving into a diagonally-adjacent cell as half that of moving into a horizontally- or vertically-adjacent cell). If, on the other hand, it perceives a resource in the neighborhood, it flies directly to the cell containing the resource.

In the Bloodmeal Seeking stage, the mosquito looks for human houses, and the search continues until it successfully finds a house. In the Gravid stage, the mosquito looks for an aquatic habitat, and once found, lays the eggs. The number of eggs it can lay is governed by the density-dependent oviposition rules (see Section 3.6 for details). If all of the eggs are laid, it goes to the Bloodmeal Seeking stage again, initiating a new gonotrophic cycle. Otherwise, it either remains in the same aquatic habitat or searches for another one to lay the remaining eggs, and this process continues until all eggs are laid. The movement activities for both stages are depicted as logical flowcharts in Figure 5.3.

5.2.4 Resource-Seeking Events

The resource-seeking behavior of adult female mosquito agents primarily encompasses two frequent events in our ABMs: host-seeking and oviposition. One primary purpose of this study is to spatially model these resource-seeking events, and then to compare results from different models. To model these events, we replace probability-based measures by spatial, distance-based measures. To be specific, the 25% probability of finding a resource site (in the non-spatial model) is replaced by specifying
Figure 5.3. Mosquito Movement during Resource-Seeking. In the spatial ABM, female mosquitoes move in Bloodmeal Seeking or Gravid stages in order to seek resources (houses or aquatic habitats, respectively) in an eight-directional Moore neighborhood. If a resource is perceived in the neighborhood, the mosquito directly moves to it in the next time-step. Otherwise, it moves in random direction to an adjacent cell.
the adult female mosquitos speed of movement as well as the density distributions of the resources in the corresponding landscape.

We pay special attention in modeling oviposition. As we mentioned in Chapter 2, oviposition is one potential factor explaining heterogeneous biting and vector distribution in a landscape with a heterogeneous distribution of habitats; female mosquitoes tend to aggregate around places where they oviposit, thereby increasing the risk of malaria there, regardless of the suitability of the habitat for larval development [160]. Thus, even if an aquatic habitat is unsuitable for mosquito emergence, it can be a significant source for malaria.

5.2.5 Simulations: Assumptions and Notation

In all simulation runs reported in this chapter, we assume the following convention (for details about the terminology, see Section 5.2.1): $AH$ denotes an aquatic habitat, $BML$ denotes a bloodmeal location (human habitat), $CC$ denotes individual carrying capacity of an aquatic habitat, $CCC$ denotes the combined carrying capacity, and $VA$ denotes vector abundance. The x-axis denotes simulation time (in days), and the y-axis denotes $VA$. In describing figures, we refer to a specific graph by the legend used (e.g., Non-Spatial, 25%). All simulations are run for at least 365 days (1-year) with 1000 initial adult mosquitoes (500 males and 500 females), and no initial eggs in any $AH$. In most cases, we present only relevant portions of the 1-year results. Male-female ratio of new mosquitoes is assumed to be 1 : 1.

5.3 Landscape Generator Tools

To facilitate the specification of parameters to the spatial ABM, we have built a landscape generator tool as a graphical user interface (GUI). Figure 5.4 depicts a screenshot of an earlier version of the tool (named AnophGUI). All landscapes used in this chapter are generated by AnophGUI.
Figure 5.4. Screenshot of an Early Version of the Landscape Generator Tool, AnophGUI. Sub-figure (a) shows a screenshot of the GUI. Sub-figure (b) shows the generated landscape. In (b), each black circle represents an aquatic habitat, and each gray rectangle represents a human habitat. For both resource-types (aquatic and human habitats), the same number of resource objects is generated. This example resembles a landscape with contiguous water bodies (e.g., a marsh or wetland) in the center, surrounded by human habitats at random distances from the water bodies.
For a specific landscape composed of hundreds of resources, AnophGUI automates the task of generating spatial attributes (e.g., location, capacity, etc.) for these resources, and allows the user to specify other simulation parameters (e.g., length of run), as well as the relevant weather parameters (e.g., temperature). The user may also select different landscapes, and modify the spatial attributes of a resource. Once the user selects a particular landscape, the resource table, as shown at the bottom-right corner of Figure 5.4 (a), is automatically populated, depending on the number of resources and the size of the landscape.

Later, we improved the tool to allow us to generate landscapes with varying spatial heterogeneity of both types of resources (aquatic and human habitats). Figure 5.5 depicts a screenshot of the latest version of the tool (renamed as VectorLand). VectorLand also allows us to control the locations of resources using the Clustering sliders across both axes, and the various vector control intervention parameters (see Chapter 6).

We emphasize that AnophGUI and VectorLand are tools to generate landscapes which are then used as spatial inputs to the ABM; and are not treated as models in themselves.

5.4 Results

In this section, we report the results of several experiments, all of which compare outputs (vector abundance, VA) from the ABMs in different settings. We first perform model verification by means of docking\(^3\) in Section 5.4.1, we compare the results from the non-spatial and spatial ABMs by modeling one of the resource-seeking events, oviposition, in two different ways. In Section 5.4.2, using the spatial ABM, we compare VA with two different landscapes: regular and random. Sections 5.4.3 and

\(^3\)For details about the docking process between several implementations of the non-spatial ABMs, see Chapter 4.
Figure 5.5. Screenshot of the Latest Version of the Landscape Generator Tool, VectorLand. VectorLand can generate landscapes with varying spatial heterogeneity of both types of resources: aquatic habitats and houses. Locations of resources can be controlled using the Clustering sliders across both axes. Intervention parameters can be controlled using separate panels (currently for LSM and ITNs). This screenshot depicts selecting medium house density, with 30% LSM coverage and 50% ITN coverage (see Chapter 6 for details on coverages). Additional statistics about the generated landscape and legends are also shown in separate panels.
5.4.4 present results of varying the relative size and density of the aquatic habitats, respectively, with landscapes that all have the same $CCC$. In 5.4.5, we report results of varying the overall capacity of the system by gradually increasing $CCC$.

5.4.1 Model Verification

In this section, we report the results of model verification by comparing outputs of the non-spatial model to the spatial model. In Chapter 4, we presented a three-phase docking process that produced agreement in models’ outputs. It served the dual purpose of increasing confidence to the core model and revealing errors in implementations of the ABMs. Following the Divide and Conquer paradigm, we also showed how to obtain a successful dock between four separate implementations that sprung from the same core model.

The process of model verification, as described in Chapter 4, requires substantial evaluations and re-evaluations of the non-spatial and the spatial ABMs. As we perform iterative refinements in several phases, however, the ABMs successively produce more similar outputs, which are in increasing agreement among themselves. Hence, the close agreement in model verification results, between the non-spatial and spatial ABMs, and between different but comparable scenarios of the spatial ABM itself, reflects successful cases of model verification performed on the ABMs. This also rules out the effects of any potential inadvertent biases which may have been introduced by poor experimental or setup design in obtaining similar outputs from the ABMs.

In the non-spatial ABM, oviposition is modeled with 25% probability of success in each hour of searching (as mentioned before, the value of 25% is chosen as a baseline, and not meant to be absolute). The spatial ABM, on the other hand, replaces this by specifying the precise movement rules (see Section 5.2.3) and the speed of movement of the adult female mosquito, as well as the density distributions of the habitats in the corresponding landscape. Figure 5.6 shows that despite modeling oviposition in
Figure 5.6. Model Verification Results. Both the non-spatial and the spatial ABMs yield consistent results with identical parameter settings. We show the results of modeling oviposition in two different ways: probability-based in the non-spatial model, and distance-based in the spatial model. 25% AH-Density is used in both cases. Each graph represents the average of 40 simulation runs, and shows only relevant portions of the one-year simulation results.

two different ways, both ABMs yield consistent results (with identical settings for other parameters).

5.4.2 Landscape Patterns

For the spatial ABM, we compare VA using two different landscapes: regular and random. We use the same number (100) and density (25%) of AHs, and the same CCC, for both landscapes. As shown in Figure 5.7 in these settings, the use of different landscape patterns does not significantly affect the mean (stabilized) abundance (≈ 8000).

It should be noted that in this experiment, due to the use of relatively small landscapes (all having dimensions 20 × 20), and relatively high resource densities (with 25% densities of aquatic habitats), we obtained statistically averaged similarities between different landscape patterns (i.e., regular and random). However, with relatively larger landscapes having relatively lower resource densities, this may change
due to the different (more sparse) distributions of the resources. As a result, the spatial heterogeneity may yield radically different abundances in those larger landscapes. Consider, for example, a 100 $\times$ 100 random landscape with only 5% $AH$-Density (i.e., with 500 aquatic habitats), in which the landscape contains some isolated aquatic habitats having none or very few human habitats within their proximity. In this case, $VA_{\text{random}}$ is expected to be much lower than $VA_{\text{regular}}$, since, in the random landscape, the female agents, originating from isolated aquatic habitats, have to travel much longer to find houses (and thus to complete their gonotrophic cycles).

5.4.3 Relative Sizes of Resources

For both the non-spatial and spatial ABMs, we vary the relative sizes of the $AH$s in landscapes that all have the same $CCC$. Recall that in the core model, the size of an $AH$ can be approximated by its carrying capacity $CC$ (see Section 3.6). For the spatial ABM, we use landscapes composed of one large $AH$ vs. many smaller $AH$s. To construct different cases, the numbers of $AH$s are increased as squares.
Figure 5.8. Results for Resource Size Variation, Part 1. For both the non-spatial and the spatial models, the numbers of $AH$s are increased as squares of the first 10 integers (1 through 10). Here, we show $VA$ with landscapes having 1 and 16 $AH$s. Once $CCC$ is kept constant, multiple $AH$s do not affect the results.

of the first 10 integers (1 through 10). In all cases, for both models, we use the same $AH$-Density. Figure 5.8 compares the results for 1 and 16 $AH$s, and Figure 5.9 compares the results for 49 and 100 $AH$s, as obtained by using the non-spatial and the spatial ABMs (other results show similar trends). As evident from the mean $VA$ level ($\approx 8000$), once $CCC$ is kept constant, $VA$ does not change significantly if the number of $AH$s is varied. Also, in all cases, a larger $AH$ can produce $n$ times adult population than the (individual) adult population produced by $n$ smaller $AH$s.

In Figure 5.9 we also find that until the systems reach equilibrium at around day 60, the non-spatial ABM ($VA_{Non-spatial,49AH}$ and $VA_{Non-spatial,100AH}$) always have equal or higher abundances than the spatial ABM ($VA_{Spatial,49AH}$ and $VA_{Spatial,100AH}$). We attribute this to the travel time required by adult female agents in the spatial ABM. Since the agents need to search for resource sites, and hence incur additional time delays to complete the host-seeking and oviposition events, the spatial ABM produces less abundances than the non-spatial ABM (in which agents do not require
to travel in space to complete these events, as explained before). After equilibrium, the \( AHs \) gradually become more full by new aquatic agents. Thus, governed by Equations (3.6), (3.9), and (3.10), the competition perceived by the *Gravid* females in an \( AH \) to find a chance to lay their eggs increases (as the biomass of the \( AH \) successively increases, \( Eggs_{potential} \) decreases). However, this does not impact the overall abundance of the system, which is already in equilibrium and hence has a steady, saturated flow of newly-emerged adult agents from the \( AHs \). Thus, after equilibrium, the models yield similar abundances.

5.4.4 Resource Density

For the spatial ABM, we also consider the density of a single resource-type: the aquatic habitat. We compare \( VA \) with varying number of \( AHs \) in \( 10 \times 10 \) hybrid landscapes. In this setting, as the number of \( AHs \) increases in a landscape, so does its \( AH\)-Density. In all cases, we use the same \( CCC \) of 100\( K \). The numbers of \( AHs \)
Figure 5.10. Results for Resource Density Variation. Using 10 × 10 hybrid landscapes, we use varying AH-Densities of 10%, 30%, 50%, and 80%, by increasing the number of AHs within the same landscape. Once CCC is kept constant (100K), varying the AH-Density only slightly increases the mean population. Each graph represents the average of 40 simulation runs, and shows only relevant portions of the one-year simulation results.

are increased from 10 to 80, in increments of 10. Figure 5.10 shows the results of varying resource density: for 10%, 30%, 50%, and 80% AH-Density cases, with the same CCC, varying the AH-Density does not significantly affect the mean population (other cases show similar trends).

However, two interesting observations are made:

- Until the population reaches equilibrium (at around day 60), \( VA_{10\%} \) is less than \( VA_{30\%}, VA_{50\%} \) and \( VA_{80\%} \): due to lower AH-Density, \( ATT \) (Average Travel Time) increases, and becomes a limiting factor. For example, on the average, in \( VA_{10\%} \), a Gravid female agent has to search larger number of cells to find a resource than in the other three higher-density cases. Thus, in \( VA_{10\%} \), less number of females may find a chance to lay eggs.

- After equilibrium, \( VA_{30\%} > VA_{50\%} > VA_{80\%} \): as AH-Density increases, \( CC \) (per AH) and \( ATT \) decrease. However, with increasing AH-Density, even though Gravid females find opportunities to visit AHs more frequently, they are more restricted to lay eggs, as regulated by Equation (3.10). In this case, \( ATT \) no longer being a limiting factor in successfully finding a habitat, the smaller capacities (CCs) of larger number of AHs (i.e., in higher AH-Density cases) dominate in restricting the abundances.
Figure 5.11. Results for System Capacity Variation. Using $10 \times 10$ hybrid landscapes, we vary the system capacity ($CCC$). As $CCC$ is gradually increased from $10K$ to $80K$, the mean population ($VA$) increases at a steady rate. Each graph represents the average of 40 simulation runs, and shows only relevant portions of the one-year simulation results.

5.4.5 Combined Carrying Capacity ($CCC$)

For the spatial ABM, we vary $CCC$ with same-sized $AH$s in $10 \times 10$ random landscapes. In this setting, the $AH$-Density remains the same (25%) across all landscapes. As the results indicate in Figure 5.11, the overall capacity of the system, as represented by $CCC$, indeed drives abundance: in all cases, as $CCC$ is gradually increased from $10K$ to $80K$, mean $VA$ increases at a steady rate.

It is worth noting that the first few dips seen in adult female populations, as seen in Figures 5.6–5.11, can be attributed as artifacts of the simulation warm-up period. For example, as depicted in these figures, the first dip (at around days 10 to 15) is due to the deaths of the initial cohort of adults, all of which entered the system with the same age. The second dip (at around days 25 to 30) is due to the deaths of the first cohort of surviving adults (that emerged from the aquatic habitats in the Immature Adult stage, see Figure 3.1), followed by a rise caused by the next cohort. The subsequent dips, diminished successively in magnitude, reflect the deaths and emergence of subsequent cohorts, as the system approaches to the steady state.
5.5 Spatial Heterogeneity

In our spatial ABM, spatial heterogeneity encompasses the distribution and relative distances between various resources sought by the adult female mosquito agents within a given landscape. As shown by the following results, spatial heterogeneity, in some cases, may directly influence the mosquito population level. For each resource-type, it is controlled by two parameters: resource density and resource distribution.

To analyze how resource density affects abundance, we use landscapes with different resource densities of both resource-types: aquatic and bloodmeal locations (BMLs, referring to the houses), and also with varying CCCs. For this section, we denote each landscape by the tuple (\(\#AH, \#BML, CCC\)), where \(\#AH\) and \(\#BML\) denote the number of AHs and BMLs in the landscape, respectively.

We find that VA in the spatial ABM is eventually driven by two parameters: 1) \(ATT\) (Average Travel Time) and 2) \(CCC\). \(ATT\) is inversely proportional to resource densities. Since female mosquitoes must travel through the landscape in order to search for resources, \(ATT\) can increase or decrease with decreasing or increasing resource densities, respectively. For example, considering two hypothetical landscapes \(L_{sparse}\) and \(L_{dense}\) (both having the same dimensions), where \(L_{dense}\) has more resource densities than \(L_{sparse}\), adult females in \(L_{sparse}\) would incur higher \(ATT\) than that incurred in \(L_{dense}\). In terms of the mosquito population, this translates to the degree of ease with which adult female mosquitoes may find resources. Thus, in landscapes with the same dimensions, as we increase resource densities, \(ATT\) gradually declines until resource densities reaches a critical level, which we call the critical resource densities. Until the landscape possesses enough resources to reach this critical threshold, the abundance depends on both resource densities and \(CCC\). The less are the resource densities, the higher is the \(ATT\) (and vice versa). However, if the landscape has resource densities that exceed the critical threshold, \(ATT\) may no longer affect the resource-seeking behavior. On the other hand, if the landscape possesses enough
resources at or above the critical resource densities, abundance is driven primarily by $CCC$; and does not change significantly until $CCC$ is changed (irrespective of increasing the resource densities at that point, as long as it remains at or above the critical level).

In this study, though we do not empirically measure the average travel times or the critical resource densities for different landscapes, our preliminary simulation results confirm to the above insight. We start with different landscapes that all possess resource densities at or above the critical level. As the resource densities of both resource-types are increased (keeping the same landscape dimensions and the same $CCC$), we find that abundance remains unchanged until the $CCC$ is changed. For example, using $10 \times 10$ landscapes with $20K$ $CCC$ each, and denoting each landscape (as mentioned above) by the tuple $(\#AH, \#BML, CCC)$, we find that $(10, 10, 20K)$, $(10, 20, 20K)$, $(20, 10, 20K)$, and $(20, 20, 20K)$ all yield the same $VA (\approx 3000)$. In these cases, increasing the resource densities (by increasing the number of resources, e.g., from 10 to 20 $AH$s) does not affect $VA$ (note that these landscapes already had resource densities above the critical level). Abundance, in these cases, is primarily controlled by $CCC$. Similarly, using the same dimensions, as $CCC$ is increased to $40K$, landscapes $(20, 10, 40K)$ and $(20, 20, 40K)$ also yield the same $VA (\approx 6000)$. Again, abundance is limited by $CCC$ and not by resource densities. For brevity, we omit these results.

Given these initial results, we explore the effect of resource densities in a larger landscape (of dimension $30 \times 30$). In Figures 5.12 and 5.13 we show two sample landscapes with resource densities below and above the critical level, respectively. Figures 5.14 and 5.15 depict the results ($VA$) produced using these landscapes.

First, we explore the case with resource densities below the critical level. As shown in Figure 5.14 as we increase the $AH$-density (keeping $BML$-density and $CCC$ unchanged, i.e., with 20 $BML$s and 30$K CCC$ in all cases), $VA$ increases
Figure 5.12. Sample Landscape with 50 AHs and 20 BMLs. A sample $30 \times 30$ landscape with 50 AHs and 20 BMLs, generated by using our landscape generator and analysis tool, AnophGUI. The landscape possesses resource densities below the critical level. Each black circle represents an aquatic habitat, and each gray rectangle represents a house. Figure 5.14 shows the results using this landscape.

Figure 5.13. Sample Landscape with 100 AHs and 100 BMLs. A sample $30 \times 30$ landscape with 100 AHs and 100 BMLs, generated by using our landscape generator and analysis tool, AnophGUI. The landscape possesses resource densities above the critical level. Each black circle represents an aquatic habitat, and each gray rectangle represents a house. Figure 5.15 shows the results using this landscape.
Figure 5.14. Results for Resource Density below the Critical Level. VA in a 30 × 30 landscape with 30K CCC (for the landscape, see Figure 5.12), and resource density below the critical level. Each graph represents the average of 40 simulation runs, and shows only relevant portions of the one-year simulation results.

Figure 5.15. Results for Resource Density above the Critical Level. VA in a 30 × 30 landscape with 30K CCC (for the landscape, see Figure 5.13), and resource density above the critical level. Each graph represents the average of 40 simulation runs, and shows only relevant portions of the one-year simulation results.
and eventually reaches \( \approx 4500 \). This substantiates the first part of our previous claim: as resource density is increased and is kept below the critical level, \( ATT \) gradually declines, and, as a result, \( VA \) successively increases. We also make two other observations in Figure 5.14: 1) the rate of rise in \( VA \) is always faster with higher resource density cases; and 2) with a single aquatic habitat, the population dies out - due to the fact that it is too insufficient to maintain a sustainable mosquito population within the relatively larger dimensions \((30 \times 30)\) of the landscape.

Next, we explore the case with resource densities above the critical level. We investigate whether higher density of both resource-types affects the upper limit of sustainable population within the same landscape. Figure 5.15 shows the results: all cases with higher resource densities reach equilibrium with \( VA \approx 4500 \). This substantiates the second part of our previous claim: as resource density is increased above the critical level (keeping the same dimensions and the same CCC), \( VA \) remains unchanged, and is eventually limited by CCC. However, as seen in Figure 5.15, the rate of rise after the second dips (between days 45 to 76) is always faster with higher resource density cases, because, in these cases, adult female mosquito agents are able to find resources more frequently.

5.6 Summary

In this chapter, we described the spatial extension of our previous non-spatial ABM, and our landscape generator tools, which helped to create landscapes with desired characteristics for the spatial ABM. We discussed the results of model verification between the non-spatial and the spatial ABMs.

The spatial extension provided unique opportunities to investigate the effects of various spatial factors. We showed the effects of using different landscape patterns, variation in the relative size and density of the aquatic habitats, and variation in the overall capacity of the system. We also demonstrated the effects of spatial hetero-
geneity of the landscapes, by considering the resource density.

In the following two chapters, we use the spatial ABM to evaluate the impact of several vector control interventions (Chapter 6), and demonstrate an example application of the spatial ABM by integrating data from a geographic information system (Chapter 7).
6.1 Overview

As we discussed in Section 2, malaria modeling, both mathematical and agent-based, can play important roles to quantify the effects of malaria-control interventions and to answer other interesting research questions. Models can play key roles in selecting appropriate combinations of interventions to interrupt transmission and in setting response timelines and expectations of impact. In Section 2.5 we discussed several malaria models from the perspective of the design, implementation and applications of our ABM of An. gambiae. We also showed a systematic comparison of some features and assumptions of several recent malaria models, including those that are extended, improved, or modeled for the first time by our ABM, in Table 2.1.

Malaria research and control in Africa is experiencing enormous funding support (a six-fold increase from 2003 to 2009) to scale up a wide array of interventions [216], and vector management is the primary means of malaria prevention and control in Africa [257]. Since the Anopheles mosquitoes need to access bloodmeals and aquatic oviposition sites to complete their life-cycle, availability of these ecological resources, i.e., the human houses and aquatic habitats, has long been recognized as a crucial determinant of malaria transmission [197]. Reduced availability of either type of these spatial resources would prolong the gonotrophic cycle of the female mosquito and potentially affect malaria transmission. Also, these resources define landscape features

\footnote{Major portions of this chapter appeared in Arifin et al. [11].}
such as spatial heterogeneity, host availability, etc. In Chapter 5, we discussed the effects of spatial heterogeneity and resource availability on vector abundance. Importance of these factors for vector control have also been demonstrated by several studies. For example, using an availability-based model, Killeen et al. showed the influence of host availability on malaria vectors in African communities [120]. Menach et al. showed how the heterogeneity in human biting reflects the underlying spatial heterogeneity in the attractiveness, distribution and suitability of human houses and aquatic habitats [160]. To demonstrate the spatial characteristics of transmission by the An. gambiae complex in sub-Saharan Africa, Carter et al. identified some breeding sites as the foci of transmission, which are closely associated with particular locations; and the non-random distribution (clustering) of malaria case incidences in different households [34]. Conclusions from the above studies naturally lead to habitat-based interventions, which necessitates a landscape approach to incorporate the spatial processes of mosquito foraging for oviposition and host-seeking [87]. Spatially-explicit models, which permit the refined characterization of resource seeking to predict the impact of habitat-based interventions, can prove valuable to this end [87, 86, 85].

Larval source management (LSM), insecticide-treated nets (ITNs), and indoor residual spraying (IRS) have been extensively used as intervention tactics to reduce and control malaria in sub-Saharan Africa. Impact of various interventions (including LSM, ITNs and IRS) have been investigated by early and recent studies [114, 206, 121, 86, 85, 257, 40, 57].

There is now a consensus that malaria elimination with current tools is far more likely if the best available tools are used in combination [145]. The Integrated Vector Management (IVM) approach actively considers the notion of combining multiple interventions to control vector-borne diseases [65]. In the context of IVM, the WHO recommends the use of appropriate combinations of non-chemical and chemical meth-
Replicability of the *in-silico* experiments and simulations performed by early malaria models bear special importance. Replication confirms reproducibility, which, in turn, is regarded as one of the foundations of the entire scientific method.

In this chapter, using our spatial ABM, we first investigate the effects of LSM and ITNs separately (in isolation), and compare our results to those reported by two previous ABM-based studies, performed by the same authors, which explored the impact of applying LSM [86] and ITNs [85] as stand-alone interventions. For brevity, we hereafter refer to the studies as GN-LSM and GN-ITN, and the ABM used as GN-ABM, where *GN* refer to the initials of last names of the authors (Gu and Novak). We emphasize that the focus of replication is to achieve a qualitative (not absolute) match between results of our ABM and those reported in GN-LSM [86] and GN-ITN [85].

Then, using different population profiles to explore the human density effect, we investigate the combined impact of LSM and ITNs, and discuss similar results reported by other studies. Lastly, we recommend some guidelines for future ABM modelers, summarizing the insights and experience gained from our work of replicating some other studies.

The organization of this chapter is as follows: In the remainder of this section, we briefly introduce LSM and ITNs as vector control interventions. In Section 6.2, we discuss why replicability of simulations is important from the context of validity of assumptions made by independent studies. In Section 6.3, we discuss multiple definitions of *ITN coverage*, as found in recent malaria literature. We precisely define three coverage schemes, and highlight the importance of simulating the schemes and assessing their relative impacts by model-based studies. Section 6.4 introduces the *Integrated Vector Management (IVM)* approach, and discusses the benefits of using combined interventions to achieve synergistic effects on top of individual impacts.
offered by stand-alone interventions. Section 6.5 lists the assumptions and notations for simulations performed in this study.

Section 6.6 describes our methods of applying LSM and ITNs, both in isolation and in combination. In Sections 6.7, 6.8, and 6.9 we present the results of applying LSM in isolation, ITNs in isolation, and both interventions in combination, respectively. Section 6.10 discusses some additional insights obtained from the results. In Section 6.11, we provide some recommendations and guidelines for future ABM modelers of malaria. Section 6.12 concludes. Some additional materials for this chapter are presented in Appendix C.

6.1.1 Larval Source Management (LSM)

Larval Source Management (LSM), also known as source reduction, is one of the oldest tools in the fight against malaria. LSM refers to the management of aquatic habitats in order to restrict the completion of immature stages of mosquito development. In a recent study, Fillinger and Lindsay suggest that LSM can be successfully used for malaria control in African transmission settings by highlighting historical and recent successes, and discuss its potential in an IVM approach working towards malaria elimination [66, 65]. In areas with moderate and focal malaria transmission where larval habitats are accessible and well-defined, LSM is also cost-effective when compared with IRS and LLINs [253].

6.1.2 Insecticide-Treated Nets (ITNs)

Insecticide-treated Nets (ITNs), particularly the long-lasting insecticidal nets (LLINs), are considered among the most effective vector control strategies currently in use [257, 145, 91, 28]. To combat against the major malaria vectors (including An. gambiae) in Africa, scale-up applications of ITNs, which can offer direct personal protection to users as well as indirect, community protection to non-users
(through insecticidal and/or repellent effects), are advocated [85, 91]. Primarily due to mathematical convenience, earlier models that studied the impact of ITNs on malaria transmission assumed a uniform contact structure between mosquitoes and hosts across the landscape [122, 134]. However, empirical data indicating limited flight ranges and sensory perception of mosquitoes suggest that proximity between the mosquitoes and their hosts can play a crucial role in the mosquito biting behavior [75, 73, 76, 74, 162]. Hence, spatially-explicit models are needed to analyze the local host-seeking process of the mosquitoes, and to study the responses of mosquitoes to ITNs. Such models can also provide evidence for the need of entomological surveillance for evaluation of scale-up ITN programs [85].

6.2 Replicability of Simulations

Replication, which is treated as the scientific gold standard to judge scientific claims, allows independent researchers to address a scientific hypothesis and produce evidence for or against it [184, 108]. Although computational science has led to exciting new developments, the nature of the work has also exposed shortcomings in the general ability of the research community to evaluate published findings [184]. Replication ensures reproducibility, which refers to the independent verification of prior findings, and is at the core of the spirit of science [204, 203]. In agent-based modeling and simulation (ABMS), replication is also known as model-to-model comparison, alignment, or cross-model validation. It falls under the broader subject of verification and validation (V&V). One of its goals is to try to align multiple models in order to investigate whether they produce similar results [116, 255]. When the original models (e.g., the source codes) are available, a stricter form of model verification, known as docking, may also be performed. In Chapter 4 we showed how to achieve docking between separate implementations of our malaria ABMs.

As mentioned before, one of the goals of this study is to replicate the results
and improve/extend some assumptions of two published studies, GN-LSM [86] and GN-ITN [85]. Using an ABM that simulates the life-cycle of individual mosquitoes and the mosquito foraging (resource-seeking) behavior, GN-LSM [86] and GN-ITN [85] study the impact of source reduction (LSM), and ITNs, respectively, as stand-alone interventions. Critical examination of these studies reveals that although they provide reasonably plausible results, some assumptions may be extended to include two major modeling improvements:

- number of replicated simulation runs, and
- boundary type of the landscapes

Both of these issues are described below.

### 6.2.1 Replicated Simulation Runs

Any simulation model which involves substantial stochasticity should conduct sufficient number of replicated runs (with identical parameter settings but different random seeds), and the average and/or aggregate results of these replicated runs should be reported, as opposed to reporting results from a single run. Sufficient number of replications are required to ensure that, given the same input, the average response can be treated as a deterministic number, and not as random variation of the results. This allows to obtain a complete statistical description of the model variables. The same principle also applies to a set of stochastic (Monte Carlo) simulation models in other domains (e.g., traffic flow, financial problems, risk analysis, supply chain forecasting, etc.), where, in most cases, the standard practice is to report the averages and standard deviations of the measures of interest (known as the Measures of Effectiveness, or MOEs) [35, 38].

Since most epidemiology models (including ABMs) involve substantial stochasticity in the forms of probability-based distributions and equations, performing sufficient
number of replicated runs is also important for validation of the results. In malaria ABMs, decisions are often simulated using random draws from certain distributions. These sources of randomness are used to represent the diversity of model characteristics, and the behavior uncertainty of the agents’ actions, stages, etc., with the goal to mimic/simulate the reality as closely as desired. For example, in our ABM, when a host-seeking mosquito searches for a bloodmeal in a ITN-covered house, a 50% ITN mortality would mean that it may die with a probability of 0.5, which can be simulated using random draws from a \textit{uniform} distribution. As another example, the number of eggs in each egg-batch of a \textit{Gravid} mosquito is simulated using random draws from a \textit{normal} distribution with mean (average) = 170 and standard deviation = 30. The randomness has significant impact on the results of the simulation, and different simulation runs can therefore produce significantly different results, due to a different sequence of pseudo-random numbers drawn from the distributions. So, we perform replicated runs for all simulations reported in this study, as opposed to single runs performed in GN-LSM and GN-ITN \cite{86, 85}.

6.2.2 Boundary Type of Landscapes

The second issue, the use of a specific boundary type, may greatly impact the mosquito movement process. In general, three different boundary types are commonly used in ABMS: absorbing, non-absorbing, and reflecting. With an \textit{absorbing boundary}, mosquitoes are permanently removed (effectively killed) when they hit an edge of the landscape’s boundary. On the other hand, with a \textit{non-absorbing boundary}, when mosquitoes hit an edge, they re-enter the landscape from the edge directly opposite of the exiting edge (and thus are not killed due to hitting the edge). Unless the underlying landscape reflects a completely isolated geographic location (e.g., an island far away from the mainlands), in reality, when mosquitoes hit an edge, logical approaches are either to reflect the mosquito back from the same edge (reflecting boundary), or
to coerce the mosquito to re-enter from the opposite edge (non-absorbing boundary).
We argue that a non-absorbing boundary more realistically captures the mosquito population dynamics\(^2\). This is specially true when the resource densities are high and the resources are more evenly distributed across the landscape. The GN-ABM uses an absorbing boundary for all landscapes. In our ABM, all landscapes are modeled topologically as 2D torus spaces, and use a non-absorbing boundary (however, to compare with GN-LSM \([86]\), we first report results that use an absorbing boundary).

6.3 Multiple Definitions of ITN Coverage

As we describe later (in Section 6.6.2), application of ITNs is modeled as a series of three ITN parameters: coverage, repellence, and mortality (insecticidal effect of the bednets). However, in malaria literature, multiple definitions of the term ITN coverage can be found. The Roll Back Malaria (RBM) Partnership uses ITN coverage as the proportion of households owning a bednet or sleeping under a bednet \([181]\). This definition has been used as an outcome indicator (of population coverage) for evaluating RBM Partnership’s technical strategies, and is also used by GN-ITN \([85]\).

On the other hand, the World Health Organization (WHO) reports ITN coverage as the number of bednets distributed per person at risk \([251]\). In some studies, ITN coverage is also defined as the proportion of populations sleeping under treated bednets \([122]\), and is used more widely in recent models \([122, 257, 40, 57]\).

However, this distinction in multiple definitions of ITN coverage, primarily concerning coverage levels of households and individuals, has not been addressed (within a single study) by most recent models. The WHO emphasizes the importance of scale-up ITNs coverage beyond vulnerable population (children under five years of age and pregnant women) as a priority for combating malaria in tropical Africa \([251]\). Also,

\(^2\)Recall that we used a non-absorbing boundary for all simulations reported in Chapter 5.
several studies have shown that the patterns of coverage and effective coverage are important determinants of ITN/LLIN success [83], and simple ITN/LLIN models in which the coverage scheme is not carefully designed can lead to overly optimistic results [119, 134, 218]. Without a precise definition of the scheme used in a particular model, the task of replication becomes much harder. The comparison of results from using the three schemes may guide future modelers to decide and choose one to use in their models.

Based on these, we argue that simulating different definitions of ITN coverage and assessing their relative impacts are important, especially when replicating and validating results of an earlier model that used either of these definitions (e.g., [85]). Hence, as an extension to GN-ITN [85], we simulate and compare three different definitions/schemes of ITN coverage which differ by the number of persons actually covered by bednets in a ITN-covered house:

- household-level partial coverage with single chance for host-seeking,
- household-level partial coverage with multiple chances for host-seeking, and
- household-level complete coverage

These three schemes of ITN coverage differ by the number of persons actually covered by bednets in a house that is under ITN coverage. We define household-level coverage and population-level coverage as:

\[
\text{Household-level coverage (\%)} = \frac{\text{Number of houses with coverage}}{\text{Total number of houses}} \times 100
\]

\[
\text{Population-level coverage (\%)} = \frac{\text{Number of bednet users}}{\text{Total human population}} \times 100
\]

In the following, we describe the three coverage schemes in details.
6.3.1 Household-Level *Partial Coverage with Single Chance* for Host-Seeking

In household-level *partial coverage with single chance* for host-seeking, each house with ITN coverage is assigned a single bednet, and two randomly selected persons in the house are protected by the bednet (irrespective of the total number of persons in the house). Once a host-seeking mosquito enters a ITN-covered house, and is not deterred by the repellence, it gets a single chance of obtaining a bloodmeal by picking a random host. Since at most two persons can sleep under the bednet, the probability of a random host sleeping under the bednet is $2/n$, where $n$ is the number of persons in the house. Thus, the probability to obtain a bloodmeal from a non-protected host in the house is $1 - 2/n$. If the host is protected (sleeps under the bednet), the mosquito cannot get a bloodmeal but still runs the risk of being killed by the ITN mortality (insecticidal effect of the bednets). If it can survive, it must start searching for another house. Otherwise (i.e., if the host is unprotected), the mosquito gets a bloodmeal. The scheme is depicted as a logical flowchart in Figure 6.1.

6.3.2 Household-Level *Partial Coverage with Multiple Chances* for Host-Seeking

The second scheme, household-level *partial coverage with multiple chances* for host-seeking, works similarly as the first one, except for the fact that a host-seeking mosquito gets $n$ chances in the same house (where $n$ is the number of persons in the house). If it cannot get a bloodmeal within $n$ chances and still survives the ITN mortality, it must start searching for another house. Note that with this scheme, even though the mosquito gets multiple chances for host-seeking, it also encounters the risk of being exposed to the ITN mortality each time (if the randomly selected host sleeps under bednet). With both these schemes, even when all houses are ITN-covered (i.e., 100% household-level coverage), a portion of the population may still remain unprotected, and thus, vector population may not be completely suppressed. The scheme is depicted as a logical flowchart in Figure 6.2.
Figure 6.1. Logical Flowchart for Household-Level Partial Coverage with Single Chance for Host-Seeking. In this scheme, each house with ITN coverage is assigned a single bednet, and two randomly selected persons in the house are protected by the bednet. Once a host-seeking mosquito enters a ITN-covered house, and is not deterred by the repellence, it gets a single chance of obtaining a bloodmeal by picking a random host in the house. The probability to obtain a bloodmeal from a non-protected host in the house is $1 - 2/n$. If the host is protected (sleeps under the bednet), the mosquito cannot get a bloodmeal, but still runs the risk of being killed by the ITN mortality (insecticidal effect of the bednets). If it can survive, it must start searching for another house. Otherwise (i.e., if the host is unprotected), the mosquito gets a bloodmeal.
Figure 6.2. Logical Flowchart for Household-Level Partial Coverage with Multiple Chances for Host-Seeking. This scheme works similarly as the previous one (see Figure 6.1), except for the fact that a host-seeking mosquito gets \( n \) chances to obtain a bloodmeal in the same house, where \( n \) is the number of persons in the house. If it cannot get a bloodmeal within \( n \) chances, and still survives the ITN mortality, it must start searching for another house.
6.3.3 Household-Level Complete Coverage

With the last scheme, household-level complete coverage, if a house is ITN-covered, all persons in the house are protected by bednets (and hence we use the term complete). This can simulate, for example, an ITN study over a region where there are enough bednets to protect every person in a ITN-covered house. In this scheme, when a host-seeking mosquito enters a ITN-covered house and is not deterred by the repellence, it cannot get a bloodmeal (because all persons are covered), and must search for another house. Thus, it incurs additional delays and risks for the mosquito to be eventually successful in obtaining a bloodmeal. The scheme is depicted as a logical flowchart in Figure 6.3.

6.3.4 Population Profiles for ITNs

Since we replicate and compare our results of applying ITNs with GN-ITN [85], we build two population profiles for ITNs. The 50 houses in the landscape (see Figure C.2) used in the GN-ITN study [85] accommodate a total human population of 185, with the average of household residents being 3.7 (with standard deviation of 1.2). We follow the same distribution to generate our population profiles, ensuring that the total human population is 185, with each house having at least two residents. The profiles are shown in Tables 6.1 and 6.2. Table 6.1 shows the differences in household-level coverage and population-level coverage, as well as the variation in number of bednet users and non-users, for varying levels of ITN coverage with the partial coverage schemes (see Section 6.3.1 and Figure 6.1). In these profiles, the same household-level coverages, used with different ITN schemes (as described above), may yield different population-level coverages. Table 6.2 shows the same with the complete coverage schemes (see Section 6.3.3 and Figure 6.3).

The distinction between partial (with single or multiple chances) and complete schemes becomes apparent when we compare the respective numbers for varying
Figure 6.3. Logical Flowchart for Household-Level *Complete* Coverage. In this scheme, if a house is ITN-covered, all persons in the house are protected by bednets (and hence we use the term *complete*). In this scheme, when a host-seeking mosquito enters a ITN-covered house and is not deterred by the repellence, it cannot get a bloodmeal (because all persons are covered), and must search for another house. Thus, it incurs additional delays and risks for the mosquito to be eventually successful in obtaining a bloodmeal.
### Table 6.1

**POPULATION PROFILES FOR VARYING LEVELS OF ITN COVERAGE WITH THE PARTIAL COVERAGE SCHEMES**

<table>
<thead>
<tr>
<th>ITN Coverage</th>
<th>Household-level Coverage (%)</th>
<th>Population-level Coverage (%)</th>
<th>Number of Net Users</th>
<th>Number of Non-users</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>40</td>
<td>21.62</td>
<td>40</td>
<td>145</td>
</tr>
<tr>
<td>0.6</td>
<td>60</td>
<td>32.43</td>
<td>60</td>
<td>125</td>
</tr>
<tr>
<td>0.8</td>
<td>80</td>
<td>43.24</td>
<td>80</td>
<td>105</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>54.05</td>
<td>100</td>
<td>85</td>
</tr>
</tbody>
</table>

Levels of ITN coverage in Tables 6.1 and 6.2 for any ITN coverage level (column 1 in both tables), with the same household-level coverages (compare columns 2 in both tables), the complete coverage scheme has almost twice the population-level coverages than that in the partial coverage schemes (compare columns 3 in both tables). The same applies to the corresponding number of bednet users (compare columns 4 in both tables).

6.4 Combined Interventions

The IVM approach, promoted by the WHO [251], is defined as a rational decision-making process for the optimal use of resources in the management of vector populations, with the long-term goals of reducing or interrupting transmission of vector-borne diseases [166]. Because of improved efficacy, cost-effectiveness, ecological soundness and sustainability, IVM is increasingly being recommended as an option for sustainable malaria control [166].
TABLE 6.2

POPULATION PROFILES FOR VARYING LEVELS OF ITN COVERAGE WITH THE COMPLETE COVERAGE SCHEME

<table>
<thead>
<tr>
<th>ITN Coverage Household-level Coverage (%)</th>
<th>Population-level Coverage (%)</th>
<th>Number of Net Users</th>
<th>Number of Non-users</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>40</td>
<td>41.08</td>
<td>76</td>
</tr>
<tr>
<td>0.6</td>
<td>60</td>
<td>59.46</td>
<td>110</td>
</tr>
<tr>
<td>0.8</td>
<td>80</td>
<td>82.70</td>
<td>153</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>100.00</td>
<td>185</td>
</tr>
</tbody>
</table>

The rationale of using combined interventions is that multiple interventions can offer synergistic effects on top of individual impacts offered by each intervention (when applied alone), thus producing a result which is greater than the sum of their individual effects. Such synergistic effects have been demonstrated by several model-based and field-based studies. Using a mathematical model, Yakob and Yan theoretically examined the application of LSM with ITNs in reducing malaria transmission [257].

The combined impact of ITNs (or LLINs) and IRS is examined by Chitnis et al. using the OpenMalaria model [40], and by a recent field-based study in south eastern Tanzania by Okumu et al. [175]. Using an ecological model, White et al. explored the impact of LLINs, IRS, larvicide and pupacide [246]. Eckhoff used a cohort-based vector simulation model to demonstrate the effects of increasing coverage with perfect IRS, combining IRS and ITNs, and combining larval control (using larvicides) and space spraying [57]. Using an individual-based simulation model with different combinations of LLINs, IRS, artemisinin-combination therapy (ACT), mass screening and treatment (MSAT), and vaccines, Griffin et al. showed that the combined interventions can result in substantial declines in malaria prevalence across a wide
range of transmission settings [33]. Kleinschmidt et al. presented a summary of studies comparing the effect of IRS combined with ITNs [125]. Some of these studies suggest that when combined interventions are applied, it may be more beneficial to target different stages of the mosquito’s life-cycle, rather than applying interventions that may interfere with each other (e.g., LLINs and IRS) [246].

Two important notions emerged from the conclusion of these studies:

• when combined interventions are applied, the individual efficacy of each intervention needs to be ensured, and
• attacking different behaviors or life-cycle stages of the mosquito may be more synergistic

Based on the above, we select LSM and ITNs, and explore their combined impact with our spatial ABM. To ensure (1), we first examine the impact of both as stand-alone interventions. In doing so, we replicate the two GN studies [86, 85], and also extend/improve some of the original assumptions. It is interesting to note that no ABMs ever explored the combined impact of LSM and ITNs as interventions before (although some other combinations were explored using ABMs). Since LSM and ITNs primarily affect two different life-cycle stages (i.e., larval and adult stages, respectively) and involve two different types of ecological resources (i.e., aquatic habitats and human houses, respectively), this combination is potentially important.

6.5 Assumptions and Notations

For this study, we use the spatial ABM described in detail in Chapter 5. The life-cycle of mosquito agents in the ABM was shown in Figure 3.1. Details about the movement of adult female mosquitoes was described in Section 5.2.3.
6.5.1 Assumptions

The work presented in this chapter is theoretical. We assume the presence of only one vector, *An. gambiae*. We concentrate on the vector life-cycle dynamics and do not yet include the parasite life-cycle and the malaria transmission cycle. Mosquitoes senesce, and their probability of death increases with age. The human population is modeled as static, i.e., humans do not move in space. All humans are assumed to be identical. For host-seeking, we do not model alternative hosts for blood-feeding (e.g., cattle), and the only bloodmeal-sources are humans in the houses. Daily temperature, variations in which can affect our model’s output (see [8, 263] for details), is fixed at 25°C for this study. We do not account for seasonality and other weather/climate parameters. Each aquatic habitat is set with a carrying capacity $CC$ of 1000 (see Section 3.6 for details). Time is modeled with hourly (instead of daily) time-steps, since this approach provides much more flexibility in modeling certain agent behaviors (e.g., host-seeking to start at a particular hour at night). For the grid-based landscapes, we set the size of each cell to be 50 m, reflecting the limited perceptual range of *An. gambiae* [86]. For LSM, all aquatic habitats are treated indifferently, i.e., with no inherent differences in their attractiveness and productivity. For ITNs, we ignore transient effects such as the decay of insecticide effectiveness of the bednets. Complete usage (adherence) is assumed, i.e., humans provided with a bednet are always assumed to sleep under it during night. We assume all ITN parameters (coverage, repellence, and insecticidal effect) to be invariant over time, and ignore any possible development of insecticide resistance in the mosquitoes.

As before, female adult mosquito abundance is treated as the primary output of the model. The associated $CC$ of each aquatic habitat determines soft limits on larval density of the habitat, and approximates the *Gravid* female’s inclination to avoid less suitable (e.g., over-crowded) habitats (see Section 3.6). Unlike other studies [86, 85], $CC$ is not treated as a hard limit. When no intervention is in
action, the mosquito population is governed by the combined carrying capacities of all aquatic habitats, and the density-dependent oviposition mechanism, which limits the potential number of eggs that a female mosquito may preferentially lay in an aquatic habitat, considering both the associated $CC$ and the biomass already present in the habitat.

6.5.2 Simulations

All simulations are started with 20000 *Gravid* mosquitoes seeking human blood-meals, which are initially placed at randomly selected houses. Each simulation is run 50 times, and average results of all 50 runs are reported. Each simulation is run for at least one year (unless we find it necessary to run longer). Intervention(s) are applied on day 100, and continued up to the end of the simulation. Thus, we ensure that a long enough *warm-up* period has passed to reach a steady state (which, without any intervention, occurs around day 50), and that the results are reported after the simulation reaches equilibrium. Where applicable, percent reduction ($PR$) values in mosquito abundance are calculated by averaging 30-day abundances (after the population reaches steady-state) from two intervals: *before* and *after* applying the intervention(s) to the base mosquito population. All simulations are submitted using the Sun Grid Engine (SGE) job scheduler, and run as single-threaded programs, in single-process per core mode in computing clusters with multiple cores.

6.6 Applying LSM and ITNs

In this section, we describe, in details, our methods of applying LSM and ITNs. In Section 6.6.1 we describe the application of LSM in isolation, the digitization process of the 18 grid-based landscapes used in GN-LSM [86], and the construction of different LSM scenarios (targeted and non-targeted). Section 6.6.2 describes the application of ITNs in isolation, the replication of settings, and the digitization process of the single
grid-based landscape used in GN-ITN [85]. Section 6.6.3 describes the application of LSM and ITNs in combination, the three landscapes with varying density (Low, Medium, and High) of bloodmeal locations that we create using VectorLand, and other parameter settings.

6.6.1 Applying LSM in Isolation

To explore the impact of LSM in isolation (i.e., without any other intervention) and to replicate the results of GN-LSM, we discretize and digitize the 40 × 40 grid-based landscapes used in GN-LSM [86]. In the digitization process, the original tiny landscapes (from [86]) are enlarged, and grid-lines are added to aid in measuring the objects’ coordinates. The coordinates are then measured by inspection. To locate the center of each object (an aquatic habitat or a house), we use distances (in both x- and y- axes) from the nearest grid-lines. Whenever multiple objects overlap and appear to be rendered on top of one another, we use the best guess to infer the center coordinates. The landscapes are then generated by using our landscape generator tool VectorLand. Each of the 18 landscapes, depicted in Figure C.1 contains 70 aquatic habitats (blue circles) and three different arrangements of 20 houses (black house icons): diagonal, horizontal, and vertical. For each arrangement, different LSM scenarios (targeted and non-targeted) are also constructed, as was done by [86]. The three targeted interventions T1, T2 and T3 refer to the removal of aquatic habitats within 100, 200 and 300 m of surrounding houses, accounting for 4, 17 and 28 of 70 habitats, respectively. C1, C2 and C3 refer to non-targeted, random removal of the same numbers of aquatic habitats as the corresponding targeted interventions. Removal of an aquatic habitat makes it completely inaccessible to Gravid mosquitoes, and no eggs can be laid in it during oviposition. In practice, this is usually done by habitat modification (a category of LSM), which results in permanent change of land and water, and is performed by means that include landscaping, drainage of
surface water, land reclamation and filling, coverage of large water storage containers, wells and other potential breeding sites, etc. [65]. Increasing LSM coverage, although affecting the larval population (by killing the biomass in the corresponding aquatic habitats), does not increase the mortality of adult mosquitoes; it just decreases the probability of successfully finding an aquatic habitat (and hence delaying the process) by adult females trying to oviposit. Note that the digitization of these landscapes from GN-LSM [86] (and later from GN-ITN [85]) is conducted primarily for validation, comparison and replication purposes. It is much easier and less time-consuming to generate new landscapes with any desired spatial distribution and parameter combinations using VectorLand (as we do later for applying LSM and ITNs in combination). However, to be able to directly compare our results with GN-LSM, and to adhere to the requirements of a standard replication process, the digitization of the original landscapes is necessary.

To compare the impact of LSM using the above landscapes, we use a fixed daily mortality rate (DMR) of 0.2 for the absorbing boundary in order to match the DMR of the GN-LSM study [86]. However, our original model uses age-dependent DMRs for some stages in the life-cycle of mosquito agents [8, 263, 9]. In our model, DMRs for all the Adult stages and the Larva stage are age-dependent, and the other stages (Egg and Pupa) have fixed DMRs of 0.1. Hence, in simulations that use a non-absorbing boundary, we use age-dependent DMRs for all Adult stages and the Larva stage (see Section 6.10 for details).

6.6.2 Applying ITNs in Isolation

Response of host-seeking mosquitoes to ITNs is modeled as a series of three ITN parameters: coverage $C$, repellence $R$, and mortality $M$ (note that we use the term mortality to refer to the insecticidal effect of the bednets, i.e., the mortality concurred by ITNs). When a female mosquito (being in the Bloodmeal Seeking stage) finds a
house, coverage is checked first to ensure whether the house is ITN-covered. If it is covered, repellence comes into action: the mosquito may be repelled by ITN and thus forced to search for another house. If it can avoid repellence, a random host is picked in the house. If the host sleeps under bednet, mortality comes into action: it may be killed due to mortality. If it survives the mortality, depending on the ITN coverage scheme (see Section 6.3), it either picks another random host in the same house or must search for another house. If, on the other hand, the host does not sleep under bednet, feeding is assumed to be always successful.

As stated before, simulating the three different definitions (schemes) of ITN coverage is important because although different studies used different schemes [85, 122, 227, 40, 57], none (including ABMs and mathematical models of malaria) actually compared their relative impacts side-by-side. Without a precise definition of the scheme used in a particular model, the task of replication becomes much harder. Hence, we argue that the comparison of results from using the three schemes may guide future modelers to decide and choose from which one to use in their models. These three schemes of ITN coverage differ by the number of persons actually covered by bednets in a house that is under ITN coverage. Note that the same household-level coverage in different schemes may yield different population-level coverage, as shown in Tables 6.1 and 6.2. These three schemes are depicted as logical flowcharts in Figures 6.1–6.3.

To evaluate the impact of ITNs, we replicate the settings used in the GN-ITN study [85]. Using our landscape generator tool VectorLand, we digitize the single 40×40 grid-based landscape used in GN-ITN by the same procedure as described before (to digitize the 18 landscapes from GN-LSM). The landscape, depicted in Figure C.2, contains 90 aquatic habitats (blue circles) which are randomly distributed, and 50 houses (black house icons) which are arranged diagonally. To run the simulations, we use representative sample values from the parameter space for the three ITN
parameters (coverage, repellence, and mortality). The parameter values are shown in Table 6.3. Combining these values yields 60 distinct parameter combinations. For each combination, 50 replicated simulations are run for each of the three coverage schemes (see Section 6.3), yielding a total of 15000 simulations. The average results are reported. A non-absorbing boundary is used for all cases.

### 6.6.3 Applying LSM and ITNs in Combination

To evaluate the impact of applying LSM and ITNs in combination, we create three $40 \times 40$ landscapes (using VectorLand), with varying density of bloodmeal locations ($BML$s, or houses) where the density refers to the number of $BML$s: Low (20), Medium (70), and High (200). For each BML density, a corresponding human population density (total human population) is also set: Low (100), Medium (350), and High (1000). Sample landscapes with the three BML density levels are shown in Figure 6.4. Parameter values used to run simulations with these three landscapes are shown in Table 6.4. For each of the 240 distinct parameter combinations, 50 replicated simulations are run, yielding a total of 12000 simulations. For all cases, the household-level complete coverage scheme is used for ITNs, and ITN repellence ($R$) is ignored (i.e., $R$ is set to 0.0). Initially, aquatic habitat density is fixed at 200.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BML density</td>
<td>Low (20), Medium (70), High (200)</td>
</tr>
<tr>
<td>Population density</td>
<td>Low (100), Medium (350), High (1000)</td>
</tr>
<tr>
<td>LSM coverage</td>
<td>0.0, 0.1, 0.3, 0.6, 0.9</td>
</tr>
<tr>
<td>ITN mortality</td>
<td>0.0, 0.3, 0.7, 1.0</td>
</tr>
</tbody>
</table>

(in all landscapes), and later reduced as LSM coverage is increased.
Figure 6.4. Sample Landscapes for Applying LSM and ITNs in Combination. The three $40 \times 40$ landscapes, each containing 200 aquatic habitats, represent different density of bloodmeal locations (BMLs): Low (20), Medium (70), and High (200), with corresponding human population density of 100, 350, and 1000, respectively. Aquatic habitats and bloodmeal locations are shown as blue circles and house-shaped icons, respectively. For 240 distinct parameter combinations (involving BML density, LSM coverage, ITN coverage, and ITN mortality, see Table 4), similar landscapes are generated, and 50 replicated simulations are run for each (see Figure 6.11 for results). All landscapes are generated using our landscape generator tool VectorLand.
6.7 Results: Impact of LSM

In this section, we present the results of applying LSM in isolation. Figures 6.5 and 6.6 show the impacts of applying LSM on mosquito abundance, using absorbing and non-absorbing boundaries, respectively (the reproduced landscapes used for LSM application are shown in Figure C.1). For brevity, we show the 150 days results (day 100 - day 249), with LSM being applied on day 100; the full one-year results are shown in Figures C.3 and C.4.

To compare our results with the GN-LSM study [86], we also calculate the percent reduction (PR) values in abundance, which are shown in Table 6.5. Rows labeled with Diagonal, Horizontal, and Vertical refer to the different arrangements of houses in the landscapes (see Figure C.1). C1, C2, C3 and T1, T2, T3 refer to non-targeted and targeted removal scenarios, respectively. Each value (in the rows labeled as Absorbing and Non-absorbing) represents the average PR of 50 simulation runs.
Figure 6.5. Impact of LSM on Mosquito Abundance, using an Absorbing Boundary. The figure depicts our results of applying LSM (in isolation) as we replicate the results of GN-LSM [86] along with the scenarios and landscapes. Each sub-figure represents a specific LSM scenario: T1, T2 and T3 refer to targeted removal of aquatic habitats within 100, 200 and 300 m of surrounding houses, respectively. C1, C2 and C3 refer to non-targeted, random removal of the same numbers of aquatic habitats as the corresponding targeted interventions. Within each sub-figure, the Diagonal, Horizontal, and Vertical plots represent abundances (for the specified LSM scenario) for three different arrangements of houses in the landscapes (see Figure C.1 for the landscapes). With an absorbing boundary, mosquitoes are killed when they hit an edge of the landscape’s boundary. The x-axis denotes simulation time (in days), and the y-axis denotes mosquito abundance. For brevity, we show the 150 days results (day 100 - day 249); the full one-year results are given in Figure C.3. This figure represents averages of a total of 900 (18 × 50) simulations.
Figure 6.6. Impact of LSM on Mosquito Abundance, using a Non-Absorbing Boundary. The figure depicts our results of applying LSM (in isolation) as we replicate the results of GN-LSM [86] using a non-absorbing boundary. With a non-absorbing boundary, when mosquitoes hit an edge of the landscape’s boundary, they enter the landscape from the edge directly opposite of the exiting edge (and thus are not killed due to hitting the edge). Results within each sub-figure are obtained using the same parameters as in the corresponding sub-figure of Figure 5 (except the boundary type). For brevity, we show the 150 days results (day 100 - day 249); the full one-year results are given in Figure C.4. This figure represents averages of a total of 900 ($18 \times 50$) simulations.
### TABLE 6.5

PERCENT REDUCTIONS IN ABUNDANCE WITH LSM (APPLIED IN ISOLATION): A COMPARISON WITH GN-LSM [86]

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>T1</th>
<th>C2</th>
<th>T2</th>
<th>C3</th>
<th>T3</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diagonal</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GN-LSM (absorbing)</td>
<td>4.2</td>
<td>38.4</td>
<td>8</td>
<td>100</td>
<td>69.6</td>
<td>100</td>
<td>[86]</td>
</tr>
<tr>
<td>Absorbing</td>
<td>2.08</td>
<td>-21.63</td>
<td>3.56</td>
<td>43.82</td>
<td>31.74</td>
<td>85.32</td>
<td>This study</td>
</tr>
<tr>
<td>Non-absorbing</td>
<td>-1.82</td>
<td>-23.24</td>
<td>0.22</td>
<td>39.55</td>
<td>29.72</td>
<td>82.65</td>
<td>This study</td>
</tr>
<tr>
<td><strong>Horizontal</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GN-LSM (absorbing)</td>
<td>8.9</td>
<td>-5.7</td>
<td>44</td>
<td>100</td>
<td>34.3</td>
<td>100</td>
<td>[86]</td>
</tr>
<tr>
<td>Absorbing</td>
<td>4.35</td>
<td>7.37</td>
<td>-3.96</td>
<td>29.03</td>
<td>29.3</td>
<td>78.82</td>
<td>This study</td>
</tr>
<tr>
<td>Non-absorbing</td>
<td>3.25</td>
<td>6.71</td>
<td>-3.27</td>
<td>22.29</td>
<td>34.16</td>
<td>54.01</td>
<td>This study</td>
</tr>
<tr>
<td><strong>Vertical</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GN-LSM (absorbing)</td>
<td>2.8</td>
<td>30.6</td>
<td>16.67</td>
<td>100</td>
<td>33.14</td>
<td>100</td>
<td>[86]</td>
</tr>
<tr>
<td>Absorbing</td>
<td>5.21</td>
<td>15.45</td>
<td>24.17</td>
<td>55.54</td>
<td>43.21</td>
<td>91.79</td>
<td>This study</td>
</tr>
<tr>
<td>Non-absorbing</td>
<td>5.32</td>
<td>14.32</td>
<td>23</td>
<td>52.13</td>
<td>40.45</td>
<td>88.20</td>
<td>This study</td>
</tr>
</tbody>
</table>
Although the two models (i.e., the GN-ABM [86] and our ABM) differ in several assumptions, in most cases, we observe general agreement in changes in PRs (i.e., an increase or decrease) as we move from one landscape to another, as shown in Figures 6.5, 6.6, and Table 6.5. For all three landscape types (Diagonal, Horizontal, and Vertical), in our model, the absorbing boundary almost always (in 17 out of 18 scenarios, i.e., 94% cases) yields larger PR than that of the non-absorbing case within the same scenario. While this trend is generally expected due to the additional (but unrealistic) killing effect of the absorbing boundary, this indicates the validity of results obtained from comparing the models using different boundary types.

It is interesting to observe that in Figures 6.5 and 6.6, except for scenario T1, abundances in all other scenarios for the Horizontal landscape are greater than those for the Diagonal and Vertical landscapes. This is because the average distance between aquatic habitats and bloodmeal locations (when both of these resource types are ranked according to distances from one another) for the Horizontal landscape is less than those for the Diagonal and Vertical landscapes. As a result, female mosquitoes need to travel shorter average distances in the Horizontal case in order to find resources, and thus completing their gonotrophic cycles. For scenario T1 (which is obtained by removing four aquatic habitats from the baseline landscapes), however, abundance for the Diagonal landscape is greater than that for the Horizontal landscape. To explore why, we measure the effective shortest distance (ESD) between each of the four removed aquatic habitats and to their seven nearest bloodmeal locations. ESD measures the shortest distance, in units of number of cells, between the source and the destination cells (recall that each cell in the landscape is 50m \times 50m; thus x ESD means x \times 50m), and includes diagonal paths wherever necessary, since mosquitoes are allowed to move diagonally in the ABM. It turns out that \( ESD_{Diagonal} = 143 \) and \( ESD_{Horizontal} = 197 \), i.e., \( ESD_{Diagonal} < ESD_{Horizontal} \) (see Figure C.1 for the specific landscapes). This suggests that removal of these four
aquatic habitats in scenario T1 has less impact for the Diagonal landscape than for the Horizontal landscape - female mosquitoes can find bloodmeals more easily by traveling less distances in the former (Diagonal) case, resulting greater abundance.

6.7.1 Impact of Single vs. Multiple Runs

As explained before, different simulation runs (with identical parameter settings) can produce significantly different results due to the stochasticity involved while generating random draws from the probability distributions. The importance of multiple simulation runs, instead of a single run, is depicted in Figure 6.7, where we derive the maximum, minimum, and average abundance values obtained in each time-step across 50 replicated runs from four sample scenarios. As evident from Figure 6.7, the average plot lies within a band (envelope) defined by the maximum and minimum plots. If replication is not done (by performing multiple simulation runs), the results could have potentially taken any trajectory bounded within the band, and thus would have been less reliable. Also, note that the average plot is much smoother than the other two, suggesting much less abrupt changes (caused by the random events). All simulation results reported in this work represent the same replication mechanism of multiple runs.

6.7.2 Impact of Boundary Types

As stated above, in 94% cases, use of an absorbing boundary yields less abundance than that with a non-absorbing boundary. Also, with an absorbing boundary, even before applying LSM (i.e., before day 100), abundances with all landscapes are too low when compared to those with a non-absorbing boundary (see Figures C.3 and C.4 for the full one-year results). Since at the beginning of all simulations, female mosquitoes start their activities from randomly selected houses, a good portion of them aggregate around these clumped houses. We verified that in many cases, these
Figure 6.7. Importance of Performing Multiple Simulation Runs. The importance of performing multiple simulation runs (instead of a single run) can be seen by comparing abundances for maximum, minimum, and average cases. Four sample scenarios are shown: for landscapes C1 and C2 (see Figure C.1), the upper and lower rows represent abundances using absorbing and non-absorbing boundary, respectively. For each scenario, the results are derived from 50 replicated runs. The maximum, minimum, and average represent the maximum, minimum and average abundance values obtained across all 50 replicated runs in each time-step, respectively. The average case is used in all simulations reported in this study. Note that the scales along y-axes of the sub-figures are purposefully modified (zoomed in) to highlight the actual differences between the three cases.
clumped houses have comparatively far smaller average distance to their nearest edges in the landscape (see Figure C.1 for the landscapes). As a result, female mosquitoes that start moving around from these houses find an edge much quicker (and thus being killed) than those which start from other houses. Thus, just due to using an absorbing boundary, more mosquitoes die out due to the additional unrealistic killing effect imposed by the absorbing boundary. This suggests the importance of using a non-absorbing boundary in the ABM to avoid the potential bias created by a specific boundary type.

6.8 Results: Impact of ITNs

Impact of ITNs in isolation on mosquito abundance is shown in Figures 6.8 and 6.9 using household-level partial coverage (with multiple chances for host-seeking) and complete coverage, respectively (the reproduced landscape from GN-ITN [85] used for ITN application is shown in Figure C.2). For brevity, we show the 100 days results (day 100 - day 199), with ITNs being applied on day 100; the full one-year results are shown in Figures C.5-C.7. Figure 6.9 shows PR values in abundance obtained by applying ITNs for household-level partial coverage (with multiple chances) and complete coverage for host-seeking.
Figure 6.8. Impact of ITNs on Mosquito Abundance, using Household-Level Partial Coverage with Multiple Chances for Host-Seeking.

The figure depicts our results of replicating GN-ITN [85]. Each row represents a specific coverage (C) for ITNs (e.g., C = 0.8). Each column represents a specific repellence (R) for ITNs (e.g., R = 0.5). Within each sub-figure, each color-coded plot represents a specific mortality (M) value for ITNs (e.g., M = 0.25), with mortality (M) color keys at the bottom of the figure. The x-axis denotes simulation time (in days), and the y-axis denotes mosquito abundance. For brevity, we show the 100 days results (day 100 - day 199). The figure represents averages of a total of 3000 (4 × 3 × 5 × 50) simulations. A non-absorbing boundary is used. For the partial coverage schemes, see Figures 6.1, 6.2 and Section 6.3.
Figure 6.9. Impact of ITNs on Mosquito Abundance, using Household-Level Complete Coverage. The figure depicts our results of applying ITNs (in isolation) with complete coverage as we replicate the results of GN-ITN [85]. Within each sub-figure, each color-coded plot represents a specific mortality (M) value for ITNs (e.g., M = 0.25), with mortality (M) color keys at the bottom of the figure. For brevity, we show the 100 days results (day 100 - day 199); the full one-year results are shown in Figure C.7. The figure represents averages of a total of 3000 (4 × 3 × 5 × 50) simulations. A non-absorbing boundary is used. For other details, see legend of Figure 6.8. For the complete coverage scheme, see Figure 6.3 and Section 6.3.
Figure 6.10. Percent Reductions in Mosquito Abundance by ITNs, Applied in Isolation. We compare household-level partial coverage (with multiple chances for host-seeking) and complete coverage. The x-axis denotes ITN coverage, and the y-axis denotes ITN mortality. The upper row represents household-level partial coverage, and the lower row represents household-level complete coverage, as marked on the left. Each column represents a specific repellence (R) value, as marked on the top. ITN is applied at day 100 in the 40 × 40 grid-based landscape (see Figure C.2) with 50 houses having a total human population of 185. The percent reduction (PR) values, represented as filled contour plots in each sub-figure, are calculated from data used in Figures 6.8 and 6.9. The colorbar on the right quantifies the PR isolines.
The two partial coverage schemes with single or multiple chances produce little difference when compared (see Figures C.5 and C.6). Searching for a host within the same house for \( n \) times does not give much leverage to the mosquito, because each time, if the randomly-picked host is protected by a bednet, the risk of being exposed to the insecticidal effect of ITNs (and thus getting killed) still exists. Since we replicate GN-ITN which is based on partial coverage, we compare their abundance results with Figure 6.8 (which shows household-level partial coverage with multiple chances for host-seeking). We find that with coverage \( C \geq 50\% \), abundance is reduced from 4000 to \( \approx 2000 \), as seen in sub-figures 4-12 in Figure 6.8. This seems more plausible as opposed to achieving a 100\% reduction in abundance as was shown by GN-ITN, because with the partial coverage scheme, since only 54\% of the human population are protected by bednets, a portion of the mosquitoes can still find enough bloodmeals, and hence we cannot expect the mosquito population to be suppressed completely.

With household-level complete coverage scheme, abundance is reduced from 4000 to \( \approx 1000 \) when coverage is in the range \( 60\% < C \leq 80\% \) and repellence \( R \) is not too high (see sub-figures 7 and 8 in Figure 6.9). As \( C \) approaches 100\%, i.e., complete coverage for all humans, irrespective of repellence, abundance can be completely suppressed, as seen in sub-figures 10, 11, and 12 in Figure 6.9. However, too high repellence (e.g., \( 0.5\% \leq R \leq 0.9\) ), though unlikely to be present in commonly used insecticides in real-world scenarios, can have a detrimental effect on vector control (by increasing abundance) with the same levels of coverage and mortality, but the degree of this negative impact is reduced as coverage increases (see sub-figures 2, 3, 5, 6, 8, and 9 in Figure 6.9, and sub-figures 7–8 in Figure 6.9). As seen in sub-figures 5–7 in Figure 6.9 when \( R \leq 0.5 \), around 60\% \( PR \) can be achieved with coverage and mortality being as low as \( \approx 60\% \) and \( \approx 30\% \), respectively. However, when \( 0.5 < R \leq 0.9 \), to achieve the same \( PR \), we need coverage to be as high as \( \approx 85\% \). Also, \( R = 0.9 \) means 90\% of the host-seeking mosquitoes are driven away from the
house before the ITN mortality can play any role (see the complete coverage flowchart in Figure [6.3]). This is why mortality seems to have less impact in sub-figure 8 than in sub-figures 5–7 in Figure [6.9].

Interestingly, with the complete coverage scheme, even with no ITN mortality, very high \( PR \) (around 80%) can be achieved with high coverage (≈ 90%), irrespective of repellence (see sub-figures 5–8 in Figure [6.9]). With 90% coverage, around 90% of the population sleep under bednets. Since our ABM assumes complete usage of bednets, and the \( An. gambiae \) mosquitoes are almost exclusively anthropophilic and highly endophagic, no bloodmeal can be obtained from sources other than humans, or during daytime. Thus, though no mosquitoes are killed due to ITN mortality, they cannot complete their gonotrophic cycles (because ≈ 90% of the host-seeking attempts fail), and eventually, the mosquito population dies out.

6.9 Results: Impact of Combining LSM and ITNs

To evaluate the impact of applying LSM and ITNs in combination, we use three 40×40 landscapes with varying BML density (see Section [6.6.3]). The results, depicted as \( PR \) values in mosquito abundance, are shown in Figure [6.11] (Ps are calculated as described before). Each sub-figure (1–12) represents a filled contour plot where the isolines are labeled with specific \( PRs \), whose magnitudes are shown in the colorbar on the right. Each row represents a specific mortality (M) value for ITNs (e.g., M = 0.2), as marked on the left. Each column represents a specific BML density, as marked on the top.

Figure [6.11] indicates some interesting observations. First, impact of ITN mortality (M) becomes increasingly important as the BML density increases. Comparing the sub-figures column-wise, increasing ITN mortality (M) has less impact on the landscape with Low BML density than with Medium or High cases. With Low BML density, number of available human hosts are also low, making the number of host-
Figure 6.11. Percent Reductions in Mosquito abundance as a Function of LSM Coverage and ITN Coverage. LSM and ITNs are applied in combination. The x-axis denotes ITN coverage, and the y-axis denotes LSM coverage. Each row represents a specific mortality (M) value for ITNs (e.g., M = 0.2), as marked on the left. Each column represents a specific density of bloodmeal locations (BML density), as marked on the top. ITN repellence (R) is fixed at 0.5. Each simulation is run for one year; both LSM and ITNs are applied at day 100, and continued up to the end of the simulation. Each sub-figure represents filled contour plots where the isolines are labeled with specific percent reduction (PR) values. The colorbar on the right quantifies the PR isolines. The figure represents average percent reduction values of a total of 12000 (3 × 5 × 4 × 4 × 50) simulations. For ITNs, household-level complete coverage scheme is used (see Figure 6.3). A non-absorbing boundary is used. Sample landscapes with the three BML density levels are shown in Figure 6.4.
seeking events much lower than the other two cases. Less host-seeking events in turn mean reduced possibility of a mosquito being in contact with an ITN, and as a result, increasing ITN mortality ($M$) cannot affect $PR$ values as greatly as it can with the other two cases. Also, looking across each row, with $M = 0.0$ (i.e., no ITN mortality) at row 1, as BML density increases, the blueish area which represents 10-40% $PR$ values, also increases, indicating less success with the combined interventions. In this case, mid-range coverages of both ITNs and LSM are required to have moderate $PR$ (≈ 50%). However, as ITN mortality increases in the following rows, the blueish area successively decreases, indicating the importance of at least some ITN mortality being there when ITN is applied.

Increase in ITN mortality ($M$) also influences the general shape of the $PR$ isolines. With High BML density (column 3), as M increases, the shape of the $PR$ isolines follow similar trends, and the blueish areas (10-40% $PR$ values) are successively reduced. This is also seen for Low and Medium densities (columns 1 and 2), with the Low density column having the least impact.

Next, considering the impact of each intervention in isolation (i.e., looking exactly at both the x-axis and y-axis $PR$ values with y=0 and x=0, respectively), on a per-row or per-column basis, the rate of changes in $PR$ is similar across all sub-figures. For example, with Low BML density at sub-figures 1, 4, 7, and 10 (i.e., column 1), looking at the y-axis (i.e., at x=0, meaning when ITN is ignored, and LSM coverage is gradually increased), the isolines of $PR$ values intersect the y-axes at approximately similar intervals (e.g., $PR_{10}$ with LSM coverage ≈ 0.18, $PR_{20}$ with LSM coverage ≈ 0.63, etc.). Similar trends are observed across the x-axis, and also for columns 2 and 3. This ensures that without the presence of the other intervention, both LSM and ITNs, with their respective parameters varied, yield significant impact on abundance, and confirms the first notion discussed before (ensuring the individual efficacy of each intervention).
Next, when ITN Mortality (M) is non-zero (i.e., the bednets are at least partially lethal to mosquitoes), increasing ITN Coverage is more effective in reducing mosquito abundance (i.e., increasing PR) than increasing LSM Coverage, which is observed by the more pronounced increase in PR across the x-axis than up the y-axis. This observation is in agreement with similar results obtained in reducing the basic reproductive number of malaria ($R_0$) by Yakob and Yan [257]. However, as seen in row 1, with non-lethal ITNs ($M = 0$), the efficacy of both interventions approach more equivalency as the BML density approaches from Low to High. Also, comparing the sub-figures column-wise, integrating both interventions yield more synergistic effect as the BML density approaches from Low to High. Again, this trend agrees with similar results obtained in reducing $R_0$ in [257].

Not surprisingly, the increase in PR values indicate more synergistic patterns when all parameters are in effect (i.e., have some non-zero values). For example, looking at sub-figures 5, 6, 8, 9, 11 and 12, where the BML density is Medium to High, and M is in the range 0.2 – 0.8, increasing coverages of both interventions yield more synergistic benefits, as indicated by the more convexity of the PR isolines in general. In these cases, with sufficient number of host-seeking events, and ITNs in action with some mortality (insecticidal effect), both interventions play important roles in reducing abundance and thus increasing PRs.

6.10 Discussion

In this section, we discuss some additional insights obtained from the results presented in Sections 6.7–6.9.

6.10.1 Impact of LSM (in Isolation)

In general, with LSM applied in isolation, our replicated results agree with the major findings by GN-LSM [86] that LSM coverage of 300 m surrounding all houses
can lead to significant reductions in abundance, and, while targeting aquatic habitats to apply LSM, distance to the nearest houses can be an important measure. However, as shown by our model, some of the underlying assumptions in the GN-LSM model could have seriously affected their predicted outcomes. To be specific, reporting results from a single simulation run and the use of an absorbing boundary could lead to substantially different results, invalidating the findings and thereby diminishing the predictive power of the models. Also, without a more sophisticated spatial metric that can capture the interrelations of different resources in different landscapes, simplistic features such as the general arrangement pattern of houses (e.g., diagonal, horizontal, and vertical) are insufficient to capture a landscape’s potential to transmit the disease. For example, comparing the most restrictive cases (T3) of LSM application, the reduction in abundance is more prominent with a non-absorbing boundary (from $\approx 10000$ to $\approx 1800$, as shown in sub-figure 6 of Figure 6.6) than with an absorbing boundary (from $\approx 3000$ to $\approx 500$, as shown in sub-figure 6 of Figure 6.5). Due to the random distributions of houses and aquatic habitats in the three selected patterns, the reduction effects remain unpredictable, depending on factors such as the proximity of the resources to the boundaries of the landscapes. When applied to different (e.g., more general or specific) conditions, these assumptions may produce misleading results. We provide additional insights by improving upon, and extending, some of these assumptions.

It is implausible to expect 100% reductions in abundance even with the most restrictive application of LSM (T3 in Figures 6.5, 6.6, and Table 6.5). This is because even with an absorbing boundary, some mosquitoes would always survive by roaming around in different parts of the landscape, instead of hitting the edges of the boundary (and hence dying out). This is observed in our results - the highest $PR$ value obtained is 91.79% with scenario T3 using an absorbing boundary, as opposed to 100% observed in several cases in the GN-LSM study [86].
In few cases, we obtain negative $PR$ values (see Table 6.5), suggesting that the abundances actually increase after applying LSM. A closer look at the landscapes (see Figure C.1) reveals that these cases are associated with the removal of a small fraction of all aquatic habitats (4 out of 90 for C1 and T1) by LSM. Recall that in our ABM, abundance is governed by the $CC$ of aquatic habitats and the density-dependent oviposition mechanism. Removal of only a few nearby habitats may actually save a mosquito from wasting time trying to search, locate, and compete in laying eggs in the already-crowded habitats, and instead be more productive by finding comparatively less-crowded habitats which are within close vicinity.

This points to an important insight: if the mosquito population in the environment is not unrestricted (i.e., it is restricted to be within the limit of the environment’s overall capacity, as in our ABM), and some stages of the mosquito biology are governed by special mechanisms (e.g., density-dependent oviposition), then removal of only an insufficient number of aquatic habitats may, in some cases, increase the abundance. Thus, before actually applying LSM, it may be crucial to estimate its impact (to achieve the desired level of success) by simulating varying levels of coverage.

The choice of using fixed- vs. age-dependent DMRs can have significant impact on the results, which can be compounded by the choice of boundary type as well. As stated before, we use fixed DMRs of 0.2 for the absorbing boundary, and age-dependent DMRs (for all Adult stages and the Larva stage) for the non-absorbing boundary. We get unbounded abundance by using DMRs $< 0.2$ for the absorbing boundary, and by using any fixed DMR other than age-dependent DMRs for the non-absorbing boundary (these results are not shown).

6.10.2 Impact of ITN (in Isolation)

As expected, with ITNs, different definitions of ITN coverage can lead to significantly different results. The household-level partial coverage schemes can provide
only \(\approx 50\%\) reduction in abundance with 100\% coverage and 100\% mortality. This means that even when each house is equipped with one bednet (which, overall, covers only \(\approx 54\%\) of the human population), this scheme cannot perform even anywhere close to suppress abundance. On the other hand, the household-level complete coverage scheme can provide as much as 70\% reduction in abundance with \(\geq 85\%\) coverage and mortality as low as 25\%. With this scheme, when we have 100\% coverage, abundance can be completely suppressed even when no mortality is in action (i.e., \(M = 0.0\)), as shown in sub-figures 10–12 in Figure 6.9. This is expected: since every person in every house is protected by bednets, the host-seeking mosquitoes cannot find unprotected hosts to obtain bloodmeals. While modeling the impact of ITNs, we recommend that these distinctions should be clearly marked, and the choice of the ITN coverage scheme should be made carefully.

In general, repellence, which drives the host-seeking mosquito away from a house, can have a detrimental effect on vector control when the risk (additional delay in search etc.) of finding an unprotected host in another house is less than that in the same house. With the complete coverage scheme, since every person in the house (with ITN coverage) is protected by bednet, the above turns out to be true. However, as coverage \(C\) increases, more houses fall within the range of coverage, and the probability of finding an unprotected host (in another house) during the next search decreases. Thus, with increasing coverage, the negative impact incurred by too high repellence gets reduced, as evident in the first three rows (sub-figures 1–9) of Figure 6.9.

On the other hand, with household-level partial coverage schemes (both with single or multiple chances), this effect is almost absent (see Figure 6.8 sub-figures 1–4 of Figure 6.9 and Figures C.5|C.6). Recall that with partial coverage schemes, every person in the same house (with ITN coverage) may not be protected by a bednet. Thus, the mosquito may find an unprotected host in the same house. If
it is repelled too often (due to high repellence), it is being deprived of the current positional advantage, and the risk of finding an unprotected host in another house may not be well-justified.

Interestingly, the use of a specific boundary type does not have significant impact for this particular landscape (see Figure C.2). Using absorbing and non-absorbing boundary, we simulate and compare three schemes of ITN coverage (see Section 6.6 for the schemes). No significant difference is found if age-dependent DMRs are used with both boundary types (using fixed DMRs, which is not practical for our density-regulated ABM, produces different results, which are not shown).

6.10.3 Impact of LSM and ITNs (in Combination)

While applying LSM and ITNs in combination, some synergistic effects are observed in our results. However, as shown in Figure 6.11, the combined impact is additive (and not multiplicative), and combining both interventions is more effective with high BML density, confirming similar findings in [257].

With higher BML density, impact of ITN mortality (M) becomes increasingly important. As shown in Figure 6.11, increasing ITN mortality affects the shape of the low-to-medium range ($10 - 40\%$) $PR$, represented by the blueish area in each subplot. With no insecticidal effect of ITNs (i.e., $M = 0.0$), looking at row 1, as BML density increases, more host-seeking events occur, causing more mosquitoes to seek for aquatic habitats in order to lay eggs. But with increasing LSM coverage, they are denied more opportunities to lay eggs (as more aquatic habitats are eliminated), causing the blueish area to reduce vertically (down the y-axis). However, as both BML density and ITN coverage increases (but mortality still remains 0), more host-seeking events actually encounter ITNs, but with no mortality in effect, ITNs cannot have significant impact, thus extending the blueish area horizontally (across the x-axis). As ITN mortality increases (in rows 2–4), this extension effect is gradually
reduced, and more impact is seen with higher BML density.

6.10.4 Replication of Earlier Models

Replication of earlier ABMs (that examined the impact of LSM and ITNs in isolation) pose some unique challenges. The unavailability of source codes of the original models inhibits us from performing direct model-to-model comparison (docking). The structural characteristics of ABMs, which are fundamentally different from, for example, equation-based mathematical models, also rule out the possibility of systematic verification of model features, and draw some important V&V issues. We identify the following major sources from which model differences may arise and/or the process of replication becomes more time-consuming and difficult:

- **Conceptual image of the model**: the intended logical view of the ABM may be perceived differently by different modelers, thus creating different conceptual, mental images of the logical view.

- **Choice of tools**: selection of programming languages and tools (e.g., C++ vs. Java) from the numerous options offered these days may be another potential source. The availability and limitations of a particular programming language, use of specific data structures and other language constructs, and even the coding style of individual modelers, can compound the differences.

- **Availability of additional resources**: in some cases, additional resources used by the model (in the forms of artificial maps, object-based landscapes, etc.), if not defined or made explicitly available, pose subtle challenges. Although the importance of these resources may seem somewhat arbitrary in the broader context, goals and output of the original models, for replication, their precise specification still remains important. For example, as we show before, in replicating the landscapes, the absence of a listing of the spatial coordinates of the objects (which may be provided as supplementary materials), not only forces future modelers who try to replicate the landscapes to spend a significant amount of time in reproducing the landscapes (some part of which inevitably rely on best guesses, due to the lack of additional information), it also increases the possibility of augment errors being introduced in this phase.
6.11 Recommendations for Future Modelers

In this section, we provide some recommendations and guidelines for future ABM modelers of malaria, based on several challenges we faced while replicating earlier models. Clear, detailed description of the parameter space for all model parameters used by the ABM, including their initial and other time-varying conditions, may substantially help in minimizing the conceptual image gaps. However, as our past experience shows [8, 9], merely stating model parameters, logical flowcharts, initial conditions, etc. cannot entirely eliminate the above problems, primarily because: (1) the possibility of different logical workflow paths in the programmed code still remains open, and (2) many implementation details still cannot be covered. Hence, based on our modeling exercise presented in this chapter, we recommend the following:

- **Code and data sharing**: The source code and executable programs of the ABMs should be shared with the research community. The trends of open-access research have become increasingly important and popular in recent years. To ensure a minimum standard of reproducibility in computational sciences, enough information about methods and code should be available for independent researchers to reach consistent conclusions [1184]. Many reputed journals across multiple disciplines have also implemented different code-sharing policies. For example, the journal *Biostatistics* [26] has implemented a policy to encourage authors of accepted papers to make their work reproducible by others. In this journal, based on three different criteria termed as “code”, “data” and “reproducible”, the associate editor for reproducibility (AER) classifies the accepted papers as C, D, and/or R, respectively, on their title pages [183, 184]. As reproducibility is critical to tracking down the bugs of computational science, code-sharing may be specially important for malaria ABMs. Having multiple research groups examining the same model and generating new data or output may lead to robust conclusions [201]. Some recent malaria models have partially followed this path by providing controlled access to their models. For example, the OpenMalaria epidemiology model [177] provides a general open-access platform for comparing, fitting, and evaluating different model structures. The EMOD vector ecology model, from Intellectual Ventures Lab [61], is available within controlled execution environments. However, we acknowledge that for certain reasons (e.g., during preliminary design and development phases, exploratory feature testing phases, etc.), it may not always be the ideal case to share the source code. In these cases, we recommend that for ABM-based studies which are accepted for publication, at least the associated executable
programs and other tools be made available as supplementary materials (we share our spatial ABM, a sample input file, and the landscape generator tool, with clear instructions on how to run each, with the submitted version of this work in [11]).

• **Relevant documentation**: Modelers who share the source code and/or executable programs of their ABMs should also provide well-written documentation. Documentation is an important part of software engineering. The journal *PLOS Computational Biology*, which publishes articles describing outstanding open source software, emphasizes that the source code must be accompanied with documentation on building and installing the software from the source, including instructions on how to use and test the software on supplied test data [187]. An ABM documentation may include statements describing the attributes, features, and characteristics of the agents and environments of the ABM, the overall architecture or design principles of the code, algorithms, and application programming interfaces (APIs), manuals for end-users, interpretation of additional materials (e.g., object-based landscapes), etc. Free and commercial software tools are available which can help automating the process of code annotation, code analysis, and software documentation [115, 208, 142, 149].

• **Standardized models**: The general workflow of the ABM, including the input/output requirements, program logic, etc. should follow a standardized approach. The need for standardization becomes more important when the broader utility of the model is considered within an integrated modeling platform. For example, both OpenMalaria [177] and EMOD [61] are currently being integrated within the open-access execution environment of the Vector Ecology and Control Network (VECNet) [241]. The proposed VECNet cyberinfrastructure (VECNet CI), within a shared execution environment, establishes three modes of access sharing for model developers: (1) shared data: model developers run their models on their own compute resources and upload the output data to the VECNet CI for public consumption; (2) shared execution: model developers share their software with VECNet CI developers only, allowing the CI and its operators to incorporate their model into the CI execution environment; and (3) shared software: model developers share their software at large with the public. Once integrated, these models can utilize other components of the VECNet CI, including the VECNet Digital Library, web-based user interface (UI), tools for visualization, job management, query and search, etc. in order to, for example, import and use malaria-specific data to run specific scenarios or campaigns of interest, and display their output using the visualization and/or the UI tools of the VECNet CI. We envisage that most malaria ABMs, in future, will be accommodated within the integrated modeling frameworks of similar cyberinfrastructure platforms. Hence, to expedite the integration process, we recommend that future malaria ABMs plan and follow a well-defined integration path from the early phases of model development.
6.12 Summary

In this study, we explore the individual and combined efficacy of applying LSM and ITNs by using our spatial ABM of malaria, which was described in detail in Chapter 5. To assess our results with earlier work in the field, we replicate the results of two earlier studies that explored similar research questions [86, 85], and present a systematic comparison with our results. By extending and improving upon some of the original assumptions (e.g., reporting results from single simulation runs, use of an absorbing boundary, etc.), we show that the use of these assumptions may lead to less reliable results. With the combined application of LSM and ITNs, our results indicate that varying densities of the human population can affect the degree of synergistic benefits that may be obtained from such efforts, as was previously shown by a mathematical model [257]. To the best of our knowledge, this is the first ABM-based study to explore this particular combination of LSM and ITNs (acknowledging that some other combinations were explored by other ABMs, e.g., [57]). We also discuss some challenges faced while replicating earlier models, and recommend several guidelines (code and data sharing, relevant documentation, and standardized models) for future ABM modelers of malaria from our experience.

As our results indicate, replicability of the experiments and simulations performed by malaria models published earlier bear special importance. Due to several factors (including new tools and technologies, massive amounts of data, interdisciplinary research, etc.), the task of replication may become complicated. By sharing our ABM and the landscape generator tool with the submitted version of this work, we emphasize the importance of open source software for reproducibility and replicability.

In the future, we plan to model seasonality and other weather parameters (e.g., humidity), alternative hosts for blood-feeding (e.g., cattle), aquatic habitats with varying carrying capacities to reflect the variability of habitat attractiveness and productivity, and temporal variability for certain intervention parameters (e.g., re-
pellence and insecticidal effect of ITNs). We also plan to calibrate the assumptions and parameters of the model against data from field-based studies, and explore the impact of other existing interventions (e.g., IRS, space spraying, etc.), or new interventions (e.g., spatial repellents and/or insecticides, oviposition traps, etc.), both in isolation and in combination. Lastly, we plan to improve VectorLand to aid in generating operational guidelines for targeting of aquatic habitats and houses, and thus to perform a systematic study of the effect of spatial distribution of habitats.

Our study has shown that by considering current vector control intervention tools, ABMs can help us to understand the limits of current strategies, as well as provide valuable guidelines to explore novel, alternative future tools (e.g., spatial repellents). As multiple, combined interventions are deployed targeting different stages of the mosquito’s life-cycle, model predictions may prove useful in identifying combinations of interventions that interact synergistically, as shown by several studies (e.g., [257], [83], etc.).
CHAPTER 7

INTEGRATING THE ABM WITH A GEOGRAPHIC INFORMATION SYSTEM (GIS)

7.1 Overview

As we mentioned in Chapter 1, the first part of this dissertation includes an example application of the ABM by integrating data from a geographic information system (GIS), and analyzing some spatial metrics of malaria transmission. In this chapter, we describe the application, and the general workflow to integrate the GIS with our spatial ABM, which was described in Chapter 5.

A geographic information system (GIS) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographic data. GIS can be thought of as a system because it digitally creates and manipulates spatial data. A GIS allows users to create interactive queries (user-created searches), analyze spatial information, edit data in maps, and present the results of all these operations.

The idea of integrating GIS with ABMs is not new. Several studies, ranging across multiple domains, have shown such integration. For example, Brown et al. [29] addressed the coupling of GIS-based data models with agent-based process models, and analyze different requirements for integrating ABM and GIS functionality. They illustrate the integration approach with four ABMs: urban land-use change, military mobile communications, dynamic landscape analysis and modeling system, and infrastructure simulations.

\[\text{Major portions of this chapter appeared in Arifin et al. [10].}\]
In general, GIS and spatial statistical methods have been extensively used in entomological and epidemiological studies. In particular, for malaria as a disease, GIS applications have been used for measuring the distribution of mosquito species, their habitats, the control and management of the disease, etc. Using GIS and other geo-spatial applications, several studies have modeled malaria transmission as a function of resource availability, measured the spatial variation and socio-economic determinants of the malaria parasite *Plasmodium falciparum* infections, measured the spatio-temporal effect on malaria transmission intensity as it decreases with altitude due to its dependence on environmental and weather/climate variables, identified the increased risk factors of malaria in association with the spatial determinants, and so on. For example, Gimnig, Hightower, and Hawley [77] discussed the application of GIS to the study of mosquito ecology and mosquito-borne diseases, including malaria. Khormi and Kumar [118] presented a review of mosquito-borne diseases, with examples of the use of spatial information technologies to visualize and analyze mosquito vector and epidemiological data.

However, no model-based malaria study has yet shown how to effectively integrate an ABM with GIS and other geo-spatial features, and thereby harness the full power of GIS. There is also a *vacuum of knowledge* in building robust integration frameworks that can guide the use of ecological, geo-spatial, environmental, and other types of features (related to malaria transmission) as model inputs, as opposed to simply use these features as cartographic outputs from the models (as done by most previous studies).

In this chapter, we show how to effectively integrate a spatial ABM of malaria vector mosquitoes with a GIS. For a specific study area (Asembo, Kenya), we identify the relevant data layers, and collect, analyze, and prepare the data for the ABM. We rank different aquatic habitat types based on their characteristics. Then, we assign hypothetical carrying capacities to the habitats, and build two hypothetical scenarios.
Once the ABM is run with both scenarios, we analyze custom spatial variables (outputs of the ABM), which include adult abundance by location, cumulative biomass per aquatic habitat, cumulative number of females oviposited per aquatic habitat, and cumulative number of bloodmeals per house. Finally, we produce GIS maps by overlaying the spatial variables on top of the relevant data layers, and analyze important biological insights as discovered from the maps.

The organization of this chapter is as follows: Section 7.2 describes several malaria-related studies that use GIS and other geo-spatial features. Section 7.3 briefly describes the ABM, the study area, and the GIS-ABM Workflow. In Section 7.4, we describe, in details, the processing steps of the GIS data layers. Section 7.5 describes two hypothetical scenarios created for the ABM. Section 7.6 lists the assumptions of the simulations, and describes the four custom spatial variables produced as outputs of our spatial ABM. Section 7.7 describes our results, and Section 7.8 concludes. Some additional materials for this chapter are presented in Appendix D.

7.2 Use of GIS in Malaria Research

In this section, we discuss several malaria-related studies that use GIS, Global Positioning System (GPS), spatial statistical methods, geo-spatial features, etc. In general, GIS has been extensively used in epidemiological studies. In particular, for malaria as a disease, GIS has been used for measuring the distribution of mosquito species, their habitats, the control and management of the disease, etc. GIS and spatial statistical methods are regarded as important tools in epidemiology to identify areas with increased risk of diseases, and determine spatial association between disease and risk factors [163].

Mbogo et al. [156] studied the seasonal dynamics and spatial distributions of *Anopheles* mosquitoes and *P. falciparum* parasites along the coast of Kenya. Using hand-held GPS, they recorded latitude and longitude data at each site, and produced
the spatial distribution maps for three *Anopheles* species. Li, Bian, and Yan [137] presented a spatially distributed mosquito habitat modeling approach, integrating a Bayesian modeling method with Ecological Niche Factor Analysis (ENFA) using GIS. They used data for seven environmental variables to represent the environmental conditions of larval habitats in the Kenya highlands. The Malaria Atlas Project (MAP) developed the science of malaria cartography by modeling the global spatial distribution of *P. falciparum* malaria endemicity [93, 97]. Focusing on the spatial heterogeneity of malaria transmission intensity, they effectively produced and used maps as essential tools for malaria control [97].

Zhou et al. [261] used GIS layers of larval habitats, land use type, human population distribution, house structure, and hydrologic schemes, overlaid with adult mosquito abundance, to investigate the impact of environmental heterogeneity and larval habitats on the spatial distribution of adult *Anopheles* mosquitoes in western Kenya. Mmbando et al. [163] conducted a study of four cross-sectional malaria surveys in 14 villages located in highland, lowland, and urban areas of northeastern Tanzania during the rainy seasons. Their results show a significant spatial variation of *P. falciparum* infection in the region, identifying altitude, socio-economic status, high bednet coverage, and urbanization as important factors associated with the spatial variability in malaria. Ndenga et al. [170] used a GPS unit to classify aquatic habitats within highland sites in western Kenya. They recorded the latitude, longitude, and altitude of the habitats, and classified them as natural swamp, cultivated swamp, river fringe, puddle, open drain or burrow pit, and showed that the productivity of malaria vectors from different habitat types are highly heterogeneous.

7.3 The Study Area and the GIS-ABM Workflow

In this section, we describe the study area, the GIS-ABM workflow, and the selected GIS data layers, which can be broadly classified into two categories: aquatic
habitats and houses.

7.3.1 The Study Area

For this study, a village in Kenya’s in Rarieda Division in Nyanza Province, known locally as Asembo, is chosen as the study area (see Figure 7.1). Asembo is located within a subsection of the Siaya and Bondo Districts in western Kenya. It covers an area of 200 \( km^2 \), with a population of approximately 60,000 persons (according to estimates from the 1989 Kenya Government census statistics) \[186\]. Asembo includes a study site of 15 villages (with an area of approximately 70 \( km^2 \)) near Asembo Bay, and experiences intense, perennial malaria transmission \[167\].

The primary reason for selecting Asembo as our study area is the availability of data from the Asembo Bay Cohort Project \[157\] and the Asembo ITN project \[186\]. The Asembo ITN project, published in 2003 as a journal supplement containing a series of 23 articles, reports important public health findings from a successful trial of ITNs in western Kenya \[228\]. Research findings reported in the supplement provide substantial evidence that high coverage of ITNs in the study area will result in significant health benefits for affected communities \[167\]. A map of the study area, with the projected boundary for Kenya, is shown in Figure 7.1.

7.3.2 The GIS-ABM Workflow

The GIS module, using the ArcGIS software \[4\], produces, processes, and analyzes the relevant data layers, and converts them into plain-text ASCII format. The ASCII files are then converted into XML format by using a customized Java program (the Input Formatter), and fed as input to the spatial ABM (see Chapter 5). Once the ABM completes the simulations, the output are analyzed by using a customized Perl module (the Output Analyzer). Plots and other figures are then produced from the analyzed output. To perform the spatial analysis with GIS, ASCII files, from the
Figure 7.1. The Study Area of Village Clusters in Asembo, Kenya. The figure shows the location of village clusters in Asembo within the boundary of entire Kenya; the map unit (for the data frame properties of the projection system in the shapefile) is set to Meters. Asembo lies on Lake Victoria in Kenya’s Nyanza Province. The Asembo Bay has a human population of about 100,000.
Figure 7.2. The GIS-ABM Workflow. The GIS module, using the ArcGIS software [4], produces, processes, and analyzes the relevant data layers, and converts them into plain-text ASCII format. The ASCII files are then converted into XML format by using a customized Java program (the Input Formatter), and fed as input to the spatial ABM (see Chapter 5). Once the ABM completes the simulations, the output are analyzed by using a customized Perl module (the Output Analyzer). Plots and other figures are then produced from the analyzed output. To perform the spatial analysis with GIS, we produce new ASCII files with generated values from the analyzed output, and feed them into the GIS module. Finally, the GIS module produces spatial maps with relevant information portrayed on top of the data layers (e.g., adult abundance by location).

analyzed output, are also produced, and fed into the GIS module. Finally, the GIS module produces spatial maps with relevant information portrayed on top of the data layers (e.g., female adult abundance by location). The GIS-ABM workflow is shown in Figure 7.2.

7.4 GIS Data Layers

In this section, we describe the processing steps of the GIS data layers in details. The GIS data layers represent several thematic layers of the study area that are relevant to our spatial ABM. These layers fall into two categories: aquatic habitats
and houses. The aquatic habitat types include two types of mosquito breeding sites (type-1 breeding site and type-2 breeding site), boreholes, pit latrines, and wetland. A type-2 breeding site is composed of a type-1 breeding site and several other data points (e.g., compounds, boreholes etc.). Boreholes, also known as borrow pits, have great potentials as breeding sites in this area, and represent hole or pits made in the ground when local people use clay or soil for building houses, making pots, etc., thereby leaving depressions in the ground that easily get filled with rain water. Pit latrines are very common to households in the study area. The wetland represents a stretch of wetland lying to the north-west corner of the study area. Human houses serve as bloodmeal locations for the mosquitoes, and include houses, huts, etc.

7.4.1 Processing the Data Layers with GIS

We start with the feature identification and extraction process for entire Kenya. Then, we describe the data scale down process to the study area of Asembo, followed by the selection process of a subset of villages within Asembo, and finally, to the selection process of a polygon within the village clusters, which is used as input to our spatial ABM.

We first identify and extract different water features (rivers, wetlands and several water-points) and villages (including human houses) projections for all over Kenya, as shown in Figure 7.3. Different water features (rivers, wetlands, etc.) and villages are projected to the projection system used in Figure 7.1. In Figure 7.3, the figure on the left shows different water features (rivers, wetlands and several water-points) all over Kenya, and the figure on the right shows villages for Kenya.

Two different shapefiles for the wetland in Kenya are projected to the selected projection system. One of the wetland files possesses attributes for types of wetlands (e.g., containing lakes, mangroves, marshes, etc.). The villages are also projected similarly. Different water features and villages are then clipped within the outside
Figure 7.3. The Water Sources and Villages Projections for Kenya. The figure on the left shows different water features (rivers, wetlands, and other types of water-points) for entire Kenya. The figure on the right shows villages for Kenya.

boundary of Kenya. The clipped features are shown in Figure D.1.

7.4.2 Selecting Aquatic Sites for the Study Area

We collect water features data for different types of aquatic sites. Each water source is assigned a unique ID, which we call $ID_{data-feature}$. Once we thoroughly examine the water sources data layers, we encounter some overlapping problems among the data layers. To overcome this, we decide to rank the data layers according to precedence, by sub-grouping water source layers based on their attributes, and creating new shapefiles. In the process, we combine similar types of water features in the same data layer.

The selected subset of data layers for different water features is shown in Figure 7.4. We use the Select By Attributes tool (with SQL query) in the attribute table to
Figure 7.4. Selecting Subset of Data for Kenya Water Sources. The figure shows different water features all over Kenya.

select features from each data layer, and export the selected features to create new shapefiles in ArcGIS [4]. We also assign new IDs for each type of water feature by using the Add Field to create the new ID field in the attribute table of each water source data layer. Lastly, we use the Field Calculator tool to calculate value for the \( ID_{data\_feature} \) field.
7.4.3 Scaling Down to Village and Household Level

Since for the spatial ABM we need household level data that include water features available near the houses, we scale down the data to selected village cluster area from the entire Kenya boundary area. We select a village cluster in Asembo, based on higher frequency of availability of aquatic sites (boreholes, streams, breeding sites, etc.) near households. After analyzing the water features for all over Kenya, we also discover some wetlands and rivers features in the selected area. Figure 7.5 shows the selected village cluster area with houses and all water features located in the study area. We use the Select by Location tool in ArcGIS [4] to select the features.

For reasons of performance and complexity (for example, large number of features), we further select a subset of villages from the village cluster area (which was shown in Figure 7.5). The ABM, without explicit parallelization or multiple runs, can handle a landscape with maximum dimensions of 95 columns × 96 rows. To reflect the available field data that points to limited flight ability and perceptual ranges of Anopheles mosquitoes, each cell in the landscape is set to 50 m × 50 m, yielding a total area of ≈ 25 km². Thus, we need to further scale down the area, and select a polygon (of ≈ 25 km²). Figure 7.6 shows the selected polygon (containing a subset of villages), outlined in magenta. The process of creating the polygon is depicted in Figure D.2.

Next, we clip (crop) the aquatic habitats and houses within the outside boundary of the polygon. The clipped features include wetlands, streams, borehole, breeding sites, and pit latrines. We choose to eliminate the stream and river features, which, being moving (non-stagnant) water sources, are usually not considered as prospective breeding sites for Anopheles mosquitoes. The result is shown in Figure D.3.
Figure 7.5. Selected Village Cluster Area in Asembo, Kenya. The figure shows all houses and water features located in the study area.
Figure 7.6. Selected Polygon within the Village Clusters in Asembo, Kenya. The whole area depicts the selected subset of villages. The selected polygon, outlined in magenta, represents 25 km$^2$, which is transformed to a 95 columns $\times$ 96 rows landscape for the spatial ABM.
7.4.4 Conversion of Data Formats

Since the ABM needs data in the ASCII format, we first convert the selected layers to raster grid format (using the Polygon to Raster tool in ArcGIS [4]). The cell size for raster conversion is set to \(50 \text{ m} \times 50 \text{ m}\), with the value field set to \(ID_{\text{data-feature}}\). All point shapefiles for type-1 breeding sites, type-2 breeding sites, boreholes, pit latrines, and houses are converted using the Point to Raster tool in ArcGIS [4]. The shapefile for pit latrines is created from the shapefile for houses (since pit latrines are usually found inside household boundaries). We also assign separate \(ID_{\text{data-feature}}\) for each feature type.

Due to the resolution (cell size), more than one feature type may fall in a single cell. In these cases, to calculate the number of features (of each type) in each cell, we set the Cell Assignment Type as SUM, since it sums the attributes of all points in the cell. Thus, it acquires the summation of the value fields (of \(ID_{\text{data-feature}}\)), and helps us to determine the number of features. The raster format conversion process is depicted in Figures D.4 and D.5. Next, we set the extent for the conversion as the boundary coordinates of the polygon area shapefile (see Figure 7.6), and convert the raster files to ASCII format using the Raster to ASCII tool in ArcGIS [4].

7.4.5 Generating the Study Area for the ABM

Finally, we generate the study area for the ABM, which is shown in Figure 7.7. In the figure, Map 1 shows the study area polygon for the ABM, outlined in magenta. The same polygon, within the village cluster area of Asembo, is shown in Map 2. In Map 3, the village cluster is marked in red circle within the entire map of Kenya. The figure clearly identifies the comparative scale down process of the area (as described previously in this section).

We decide to reserve the use of shades of blue for all aquatic habitats, and square brown symbols for the houses. The updated map is shown on the left in Figure
Figure 7.7. Study Area for the Spatial ABM. Map 1 shows the study area for the ABM, outlined in magenta, within the village cluster (shown in Map 2). In Map 3, the village cluster is marked in red circle within the entire boundary of Kenya.
Figure 7.8. Study Area with Selected Data Layers and Grid-Lines for the Spatial ABM. The figure on the left uses shades of blue for all aquatic habitats, and square brown symbols for the houses. The figure on the right depicts grid-lines which are added to the map for ease of visualization.

For easy visualization, grid-lines are also added to the map to every tenth point (starting from 1), as shown on the right in Figure 7.8. The grid-lines are added using the Hawth’s Tool [92], imported to ArcGIS [4].

7.4.6 Generating the Feature Counts for the ABM

From the GIS data layers described above (i.e., house, type-1 breeding site, type-2 breeding site, borehole, pit latrine, and wetland), we generate the feature counts to use as inputs to our ABM. The feature counts, as shown in Table 7.1, appear as 1976 (982 aquatic habitats of different types, and 994 houses).
### TABLE 7.1

GIS FEATURE COUNTS FOR THE ABM

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-1 breeding site</td>
<td>4</td>
</tr>
<tr>
<td>Type-2 breeding site</td>
<td>14</td>
</tr>
<tr>
<td>Borehole</td>
<td>4</td>
</tr>
<tr>
<td>Pit latrine</td>
<td>401</td>
</tr>
<tr>
<td>Wetland</td>
<td>559</td>
</tr>
<tr>
<td>House</td>
<td>994</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1976</strong></td>
</tr>
</tbody>
</table>

#### 7.5 Creating Scenarios for the Spatial ABM

To run simulations with the selected data layers, we create two hypothetical scenarios with different combined carrying capacities (CCCs). As we described in Section 3.6, each aquatic habitat in the ABM is associated with a finite carrying capacity (CC), which is treated as a soft limit on the aquatic mosquito population. As defined in Section 5.2.1, CCC, for a given landscape with one or more aquatic habitats, represents the sum of the CCs of all aquatic habitats.

We assign carrying capacities to the selected GIS layers that represent the aquatic habitats (i.e., type-1 breeding site, type-2 breeding site, borehole, pit latrine, and wetland). However, we cannot obtain absolute CC values for the habitats (due to the lack of data for habitats in the study area). Hence, we assign hypothetical CC values, ensuring that the relative magnitudes of CCs follow the biological reality of the environment. For example, considering a cell in the spatial grid, a large breeding site should have higher CC than that of a small portion of wetland.
In terms of the magnitudes of the assigned CCs, we arbitrarily order the different aquatic habitat types as (in decreasing order of CC per cell): 1) type-1 breeding site, 2) type-2 breeding site, 3) borehole, 4) pit latrine, and 5) wetland. For wetland, which covers multiple cells in the north-west corner of the study area (see Figure 7.8), we assign the same CC value for each cell. In the future, when the data is available, the order, as well as the assigned CC values (to different aquatic habitat types), can be readily changed, and the ABM is ready to run with the newly-assigned values.

To run the ABM with the selected data layers, we create two hypothetical scenarios with different CCCs by assigning hypothetical CCs to the different aquatic habitat types, keeping the order of magnitudes intact. The CCCs for the scenarios appear as 21K and 150K. These two scenarios, depicted in Tables 7.2 and 7.3, may effectively represent two ecological settings with low (21K) and high (150K) potentials for mosquito populations, resembling the dry and rainy weather seasons, respectively, in the study area.

Note that the simplicity in the approach allows to feed different scenarios to the ABM just by using different CCCs for the different aquatic habitat types, and there is no need to change data layers, features, etc., for future simulation runs.

7.6 Simulations

In this section, we list the assumptions of the simulations, and describe the spatial variables that are produced as outputs of the simulations.

7.6.1 Assumptions

We assume the following for the spatial ABM (described in detail in Chapter 5): The model starts with 1000 initial female adult mosquito agents (no male agents).
### TABLE 7.2

**HYPOTHETICAL CARRYING CAPACITY SCENARIO 21K**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Count</th>
<th>Assigned CC</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-1 breeding site</td>
<td>4</td>
<td>1000</td>
<td>4000</td>
</tr>
<tr>
<td>Type-2 breeding site</td>
<td>14</td>
<td>500</td>
<td>7000</td>
</tr>
<tr>
<td>Borehole</td>
<td>4</td>
<td>100</td>
<td>400</td>
</tr>
<tr>
<td>Pit latrine</td>
<td>401</td>
<td>10</td>
<td>4010</td>
</tr>
<tr>
<td>Wetland (each cell)</td>
<td>559</td>
<td>10</td>
<td>5590</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>21000</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 7.3

**HYPOTHETICAL CARRYING CAPACITY SCENARIO 150K**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Count</th>
<th>Assigned CC</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-1 breeding site</td>
<td>4</td>
<td>5000</td>
<td>20000</td>
</tr>
<tr>
<td>Type-2 breeding site</td>
<td>14</td>
<td>2000</td>
<td>28000</td>
</tr>
<tr>
<td>Borehole</td>
<td>4</td>
<td>1500</td>
<td>6000</td>
</tr>
<tr>
<td>Pit latrine</td>
<td>401</td>
<td>100</td>
<td>40100</td>
</tr>
<tr>
<td>Wetland (each cell)</td>
<td>559</td>
<td>100</td>
<td>55900</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>150000</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All new agents (entering as eggs) are female. For each initial female agent, the state is set to *Gravid*, the age is set to 120 hours (for being in the *Gravid* state), and the agent is assigned to an aquatic habitat chosen at random. Since available field data points to limited flight ability and perceptual ranges of mosquitoes, the speed and range of movement (of mosquitoes) in our spatial ABM are controlled by special agent-level variables. As described in Chapter 3, unlike other traditional malaria transmission models, we assume senescence (biological aging) of the mosquitoes, and the ABM implements age-specific mortality rates for the adult mosquitoes and the larvae (i.e., the probability of death for mosquito agents increases with their age).

In order to seek for resources (aquatic habitats or houses), and hence to complete the gonotrophic cycle, the adult female mosquito agents move only while they are in *Bloodmeal Seeking* and *Gravid* states. At any point in the resource-seeking process, a mosquito’s neighborhood is modeled as an eight-directional Moore neighborhood, as described in Section 5.2.3. The landscape is assumed to have a non-absorbing boundary, modeled topologically as 2D torus spaces (see Section 6.1).

### 7.6.2 Spatial Variables

For the two hypothetical scenarios ($21K$ and $150K$), the output of our spatial ABM includes four custom spatial variables:

1. *Adult abundance by location*: shows the distribution of the adult mosquitoes over the entire landscape at the end of the simulation; the distributed abundance is overlayed on top of the aquatic habitats and houses.

2. *Cumulative biomass per aquatic habitat*: overlayed on top of the aquatic habitats, it represents the sum of biomass (eggs, larvae, and pupae) present in an aquatic habitat.

3. *Cumulative number of females oviposited per aquatic habitat*: also overlayed on top of the aquatic habitats, it represents the sum of the number of times female adults oviposited in the aquatic habitat.

4. *Cumulative number of bloodmeals per house*: overlayed on top of the houses, it represents the sum of the number of times female adults obtained bloodmeals in
the house.

The last three spatial variables are sampled across all daily time-steps throughout the entire simulation. The output GIS maps, described in the next section (Section 7.7), are produced by overlaying the above spatial variables on top of the relevant data layers. These variables allow us to analyze spatial correlations, and find spatial patterns from the output of the model.

We use a special GIS map symbolizing technique known as the \textit{graduated symbols}. Graduated symbols, used to compare quantitative values, vary in size according to the relative magnitudes of the values. In all output maps, we use graduated symbols as hollow circles, where the relative radii of the circles are determined by the spatial output values generated by the ABM.

7.7 Results

In this section, we describe the results of simulations. We begin with the results for non-spatial abundance, followed by results for the four spatial variables.

7.7.1 Mosquito Abundance (Non-Spatial)

As we described in Section 3.6 in our spatial ABM, vector abundance depends (among other factors) on the carrying capacities of the aquatic habitats. Hence, not surprisingly, the $150K$ scenario yields much higher abundance than the $21K$ scenario, as shown in Figure 7.9. This also validates the abundance patterns usually observed in the \textit{rainy} season than in the \textit{dry} season.

However, the output maps from all four spatial variables indicate that the relative distances between the aquatic habitats and the houses play a crucial role in determining the variables of interest, as shown in Figures 7.10–7.13. By close inspection of the maps, we find interesting patterns and important biological insights, which are
Figure 7.9. Mosquito Abundance (Non-Spatial). Abundance with two hypothetical scenarios with different combined carrying capacities, $21K$ and $150K$, are shown. These two scenarios effectively represent two ecological settings with low ($21K$) and high ($150K$) potentials for mosquito populations, resembling the dry and rainy weather seasons, respectively, in the study area.

7.7.2 Adult Abundance by Location

Results for adult abundance by location are shown in Figure 7.10 for scenarios $21K$ (on the left) and $150K$ (on the right), respectively. As depicted in Figure 7.10, higher abundances are associated with type-1 breeding sites, followed by type-2 breeding sites. Out of the 4 and 14 breeding sites (of type-1 and type-2, respectively), highest abundances are observed in locations where type-1 sites are in close proximity with type-2 sites, surrounded by human houses. We also observe very low (1-3 mosquitoes per cell) abundances in the wetland locations, which may be attributed to reduced human habitation around the wetland, and the low carrying capacities associated with the wetland cells.
Figure 7.10. Adult Abundance by Location. Overlayed on top of the aquatic habitats and the houses, this spatial variable shows the distribution of the adult mosquitoes over the entire landscape at the end of the simulation. The figure on the left shows results for the $21K$ scenario, and the figure on the right shows results for the $150K$ scenario.
Figure 7.11. Cumulative Biomass per Aquatic Habitat. Overlayed on top of the aquatic habitats, this spatial variable represents the sum of biomass (eggs, larvae, and pupae) present in an aquatic habitat; it is sampled across all daily time-steps throughout the entire simulation. The figure on the left shows results for the 21K scenario, and the figure on the right shows results for the 150K scenario.

7.7.3 Cumulative Biomass and Females Oviposited

Results for cumulative biomass per aquatic habitat and cumulative number of females oviposited per aquatic habitat are shown in Figures 7.11 and 7.12 respectively. In both figures, results for scenarios 21K and 150K are shown on the left and on the right, respectively.

As shown in Figures 7.11 and 7.12, higher values are associated with type-1 breeding sites, followed by type-2 breeding sites which are close to boreholes. However, an interesting insight reveals that 2 (out of 14) type-2 breeding sites, suitable to yield high output (like other type-2 sites), yield only 0.07%-0.8% cumulative biomass, and
Figure 7.12. Cumulative Number of Females Oviposited, per Aquatic Habitat. Overlayed on top of the aquatic habitats, this spatial variable represents the sum of the number of times female adults oviposited in the aquatic habitat; it is sampled across all daily time-steps throughout the entire simulation. The figure on the left shows results for the $21K$ scenario, and the figure on the right shows results for the $150K$ scenario.
only 0.005%-1% cumulative number of females oviposited, when compared to other type-2 sites. Closer inspection of the corresponding output maps (Figures 7.11 and 7.12) indicate that the nearest human houses around these 2 type-2 breeding sites are situated much farther than other type-2 sites. Since there are not enough houses in the close proximity, the female mosquitoes, ovipositing in these breeding sites, cannot find bloodmeals, and hence are forced to search longer distances. Since the mortality rate of mosquitoes increases with their age (recall that the ABM implements age-specific mortality rates that incorporate senescence, or biological aging), the additional delays in obtaining bloodmeals actually reduce abundance around these sites, causing much lower cumulative biomass and cumulative number of females oviposited.

7.7.4 Cumulative Number of Bloodmeals

Lastly, results for cumulative number of bloodmeals per house are shown in Figure 7.13 for scenarios 21K (on the left) and 150K (on the right), respectively. We again find that higher values are associated with houses which have nearby type-1 and type-2 breeding sites, and moderate values are found in houses which have nearby aquatic habitats (of different types) with at least some carrying capacities. Interestingly, a large number of houses, located at the lower left quadrant of the study area, show no bloodmeals, due to the absence of aquatic habitats around them.

7.8 Summary

As our findings suggest, availability of the ecological resources, i.e., the aquatic habitats and houses, and the relative distances between these distinct resource types, are two crucial determinants for the female mosquitoes to complete their gonotrophic cycles. From the viewpoint of mosquito agents, these resources directly define landscape features such as spatial heterogeneity, host availability, etc., the importance of which for vector control have been demonstrated by several studies (see Chap-
Figure 7.13. Cumulative Number of Bloodmeals, per House. Overlayed on top of the houses, this spatial variable represents the sum of the number of times female adults obtained bloodmeals in the house; it is sampled across all daily time-steps throughout the entire simulation. The figure on the left shows results for the 21K scenario, and the figure on the right shows results for the 150K scenario.
ters 5 and 6). Reduced availability of either type of these spatial resources would prolong the gonotrophic cycle of the female mosquito, and potentially affect malaria transmission.

In this study, spatial analysis of the output variables generated by the ABM reveals important biological insights. The use of maps and spatial statistical methods readily allows identifying and displaying interesting spatial patterns, which, without using the maps, are difficult to detect. The output maps also reveal potential hotspots with higher rates for the measured variables of interest.

The proposed robust integration framework also allows easy parameterization of the model. For example, the arbitrary order of the different aquatic habitat types, and the assigned CC per habitat, can be readily changed to suit new scenarios and/or new areas of study. This will allow the spatial ABM to produce site-specific output, without the need of modifying the ABM itself. The simplicity in the scenario-based approach also allows to feed different scenarios to the ABM by using different CCCs for various aquatic habitat types, without requiring to change the data layers, features, etc., for future simulation runs.

Although in this pilot study we handled a comparatively small study area of \( \approx 23 \text{ km}^2 \) (which transforms to a 95 columns \( \times \) 96 rows landscape for the spatial ABM), the methodology described here can be readily extended to include larger areas (e.g., the whole Asembo area). For the new regions to be modeled, either real data can be used, or synthetic/predicted data can be interpolated from a few point regions (on which the described methodology is applied first).

Our results also indicate that disease-specific maps can play important roles in disease control activities, including monitoring the changes of malaria epidemiology, guiding resource allocation for malaria control, and identifying hotspots for further investigation. For example, the results highlight the importance of eliminating the aquatic habitats close to human habitations by means of environmental modifications
and manipulations, supporting the arguments presented by several malaria control studies (e.g., [65]).

We conclude that such integrated approaches, which combine knowledge from entomological, epidemiological, simulation-based, and geo-spatial domains, are required for the identification and analysis of relationships between various transmission variables, as demonstrated by our study. Eventually, such integration efforts may facilitate the Integrated Vector Management (IVM) agenda, promoted by the World Health Organization (WHO) [251], to achieve improved efficacy, cost-effectiveness, ecological soundness, and sustainability of vector control.

This chapter concludes Part 1 of this dissertation. In the next three chapters (Chapters 8–10), we describe Part 2, which involves a multi-dimensional data warehouse for malaria-related data.
8.1 Overview

As we mentioned in Chapter 1, Part 2 of this dissertation (Chapters 8–10) involves the design and implementation of a multi-dimensional data warehouse (DW) for malaria-related data, as part of the VECNet CI [241]. In this chapter, we present a general background of data warehouses and dimensional modeling (DM). Major portions of this chapter are adopted from the recent DW/DM literature, especially from [123, 146]. As we describe various concepts and vocabulary relevant to DW/DM, we draw examples from a Point of Sale (POS) system that deals with retail transactions at customer checkouts.

The organization of this chapter is as follows: In Section 8.2, we define data warehouses, dimensional modeling, business intelligence, and decision support systems. Section 8.3 describes the basic concepts and vocabulary which are commonly used for DW/DM. Section 8.4 describes the four separate components of a DW. Section 8.5 describes the four-step dimensional design process, and Section 8.6 concludes.

8.2 Data Warehouse (DW)

In this section, we define data warehouses, and the general concepts of dimensional modeling, business intelligence, and decision support systems, as they are integrally connected to the concept of DWs. We also discuss the characteristics and goals of a
A data warehouse (DW) is a central repository of data which is created by integrating data from one or more disparate sources. It is usually defined as a collection of subject-oriented, integrated, nonvolatile, and time-varying data to support management decisions [106]. In addition, a high-performance DW also advocates achieving speed, scale, complexity, and concurrency [202]. Data warehouses, also known as online analytical processing (OLAP) systems, flourished during the early 1990s. They can be a viable solution to traditional databases, by offering a better response to the increasing demands of decision-making users. Malinowski and Zimányi [146] define data warehouses as a specific type of database that periodically collects information about activities performed by an organization. This information, accumulated over a period of time, is then used to analyze its evolution trends and to discover strategic information such as correlations, patterns, etc.

A DW usually stores integrated data and definitions, thus serving as an unambiguous source of informational truth, and providing decision-makers a common understanding of the activities [188]. Being a single place serving as the source of all decision-making information, a DW ensures a common set of data for all users, with evolved standards and practices for reporting and interpretation of the information. Collecting data from various operational databases (or operational source systems, as we describe in Section 8.2.3) and other data sources, a DW transforms it into new structures that fit better for performing business analysis tasks [146].

8.2.1 Dimensional Modeling (DM)

*Dimensional modeling (DM)* is a design technique used in DW design. Being different from the entity-relationship modeling (ER) technique (which is usually used to model traditional relational databases), DM views data as consisting of facts
linked to several dimensions (described later) of interest [123, 146], and is at the heart of the OLAP systems [136]. Aggregations are performed to summarize many records with potential filtering or categorization based on attributes of relevant facts and dimensions [188]. Dimensional modeling also has emerged as the only coherent architecture for building distributed data warehouse systems [123].

DM serves as a methodology for logically modeling data for query performance and ease of use. Because of their predictable symmetric nature, DMs have proved to be understandable, extendable, and highly resistant to the ad hoc queries from DW users. DMs are the basis of many RDBMS performance enhancements (e.g., powerful indexing approaches and aggregations), as well as the logical foundation for all OLAP systems.

Dimensional modeling addresses the problem of overly complex schemas in the data presentation area (described in Section 8.4). Traditional normalized models, also referred to as third-normal-form (3NF) models, or entity relationship (ER) models, store data into many discrete entities, each of which becomes a table in the relational database. A dimensional model contains the same information as a normalized model. However, it packages the data in a format whose design goals are user understandability, query performance, and resilience to change.

The roots of the data warehouse industry lie in business intelligence (BI), decision support systems (DSSs), and dimensional modeling (DM). Business intelligence (BI) is a generic term to describe the ability of an organization to leverage its internal and external information assets for making better business decisions [123]. BI technologies provide historical, current, and predictive views of business operations by collecting, maintaining, and organizing large amounts of information.

A decision support system (DSS) is a computer-based information system that supports business or organizational decision-making activities by serving at the management, operations, and planning levels. It provides informed assistance to decision-
makers at various organizational levels for analyzing strategic information [140]. A
DSS can support long-term strategic or short-term tactical decision making [188].
Both of these categories have important applications for malaria control. For ex-
ample, based on existing malaria vector control interventions, a DSS can help in
making decisions about introducing new, novel interventions in a particular location
by considering and analyzing the long-term demographic, entomological, epidemi-
ological and other characteristics of that location (e.g., the incidence and prevalence
data collected and stored for several decades). As an example of short-term decision
making, a DSS can help predict the outcome of modifying and/or increasing the
levels of coverage for existing vector control interventions, or introducing new drugs
or insecticides, for which the immediate impact needs to be measured. DSS is the
term originally used for data warehousing [123]. Typically, a DSS uses a DW as an
inventory of information assets, developed and deployed as an integral part of the
DSS [140]. OLAP systems used in a DSS can perform analytical processing primarily
by analyzing patterns and trends. They usually contain information for a specific
point in time (i.e., a snapshot of the data), and do not hold daily updates on the
data, and thus sometimes termed as read-only systems [188]. Thus, a DW can also
be defined as a read-only analytical data storage that can be used as the foundation
of a DSS.

8.2.2 Characteristics and Goals

The DW should consider the needs of the organization. From the organization’s
point of view, in terms of contents, the DW should have the following:

- **understandability and ease of access**: since understandability implies legibility,
  the contents of the DW should be easily understandable, intuitive, and obvious
to the users; the contents need to be labeled meaningfully; users may need to
  separate and combine the data in the DW in endless combinations (e.g., by
  slicing and dicing operations, as described later); hence, the tools that access
  the DW should be simple, easy to use, and efficient (e.g., with minimal wait
times to return query results to the user)

- **consistency**: in terms of DW contents, consistency implies quality and completeness; the data in the DW should be credible; data should be carefully assembled from a variety of sources around the organization, cleansed, quality-assured, and released only when it is fit for consumption by the users; information that exist across organizational boundaries (e.g., in multiple departments) should be consistent

- **adaptivity**: the DW should be adaptive and resilient to changes to user needs, business conditions, data, and technology, all of which are subject to inevitable changes with course of time; these changes should also be graceful, i.e., when the DW community asks new questions, or new data is added to the DW, they should not invalidate or disrupt existing data or applications

- **security**: the DW should effectively control access to the organization’s confidential information, and provide a secure environment to its information assets

- **support decision making**: the DW should serve as the foundation for improved decision making by presenting the right data; these decisions deliver the business impact and value attributable to the DW; in other words, the DW should work as a decision support system (DSS)

- **acceptance**: the DW should be well-accepted and embraced by the users

8.2.3 DW vs. OSS

Traditional databases, also known as operational source systems (OSS), operational or transactional databases, or online transaction processing (OLTP) systems, support daily operations of the organization. They handle the day-to-day operations of an organization [188], and strive to ensure concurrent multi-user access, data consistency, and recovery techniques [146]. However, due to their highly normalized nature, they do not always satisfy the complex requirements of advanced data analysis, and often perform poorly when executing complex queries (which involve a series of joins of many relational tables), or aggregating large volumes of data [146, 136, 128].

Users of an OSS almost always deal with one record at a time, and repeatedly perform the same operational tasks [123]. On the other hand, users of a DW almost never deal with one record at a time; rather, they are mostly interested in aggregates
of data/contents that often represent hundreds or thousands of records which are searched and compressed into an answer set [123]. Hence, DW users usually have drastically different needs than OSS users.

Queries against an OSS are narrow, one-record-at-a-time queries that are part of the normal transaction flow, and severely restricted in their demands on the operational system. It is usually assumed that OSSs are not queried in the broad and unexpected ways that DWs typically are queried. The OSSs maintain little historical data; usually, a well-designed DW relieves the OSSs of much of the responsibility for representing the past.

Each OSS often exists as a natural stovepipe application, with little investment made to sharing common data (e.g., date, time, calendar, location, geography, product, customer, etc.) with other OSSs in the organization. To ensure a consistent view for the DW, these OSSs may need to be re-engineered using enterprise application integration (EAI) frameworks [123].

8.3 Basic Concepts and Vocabulary

In this section, we describe some basic DW concepts and vocabulary in the general context of dimensional modeling[1]. Detailed examples of some of the following terms, from the context of our DW of malaria-related data, are given in Chapter [10].

8.3.1 Facts and Dimensions

A fact, or measure, usually represents a numeric performance measurement. It is taken at the intersection of all the associated dimensions. For example, a Daily Sales fact table may contain dollar sales amount and quantity sold as the facts, connected to time, product, and store dimensions.

---

[1] In the text, we adopt the convention of italicizing the facts, dimensions, fact tables, and dimension tables, and capitalizing the first letters of words in fact tables and dimension tables.
In general, the most useful facts in a fact table are numeric and additive (for example, dollar sales amount). As stated before, DW applications almost never retrieve a single fact table row. Instead, they retrieve hundreds, thousands, or even millions of fact rows at a time, and in most cases, the most useful operation with so many rows is to add them up [123]. Thus, additivity of facts is crucial.

Occasionally, facts can also be semi-additive and non-additive. Semi-additive facts can be added only along some of the dimensions and not along others. An example of this is inventory level, where we cannot tell what a level means simply by looking at it. Non-additive facts cannot be added at all. Usually, averages, percentages, and ratios are non-additive. Gross margin, unit price, etc. are examples of non-additive facts.

A dimension is an independent entity in a dimensional model that serves as an entry point or as a mechanism for slicing and dicing the additive measures (facts) located in the fact table. It is a descriptive attribute that describes the “who, what, when, where, why, and how” about the facts. It provides various perspectives that are used to analyze the facts [146].

Dimensions play a vital role in the DW. Since they serve as the source of virtually all interesting constraints and report labels for the DW, they are the key to making the DW usable and understandable. Thus, designing robust dimension attributes is a precondition to achieve robust analytic slicing and dicing capabilities for the DW. The dimensions also guide in implementing the user interface to the DW. As Kimball and Ross [123] describe, the power of the DW is directly proportional to the quality and depth of the dimension attributes, and in many ways, the DW is only as good as the dimension attributes.
8.3.2 Fact Tables and Dimension Tables

A *fact table* is a primary table in the dimensional model where the facts are stored. The facts are characterized by a composite key, each of whose elements is a foreign key drawn from a dimension table. Continuing with the previous example, a *Daily Sales* fact table may contain the facts *quantity sold*, *dollar sales amount*, etc., and its composite key may be comprised of *time* key, *product* key, and *store* key, all of which are foreign keys to dimension tables *Time*, *Product*, and *Store*, respectively.

Fact tables use foreign keys to connect to the dimension tables’ primary keys, thus satisfying the *referential integrity*. For example, the product key in the fact table will always match a specific product key in the *Product* dimension table. The primary key of the fact table itself is generally made up of a subset of the foreign keys, and this key is often called a *composite* or *concatenated* key. Typically, only a subset of the components in the fact table composite key is needed to guarantee row uniqueness. Thus, in dimensional models, fact tables express the *many-to-many* relationships between dimensions.

A *dimension table* is a table in the dimensional model with descriptive (textual) attribute columns. As integral companions to a fact table, dimension tables are the entry points into the fact table: all fact tables are accessed via the dimension tables joined to it. In a well-designed dimensional model, dimension tables may have many columns or attributes (can be as high as 50 to 100), which describe the rows in the dimension table. Dimension tables tend to be relatively shallow in terms of the number of rows (often far fewer than 1 million rows), but are wide with many large columns. In a dimension table, each dimension is defined by its single primary key, which serves as the basis for referential integrity with any given fact table to which it is joined.

The list of dimensions defines the *grain* of the fact table, declaring the scope of the measurement. All the measurements in a fact table must be at the *same* grain.
Declaring the correct grain means to exactly specify the meaning of individual fact table rows. Thus, the grain conveys the level of detail associated with the facts.

Usually, the measurement data resulting from individual business processes are stored in separate data marts. Since measurement data is the largest part of any data mart, a well-designed DW avoids duplicating it in multiple places.

Dimension attributes serve as the primary source of query constraints, groupings, and report labels. Generally, in a query or report request, dimension attributes are identified as the by words. For example, in a DW query to retrieve dollar sales by week by brand, week and brand must be available as dimension attributes.

8.3.3 Dimension Hierarchies

Dimensions can be broadly classified as hierarchical and non-hierarchical. The purpose of a hierarchy is to provide navigational structure for a dimension, so that facts with different levels of aggregation can be obtained by drilling down or rolling up. Dimension hierarchies are the key to navigating dimensions. As explained in Section 8.3.2, dimension data is typically collected at the lowest level of detail (the grain), and then aggregated into higher level totals that are more useful for analysis. Hierarchies within a dimension table allow these natural roll-ups or aggregations to be performed. Within a hierarchy, each level is logically connected to the levels above and below it. Data values at lower levels aggregate into higher levels. For example, the hierarchical dimensions date and location can have hierarchies of year-month-day and continent-country-district-region, respectively. Examples of non-hierarchical dimensions may include active ingredient, chemical class, etc., for a specific product.

Any time a DW user wishes to drill up, down, or into the data, a dimension hierarchy is implicitly referred. Thus, for a large DW/BI system to perform well, the hierarchies must be correctly designed, cleaned, and maintained, so that the drill paths may work properly.
Apart from usability, hierarchies also play an important role in query performance. Using hierarchies, aggregations are often precomputed and stored for intermediate hierarchy levels, and later, transparently used in user queries. These precomputed aggregations are one of the most valuable tools to improve query performance.

For user-oriented dimensions (e.g., time, product, location, etc.), it is natural and common for a dimension to simultaneously support multiple independent hierarchies [123]. These hierarchies may have different numbers of levels. Ideally, the attribute names and values should be unique across these multiple hierarchies. For example, the date dimension may have several possible hierarchies of year-month-day, year-month-week, year-quarter-month, etc. The location dimension may have hierarchies of continent-country-district-region, ZIP code-country-state, etc.

8.3.4 Schemas

DW environments, and DMs, usually transform the relational data model into some special architectures, known as schemas. The most commonly used schemas include the star schema, the snowflake schema, and the constellation schema. In the following, we describe each in detail.

- **Star schema**: A star schema, also known as a star-join schema, is the simplest type of DM schema. It consists of one or more fact tables referencing any number of dimension tables. The star schema gets its name from the logical model’s resemblance to a star with a fact table at its center, and the dimension tables surrounding it representing the star’s points. It is a generic representation of the DM in a relational database in which the fact table’s composite key is joined to a number of dimension tables, each with a single primary key. The star schema is an important special case of the snowflake schema, and is more effective for handling simpler queries.

- **Snowflake schema**: A snowflake schema is represented by centralized fact tables which are connected to multiple dimensions. The term “snowflaking” refers to the process of normalizing the dimension tables in a star schema. In this process, the redundant attributes are removed from the flat, denormalized dimension tables, and placed in normalized secondary dimension tables [123]. Once completely normalized along all the dimension tables, the resultant structure
resembles a snowflake with the fact table in the middle, and the schema corresponds to a full third-normal-form (3NF) entity-relationship (ER) diagram. The principle behind snowflaking is the normalization of the dimension tables by removing low-cardinality attributes and forming separate tables \[189\]. The major advantages of a snowflake schema are space savings and maintainability. However, for DMs, snowflaking generally compromises user understandability and query performance.

- **Constellation schema**: A constellation schema, also known as a galaxy schema, contains multiple fact tables that share many dimension tables. As its name implies, it is shaped like a constellation of stars (i.e., star schemas). Since it contains multiple fact tables, it is usually more complex than star or snowflake schemas, and should only be used for applications that need a high level of sophistication. The fact tables in a constellation schema are shared using conformed dimensions (described in Section \[8.4.3\]) across multiple data marts.

Determining the right choice of schema to use in a DM usually depends on the analysis of project requirements, accessible tools, and project team preferences. As we describe in Chapter \[10\] the constellation schema fits most appropriately for our DM/DW of malaria-related data.

### 8.3.5 Data Marts and Cubes

A data mart is a logical and physical subset of the data warehouse, usually representing data from a single process of interest. A data mart is a wedge of the overall presentation area pie. In its most simplistic form, a data mart presents the data from a single business process. The business processes may cross the boundaries of organizational functions. Deciding the data marts, and the sources of data they would contain, is one of the crucial early steps in the design of the dimensional model. A data mart can contain one or more fact tables. For example, data about manufacturing can be put into a data mart, and be made available to the production department. Human resource data can be put into another data mart, and be provided to the human resources department.

DM views data in an \(n\)-dimensional space, usually called a data cube or a hypercube \[146\]. Intuitively, each data mart can be thought of as a data cube, where the edges
of the cube represent the associated dimensions. Various DW operations (e.g., slicing and dicing) can be performed along each of these dimensions. Points inside the cube are where the measurements for that combination of dimensions are stored. The cube analogy enables to visualize abstract data sets involving many dimensions in a concrete and tangible way, thus promoting understandability of the DW.

8.3.6 Bus Architecture and Bus Matrix

The DW bus architecture provides a rational approach to decomposing the enterprise DW planning task by using an architected, incremental approach. By defining a standard bus interface for the DW environment, separate data marts can be implemented by different groups at different times, and can be plugged together to coexist if they adhere to the DW standard. Separate data mart development teams, while working fairly independently and asynchronously, follow the bus architecture guidelines.

The DW bus matrix is used as a tool to create, document, and communicate the DW bus architecture. It is defined by a tabular matrix layout in which the rows represent separate business processes (or data marts), and the columns represent conformed dimensions used across the DW. If a dimension column is related to a business process row, the corresponding cell in the bus matrix is marked accordingly (see Figure 10.5 as an example bus matrix for our DW of malaria-related data). Usually, the DW bus matrices look surprisingly dense. Creating the bus matrix is one of the most important up-front deliverables of a DW implementation.

The bus matrix, when inspected by rows, instantly reveals the dimensionality of each data mart at a glance. However, the real power of the matrix comes from looking at the columns, since they depict the interaction between the data marts and common dimensions. This allows us, for example, to immediately identify the dimensions that participate in multiple data marts, and thus warranting special attention. It also
allows us to communicate effectively within and across data mart teams, as well as upward and outward throughout the organization. Thus, the bus matrix acts as a powerful device for both planning and communication.

8.4 Data Warehouse Components

A DW typically consists of four separate and distinct components: operational source systems, data staging area, data presentation area, and data access tools [123]. We describe each in the following.

8.4.1 Operational Source Systems (OSSs)

The operational source systems (OSSs) capture the transactions of the organization (see Section 8.2.3). Since in most cases, DW designers have little to no control over the content and format of the data in these operational legacy systems, they are usually considered outside the DW. The main priorities of the OSSs are processing performance and availability.

8.4.2 Data Staging Area

The data staging area of the DW serves as both a storage area and a set of processes commonly referred to as extract-transformation-load (ETL) [123]. The data staging area is everything between the operational source systems and the data presentation area. Kimball and Ross [123] analogizes it to the kitchen of a restaurant, where raw food products (raw operational data) are transformed into a fine meal (the DW deliverables). Similar to the restaurant’s kitchen, the backroom data staging area is accessible only to skilled professionals. The key architectural requirements for the data staging area are to keep it off-limits to DW users, and to isolate it from query and presentation services of the DW. The three-step extract-transformation-load (ETL) process is described below.
Extraction is the first step in the process of getting data into the DW environment. Extracting means reading and understanding the source data, and copying the data needed for the DW into the staging area for further manipulation.

After extraction, numerous potential transformations may be performed, which include data cleansing (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, de-duplicating data, and assigning DW keys, sorting and sequential processing, etc. In many cases, the data staging area is not based on relational technology, but instead may consist of a system of flat files. Though the use of normalized databases (e.g., in a third-normal-form relational format) for data staging storage is acceptable, these normalized databases must be off-limits to the queries of DW users, and excluded from the data presentation area (which should be strictly dimensionally structured).

The final step of the ETL process is the loading of data into the DW presentation area, which usually takes the form of presenting the quality-assured dimensional tables to the bulk loading facilities of each data mart (described later). The target data mart should then index the newly arrived data for query performance. When each data mart is freshly loaded, indexed, supplied with appropriate aggregates, and further quality-assured, the DW user community is notified that the new data has been published.

8.4.3 Data Presentation Area

In the data presentation area, data is organized, stored, and made available for direct querying by DW users, report writers, and other analytical applications. Since the data staging area is off-limits to DW users, from the users’ perspective, the presentation area appears to be the DW; it is all they may interact via data access tools. The presentation area is typically referred to as a series of integrated data
data marts (described in Section 8.3). The data in the data presentation area should be modeled, stored, and accessed in dimensional schemas.

The data marts in the presentation area should contain detailed, atomic data. Atomic data is the most finely-grained data in a dimensional form. It ensures the DW’s response against unpredictable ad hoc user queries. Being the most granular, atomic data is also the most expressive, and has the most dimensionality. Atomic data should be the foundation for every fact table design.

Although the data marts also may contain performance-enhancing summary data (aggregates), it is not sufficient to deliver these summaries without the atomic data. If all the data in the DW is available in the same, consistent dimensional form, most user applications can navigate to the atomic data by using a technique known as aggregate navigation. Existence of atomic data in the presentation area also ensures that the DW users can ask the most precise questions possible. They allow access to the exquisite details so that they can be rolled up to address the users requirements, which are unpredictable, and usually constantly changing.

The data presentation area also requires that all data marts contain conformed dimensions and facts, and thus adhere to the DW bus architecture (described later). Shared, conformed dimensions and facts ensure that all data marts, using the shared information, can be tied together, and do not appear as standalone stovepipe applications. Thus, commitment to the bus architecture is a precondition of building a robust and integrated DW.

Typically, in a large enterprise DW, the data presentation area contains 20 or more similar-looking data marts. Each data mart may contain several fact tables, each of which, in turn, may be connected to 5 to 15 dimension tables. In a well-designed DW, many of these dimension tables will be shared between fact tables.
8.4.4 Data Access Tools

The data access tools refer to the variety of capabilities that can be provided to DW users to leverage the data presentation area for analytic decision making. By definition, all data access tools query the data in the data presentation area.

Data access tools may include simple (e.g., ad hoc query tools) or complex (e.g., sophisticated applications for data mining, forecasting, modeling, etc.) tools. They may also include report writers and analytic applications. Typically, the majority of the DW user-base will access the data via prebuilt parameter-driven analytic applications, which do not require users to direct construction of relational queries. According to Kimball and Ross [123], approximately 80-90% of the potential users will be served by these applications.

8.5 Dimensional Design Process

In this section, we describe the four-step dimensional design process. The design of a DM should consistently follow these four steps in the particular order as described below.

1. Selecting the Business Processes: A process is a natural business activity performed in the organization. Typically, each process is supported by a source data-collection system. Communicating with the DW users and analyzing user requirements are efficient means for selecting the business processes to model. Example business processes include raw materials purchasing, orders, shipments, invoicing, inventory, general ledger, etc. This step is performed by combining an understanding of the business requirements with an understanding of the available data. Usually, the first DM to design should be the one with the most impact, since it would answer the most important business questions, and be readily accessible for data extraction.

Selecting the business processes is not identical to selecting business departments or functions in the organization. For example, a single DM should be designed to handle orders data, rather than building separate DMs for the sales and marketing departments, which both access orders data. Thus, this step should be process-centric, as opposed to department-centric.

2. Declaring the Grain: As we described in Section 8.3.2, all the facts in a fact
table must be at the same grain. Once the business processes are identified, the DW team needs to decide the level of data detail, or granularity, to be made available in the DM. Preferably, the DM should accommodate the most atomic information captured by each business process. A careful grain statement helps to determine the primary dimensionality of each fact table.

3. Selecting the Dimensions: The fact tables should be decorated with a robust set of dimensions that can represent all possible descriptions taking on distinct values in the context of each fact. Once the grain of the fact tables are selected, some of the important dimensions (e.g., date, product, location, etc.) immediately fall out. If the grain of each fact table is properly established, additional relevant dimensions to the basic grain of the fact table can be added. Usually, the additional dimensions naturally take on single values under each combination of the primary dimensions. If, however, any of the additional dimensions violate the grain by causing additional fact rows to be generated, the grain statement should be revised to accommodate the dimension.

4. Identifying the Facts: The final step in the DM design is to determine the facts in each fact table. Since the facts must be true to the grain of the fact table, the grain declaration (declared during Step 2) also helps in this step. However, as potential facts are considered, adjustments may need to be made to either the earlier grain assumptions, or to the choice of dimensions, and facts that belong to a different grain should be moved to separate fact tables.

8.6 Summary

In this chapter, we presented a general background of data warehouses and dimensional modeling. We defined DW and DM, and discussed their connections with business intelligence and decision support systems. We described the characteristics and goals of a DW, and the differences between DWs and operational source systems. We discussed some basic concepts and vocabulary which are commonly used for DW/DM, and explored the major components of a DW. Finally, we discussed the four-step dimensional design process.

In the next chapter (Chapter 9), we describe several data warehouses used in biology in general.
CHAPTER 9

DATA WAREHOUSES IN BIOLOGY

9.1 Overview

In Chapter 8, we presented a general background on data warehouses and dimensional modeling. In this chapter, we describe several data warehouses used in biology in general. We also discuss various heterogeneous data storage systems for malaria-related data, and some data integration challenges in bioinformatics.

The organization of this chapter is as follows: In Section 9.2, we describe several biological databases and DWs. Section 9.3 discusses some challenges that prevail in data integration in biology and bioinformatics. Section 9.4 describes various data storage systems for malaria-related data, and serves as a foundation for our multi-dimensional DW (described in Chapter 10). Section 9.5 concludes.

9.2 Use of Databases and Data Warehouses in Biology

In this section, we describe various databases and data warehouses (DWs) that have been used to collect, organize and manage biological data. In general, biological DW systems provide tight data integration by using common data schemas, and periodically load all data into a central repository. In the recent past, DWs have been used for management of biological data in various domains. Several systems have been developed to integrate and present heterogeneous biological data. For example, the BioMart project adapted DW ideas to create a universal software system for biological data management, and empower
biologists with the ability to create complex and customized datasets through a web interface [112 259 217 89 88]. BioMart uses a *reverse star* schema which is optimized for fast retrieval of large quantities of data [113 259]. It is based on two fundamental concepts: data agnostic modeling and data federation [112]. BioMart organizes data from multiple disparate databases, and provides users an integrated virtual database, allowing data access and cross-reference from these data sources using a single user interface [112]. The BioMart DW framework has been adopted by many different biological databases as their improved data management platform. For example, the *EnsMart* system adopts the framework as a generic DW solution for fast and flexible querying of large biological data sets, and integration with third-party data and tools [113]. The system consists of a query-optimized database and interactive, user-friendly interfaces. EnsMart supports a wide variety of complex queries (on various types of annotations for numerous species) and sequence types.

*Chado* is an integrated relational database schema to manage biological knowledge for a wide variety of organisms, especially the classes of information associated with genome sequences, or the primary RNA and protein products encoded by a genome [165 262]. It can represent many general classes of data in modern biology, such as sequence, sequence comparisons, phenotypes, genotypes, ontologies, publications, and phylogeny. Biological databases that conform to this schema can interoperate with one another, and with the application software from the Generic Model Organism Database (GMOD) open-source software toolkit [70]. GMOD is a collaboration of several model organism database groups, including *FlyBase*, which is the primary database of integrated genetic and genomic data about the Drosophilidae [56].

Chado is driven by *ontologies*, or *controlled vocabularies*, that describe data types and the relationships among them. By changing its ontologies, Chado can be customized to suit different needs. The Chado schema is partitioned into integrated subschemas, or *modules*, each encapsulating a different biological domain [165]. The
modular design allows users to choose to use only selected features that are suitable for their needs [262].

**COLUMBA** is a database of information on protein structures that physically integrates information from twelve protein structure related data sources into a single DW [199, 243]. Shah et al. present a DW software infrastructure for bioinformatics research and development for integrative bioinformatics, named **Atlas**, that locally stores and integrates biological sequences, molecular interactions, homology information, functional annotations of genes, and biological ontologies [210]. The Atlas system is based on relational data models, and managed through Structured Query Language (SQL) in a set of Application Programming Interfaces (APIs) written in C++, Java, and Perl. Töpel et al. describe a Java-based open source toolkit for building life science DWs, named **BioDWH**, which integrates biological information from multiple public life science data sources into local relational database management systems (RDBMS) [242]. Using common RDBMS (e.g., MySQL, Oracle, and PostgreSQL), it provides up-to-date integrated knowledge, platform, and database independence. For the scientific community, BioDWH is available online [25].

Fischer et al. present a DW system named **DWARF**, which integrates data on protein sequences, structures, functional annotations, and a hierarchy of families [67]. Implemented in Firebird (an open-source RDBMS), the DWARF data model consists of three major sections: the protein, the protein sequence, and the protein structure. It is applied to the Lipase Engineering Database [231]. Lee et al. describe an open-source toolkit for constructing bioinformatics DWs, named **BioWarehouse**, that integrates pathway-centric set of databases (e.g. ENZYME, BioCyc, GenBank, Gene Ontology, etc.) [135]. BioWarehouse uses RDBMS (MySQL and Oracle) to enable multi-database queries using SQL. It facilitates a variety of database integration tasks, such as comparative analysis and data mining.

Choi et al. present an integrated DW bioinformatics platform for the soil bac-
terium *Pseudomonas aeruginosa*, named *SYSTOMONAS* (SYSTems biology of pseu-
dOMONAS) [12]. It combines DW concepts with dynamic web services (via Simple Object Access Protocol, or SOAP [213]) to provide dynamically-updated data integration. Kormeier et al. describe a DW, named *CardioVINEdb*, that integrates multiple heterogeneous data sources of cardiovascular diseases [128]. It is based on an earlier DW (named *VINEdb*) designed for integration and interactive exploration of life science data [90].

9.3 Data Integration Challenges in Biology & Bioinformatics

In this section, we discuss some challenges that prevail in data integration in biology and bioinformatics. The importance of data integration in these fields has been recognized for many years. In order to meet the objectives of data analysis and data management, it is essential for scientists to access and analyze information from multiple data sources [242]. The primary challenges of data integration include combining diverse and multiple data and bringing them into a homogenous, consistent state [242], as well as capturing, modeling, and analyzing the data in a consistent way to provide new insights into complex biological systems [128].

In general, biological data management is challenging for database designers, collaborating research groups, and other users, due to factors that include complex biological concepts (which are not always well-defined), constantly changing data models, frequently emerging new techniques and new information, need for unified access to distributed data sources from different geographical locations, etc. [112]. Usually, biological data require specialized expertise for their collection, analysis, and interpretation [259]. In most cases, different biological data sources, distributed around the world, rely on their own storage and presentation solutions. Thus, they often require duplicated efforts by researchers and users in learning how to use multiple interfaces to access the data. Hence, a standardized solution is desired for
both data producers and consumers \cite{259}. The increasing volume and complexity of the data, sometimes accompanied with exponential growth, also compound the challenges, forcing the user to progress through a long learning curve \cite{217, 89}.

The integration of life science data from heterogeneous, autonomous, and distributed data sources itself emerged as an important research field, with special challenges regarding the large heterogeneity of the databases on the semantic and technical level \cite{242}. In recent years, the relatively new field of integrative bioinformatics has emerged to cope with this problem. By providing new and novel methods to integrate, manage, and analyze these data, it allows to gain new insights and deeper understanding of complex biological systems \cite{67}. Teodoro et al. list some other challenges of the integration process \cite{227}:

- lack of technical interoperability, including different hardware platforms, operating systems, database management systems, access protocols, transport formats, and programming languages
- lack of semantic interoperability within different data sources
- management of heterogeneous data quality, especially when it comes to statistical analysis, sensitivity to heteroscedasticity (of several distributions having different variances)
- security, privacy, and confidentiality across regions and countries

Some of the systems used for management of biological data, as we discussed in Section \ref{sec:9.2} also exhibit certain other problems. For example, Atlas \cite{210}, BioWarehouse \cite{135}, COLUMBA \cite{243}, etc. are not platform-independent, and are implemented with different programming languages \cite{128}. Due to the lack of logging information, they also pose difficulties in determining whether the data represent the most recent version \cite{128}. A well-designed data warehouse may offer several advantages by eliminating some of the challenges as discussed above.
9.4 Heterogeneous Data Storage Systems for Malaria

Malaria-related data exist in numerous storage systems\(^1\) assisting malaria researchers, policy-makers, and others across the globe. With the advent of information technology (IT) and ease of data storage, sharing, and distribution over the Internet, the number of storage systems (including databases) has proliferated in recent years. Often, these systems are heterogeneous in nature, and the heterogeneity encompasses all aspects of the stored data (e.g., data content, type, storage format, etc.).

As we explained in Section 9.3, the problem of integrating heterogeneous biological databases arises in various scenarios in which information from autonomous (i.e., independently developed and operated) data sources need to be combined under a common platform for storage, access, and analysis. Primary reasons of the problem involve the differences in data definition and content \([32, 146, 128]\).

For malaria, due to the complex nature of the disease, and its numerous aspects on which data are stored, there is a growing need to integrate these heterogeneous data sources. Storage of data in flat files (e.g., in spreadsheets or CSV files), or even traditional relational databases (RDBMSs), cannot always cope with the growing demand of this integration process, due to several reasons. Traditional databases, also known as operational or transactional databases, or online transaction processing (OLTP) systems, support daily operations of the organization. They primarily concern with ensuring concurrent multi-user access, data consistency, and recovery techniques \([146]\), and can handle the day-to-day operations of an organization \([188]\). However, due to their highly normalized nature, they cannot always satisfy the complex requirements of advanced data analysis, and often perform poorly when executing complex queries (which involve a series of joins of many relational tables), or aggregating large volumes of data \([146, 136, 128]\).

\(^1\)We use the more generic term storage system, as opposed to database, since the former encompasses a wider variety of systems.
The need to ensure long-term usability and adaptivity of malaria-related data also poses other challenges. Usability centers around three issues:

1. Data consistency: ensuring the validity, accuracy, and integrity of stored data between applications.
2. Data cleansing: guarding against illegitimate and/or erroneous values due to errors in data input or publication, coded values contained by many data sources, etc.
3. Quality assurance: double-checking the stored information

Adaptivity of data ensures that it is flexible to suit new data requirements or user conditions. The advent of IT and the widespread use of the Web have made it easier than ever before to publish data on the Web, making it instantly accessible across the globe. Thus, it is inevitable that new data sources are constantly being published. For any comprehensive digital framework dealing with malaria data (e.g., the VECNet CI [241]), it is thus very important to be able to incorporate the new data sources and content, while still maintaining the consistency and quality of the existing content. In some cases, even the content from the same data sources may change over time, making it difficult for integrated systems to cope with the changes.

Different databases focus on different aspects of malaria data. To name a few, the World Malaria Report (WMR) data from the World Health Organization (WHO, [251]) stores data on recommended policies and strategies for malaria control, antimalarial drug policy, intervention coverages estimated from routinely-collected surveys, household surveys, etc. The Malaria Atlas Project [232] produces/stores geographical maps on dominant malaria vectors, their spatial distribution, estimated human populations at risk, etc., organized according to different geographical regions. Various types of weather-related malaria data are stored, for example, in the Pacific Rainfall (PACRAIN) database [236], the Global Surface Summary of Day (GSOD) database [78], etc. Some of these storage systems store data in relational databases, yet others store them as spreadsheets, comma-separated values (CSV) files, plain text files, or in other file formats. Quick and clear understanding of the intuitive meaning
of the stored data is not always convenient, or even plausible, because often the user needs to decipher the definition of the data content (e.g., data stored as columns in spreadsheets) from other supporting and/or accompanying information that are provided with the original data. For example, both GSOD \cite{gsod} and PACRAIN \cite{pacificrain} provide additional information in separate files that describe the meaning of content, column headers, codes used in the content, etc. Also, due to the changing nature of some data, often the publishers may change the data definitions or columns, augment new types and entries, or reorder the existing ones. All of these enforce the users to learn the changes before they can understand and use the data.

As we show in Chapter \ref{chapter:multidimensional-dw}, a well-designed DW can cope with most of the challenges described above, thus ensuring the long-term usability and adaptivity of malaria-related data.

9.5 Summary

In this chapter, we described biological data warehouses, data integration challenges in biology and bioinformatics, and several heterogeneous data storage systems for malaria-related data. In the next chapter (Chapter \ref{chapter:multidimensional-dw}), we present a multi-dimensional DW for these types of data, and highlight its advantages as it eliminates some of the challenges discussed above.
10.1 Overview

In Chapters 8 and 9, we presented a general background on data warehouses and dimensional modeling, and discussed several biological data warehouses and data storage systems for malaria-related data. In this chapter, we describe a multi-dimensional, online analytical processing (OLAP) data warehouse named MalOLAP, designed specifically to store, access, and analyze malaria-related data scattered in heterogeneous forms and contents across many data storage systems.

In general, numerous examples of the use of DWs in biology and bioinformatics can be found in the literature, as we described in Section 9.2. For malaria research, numerous data storage systems also developed over the past few decades for malaria-related data, as we described in Section 9.4. However, to the best of our knowledge, no dimensional model or data warehouse has been designed for the specific task of providing OLAP functionalities for malaria-related data. Hence, we envisage that MalOLAP would fill the vacuum in this research area by providing a robust DW/DM environment for a variety of user groups.

MalOLAP, being a multi-dimensional OLAP data warehouse, provides a robust way to store, access, and analyze malaria-related data scattered in heterogeneous forms and contents across many data storage systems, thus solving the problem of integrating heterogeneous biological data from independently developed and operated

\footnote{\textsuperscript{1}Major portions of this chapter are in preparation in Arifin et al. \cite{12}.}
data sources, as we discussed in Section 9.3. Being a pilot project, to be integrated into the VECNet CI analytical framework, MalOLAP is designed to fit and operate in harmony with the other major components of VECNet (e.g., the digital library, the integrated modeling platform, and the front- and back-end services), providing a platform for efficient search, retrieval, analysis, and visualization of historical as well as predictive (synthetic) malaria-related data. Being a data warehouse, MalOLAP differs from traditional operational databases, and its dimensional model intuitively views the multi-dimensional data as data cubes. It possesses most of the prime characteristics of a data warehouse (being subject-oriented, integrated, non-volatile, and time-varying, as we describe in Section 10.2), and supports common OLAP operations (slice and dice, roll up and roll down, etc.).

The organization of this chapter is as follows: In Section 10.2, we define MalOLAP as a DW, and discuss its context within the VECNet framework, the two broad categories of data that it supports, and its intended user groups. Section 10.3 defines a vocabulary of relevant terms. Section 10.4 describes the external data storage systems (also known as the operational source systems). In Section 10.5, we describe the logical and physical design issues of the DW/DM. Section 10.6 describes the cubes and Django frameworks. In Section 10.7, we present some results as screenshots from the MalOLAP web portal. Section 10.8 describes some miscellaneous issues. Finally, Section 10.9 points to future directions, and Section 10.10 concludes. Some additional materials for this chapter are presented in Appendix E.

10.2 MalOLAP: A Data Warehouse for Malaria

In this section, we define the four prime characteristics (as mentioned in Section 8.2) of MalOLAP as a DW. We also discuss the broader context of MalOLAP within the VECNet CI, the two major categories of data (historical and predictive) that MalOLAP is designed to support, and its intended user groups.
From the perspective of malaria-related data, the four prime characteristics of a DW, as they are implemented in MalOLAP, may be interpreted as:

**subject-oriented:** MalOLAP targets multiple subjects of analysis according to the requirements of malaria decision-making managers/users at various levels; these subjects may include, for example, data on existing and new malaria control interventions, household surveys, antimalarial drugs, entomological inoculation rate (EIR), dominant vector species (DVS), various models and studies (e.g., mathematical, agent-based, field-based), model-specific parameters, etc.

**integrated:** the contents of MalOLAP result from integration of data from various external data storage systems, also known as operational source systems (OSSs); some of these OSSs include WMR, PACRAIN, GSOD, etc. (see Section 10.4)

**nonvolatile:** once ingested, MalOLAP data are meant to be read-only and persistent; we perform modification/curation of content from the OSSs only in the data staging area (described later); thus, MalOLAP stores curated and nonvolatile data, retained for long-term analysis and aggregation purposes.

**time-varying:** MalOLAP records the temporal evolution of data from the OSSs for a long period of time (typically many years); for example, it may keep track of temporal malaria data on endemicity, incidence, prevalence, etc. ranging over several decades

10.2.1 MalOLAP within VECNet

MalOLAP is designed to fit and operate within an analytical framework named as the Vector Ecology and Control Network (VECNet) [241]. VECNet is a consortium of institutions established to analyze malaria transmission and its reduction by one or several vector control interventions. It was originally formed as a collaborative effort to use a shared set of resources in order to facilitate interactions and communication between malaria modelers and other scientists to support research activities, surveillance, monitoring, and evaluation of malaria programs and interventions.

VECNet’s proposed cyberinfrastructure (CI) is an online collaboration, storage, and compute engine that enables its users to conduct research, and to analyze and share results from multiple models with the stakeholders and the broader scientific community. The CI primarily consists of a data warehouse (MalOLAP), an interac-
tive digital library, and an integrated modeling platform. The digital library includes software and information management tools required to store and access digital data and metadata, providing a central repository of data to inform users, enabling better analysis and optimal decision-making. It will contain primary data, drawn from public (i.e., published) as well as non-public sources, including lookup tables, summaries of actual measurements of malaria burden, transmission, vector distributions, interventions, etc. The integrated modeling platform includes middleware tools, interfaces, and pre/post-processing steps required to abstract model execution, and hosts multiple predictive models (e.g., mathematical, agent-based, etc.), to fill the digital library with comparable scenarios, guiding progress towards eradication of malaria.

The data in the digital library are imported from primary owners, transformed into a standard format, and stored in auxiliary databases as well as in the data warehouse (MalOLAP). These primary components are supported by model execution environments and a set of common tools (required by both developers to build and test frameworks, project management, etc., and by end-users to support visualization, documentation, etc.). The major components of the VECNet CI, including its front and back end services, and MalOLAP’s placement within the CI’s broader context, is depicted in Figure [10.1].

10.2.2 Historical and Predictive Data

MalOLAP is designed to store two broad categories of data: historical and predictive. The interrelations between these two categories are crucial because in general, often the predictions and interpretations of biological data are made by comparing predictive data against existing historical data [16]. In this chapter, we describe the dimensional modeling for historical data; modeling for predictive data will be described in future.

Historical data, ranging over several years, need to be stored so that the expert
Figure 10.1. MalOLAP within the Broader Context of VECNet CI. The major components of the CI include the Digital Library, the Integrated Modeling Platform, MalOLAP (data warehouse), Front End Services, and Back End Services. The three inner components are shaded in sky blue. The Digital Library provides a means to create, curate, and store historical and predictive data. The Integrated Modeling Platform consists of a system of models with definitions and processes for input and output for each model. The data warehouse, MalOLAP, houses the dimensional model, and stores the data marts, facts, dimensions, lookup tables, and other relevant objects, allowing common OLAP operations (e.g., slice and dice, aggregations, etc.) on the stored data. The arrowheads in the data paths indicate either bi-directional or uni-directional flow of data. As indicated by the dashed lines, MalOLAP may provide historical or predictive data as input to the Integrated Modeling Platform, and retrieve output data from the models to store in itself. Similarly, the Digital Library may provide historical data to MalOLAP in order to store it in the dimensional model, or retrieve (and store) predictive data from MalOLAP.
users can analyze trends of various characteristics of malaria and/or assess the impact of vector control interventions by considering various characteristics (e.g., demographic, entomological, epidemiological, etc.) associated with the data. For example, the users may be interested to understand the activities that led to an intervention being stopped to be applied in a specific location. On the other hand, *predictive* or *synthetic data*, mostly generated as output of agent-based micro-simulations of malaria transmission and other types of malaria models, is useful because they can simulate specific situations with precise settings, and help understand past trends and behaviors, leading to discovery of new insights about the current environment. Thus, equipping the DW with *predictive* data allows it to provide users an analytic foundation with its ability to pose *what-if* questions to assess the outcome of various strategies, rather than being limited to qualitative decision models [188]. Since multiple individual models will be run (possibly with distinct but overlapping set of input parameters), and produce intermediate and final outputs in various output formats, model-specific input/output must be considered in designing the DM/DW to store the predictive data. In addition, *missing data*, where applicable, may also be incorporated into the DW.

To provide a semantic framework for the DW, we present a simple ontology for MalOLAP in Figure 10.2. It depicts the high-level description of concepts and the relationships between them, organized in the upper parts of the MalOLAP knowledge domain, using a representative vocabulary (described in Section 10.3). As shown in Figure 10.2, the data warehouse, MalOLAP, exists as the single entity in the root level (level 1), and has multiple data marts and lookup tables as level 2 entities. Each data mart can be modeled as one or more data cubes (level 3). Each data cube usually has one or more fact tables (level 4), and multiple dimension tables (level 4), all of which store records (facts and dimensions, level 5) of varying granularities. Each lookup table can be either a relational table or a dictionary (level 3). A relational
table may contain records which are mostly dimension-less. A dictionary may in turn be a relational table, or may contain semantic definitions (level 4), which are stored as records.

10.2.3 User Groups

In an orchestrated effort with other components of the VECNet CI, MalOLAP will primarily serve the following representative (often overlapping) user groups:

- **researchers**, who want to query the digital library or run model simulations to explore vector behavior parameters and their relationships to vectorial capacity to expose vulnerabilities to interventions at a detailed level

- **modelers**, who may be working to support other categories of users; the modelers are either: 1) the developers of models that are or will be hosted in the VECNet CI, 2) users of the models already hosted by VECNet CI, or 3) analysts primarily contributing to the data contained in the VECNet digital library

- **malaria control managers**, from ministries of health, who will interact with the VECNet CI to optimize the impact of interventions and combinations of interventions on the numbers of malaria cases for geo-spatially defined areas

- **product developers**, of novel malaria interventions for both vector control and improved public health, who will interact with the VECNet CI to gain insights to design and evaluate Target Product Profiles to enhance the design of novel interventions and to understand where interventions will be most effective

- **funding agencies** and **policy makers**, who will interact with the VECNet CI to assist decision-making on investment strategies for interventions

10.3 Vocabulary

In this section, we describe the DW/DM vocabulary of relevant terms used in the context of malaria-related data for MalOLAP, with examples as necessary. Some of the terms were defined in Chapter 8 and re-appear here from the context of our DW. For inter-relationships between the terms, see Figures 10.2, 10.3, and 10.4.

2As in Chapter 8, we adopt the convention of italicizing the facts, dimensions, fact tables, dimension tables, and lookup tables, and capitalizing the first letters of each word in fact tables, dimension tables, lookup tables, and dictionaries.
Figure 10.2. MalOLAP Ontology. Entities and their relationships are represented by rectangles and labeled arrows, respectively. Entities within the same ontology level are marked with the same colors. The root level (level 1) has a single entity as the data warehouse, MalOLAP, which has multiple data marts and lookup tables as level 2 entities. Each data mart can be modeled as one or more data cubes (level 3). Each data cube usually has one or more fact tables (level 4), and multiple dimension tables (level 4), all of which store records (facts and dimensions, level 5) of varying granularities. Each lookup table can be either a relational table or a dictionary (level 3). A relational table may contain records which are mostly dimension-less. A dictionary may in turn be a relational table, or may contain semantic definitions (level 4), which are stored as records.
**Record** a generic term used to describe a row in the fact tables, dimension tables, and lookup tables.

**Fact** a numeric, quantitative measure stored as part of a row in a fact table; as explained in Chapter, the most useful facts are usually numeric and additive. Examples: total population who slept under an ITN, proportion of households with at least one bednet, etc. in the fact table *Household Surveys*.

**Dimension** a descriptive attribute that provides context to the facts, and describes the “who, what, when, where, why, and how” about the facts; dimension attributes serve as the primary source of query constraints, groupings, and report labels; dimensions may be hierarchical and non-hierarchical. Examples: 1) hierarchical: dimension *date* with hierarchy *year-month-day*, etc; 2) non-hierarchical: *land use*, *chemical class*, etc.

**Fact table** a primary table in the DM to store the facts; each fact is characterized by a composite key, composed of foreign keys drawn from the dimension tables. Examples: *Household Surveys*, *Operational Coverage of ITNs*, etc.

**Dimension table** a table in the DM with a single-part primary key and descriptive (textual) attribute columns; as integral companions to a fact table, dimension tables serve as the entry points into the fact table; each dimension table may have many columns or attributes, which describe its rows. Examples: *Location*, *Date*, etc.

**Data mart** a logical and physical subset of the DW, usually representing data from a single process of interest; each data mart may contain one or more fact tables. Examples: *Household Surveys*, *Operational Coverage* (of ITNs, IRS, and ACT), etc.

**Lookup table** in addition to the fact tables, lookup tables, also known as *reference tables*, contain unordered collection of values (mostly textual) that are stored in relational tables (in the RDBMS); each lookup table is an auxiliary table to hold static data, and is used to lookup values, which may be related to individual dictionaries; sometimes, it can also be used to translate an encoded/obscure epidemiology term into an explanatory description for reporting purposes; lookup tables save space, improve flexibility, and allow the description of a coded value to change while retaining its meaning. Examples: *Species*, *Entomological Endpoints*, etc.

**Dictionary** contains definitions of terms which need explanatory descriptions; the terms usually appear in lookup tables, which is related to the dictionary. Examples: *Species Bionomics*, *Entomological Endpoints Phases*, etc.

**Definition** the explanatory description of a term which is stored in a dictionary.

**Constellation schema** also known as the *galaxy* schema, the constellation schema is a collection of the star schemas modeled in MalOLAP; it ties together all the
fact tables, dimension tables, and lookup tables; it allows conformed dimensions and facts to be shared, thereby facilitating the realization of the DW enterprise bus architecture; however, the fact tables in the schema do not need to be directly related. Examples: see Figures 10.3 and 10.4.

**Conformed dimension** a common structured dimension that is shared across multiple fact tables; they are used to avoid redundant data in the DW. Examples: *Date* and *Location* in Figure 10.3.

**Bus matrix** a tabular tool used to create, document, and communicate all data marts at a glance; as shown in Figure 10.5, the rows of the bus matrix represent the data marts, and the columns represent the dimensions used across the DW; if a dimension (column) is related to a data mart (row), the corresponding cell in the bus matrix is marked accordingly.

10.4 External Data Storage Systems

In this section, we describe the external data storage systems from which the contents are ingested into MalOLAP. These systems are also known as the operational source systems (OSSs). For each OSS, we mention the corresponding fact tables, dimension tables, lookup tables, and relevant examples. Since each OSS in the DW may serve as the source for multiple fact tables, the terms *OSS* and *data mart* are used interchangeably. The constellation schemas, shown in Figures 10.3, 10.4, include high-level views of all OSSs. Detailed schemas for the fact tables, dimension tables, and lookup tables are presented in Appendix E. Note that although we could combine multiple data marts (and the corresponding ingested data originating from multiple OSSs), most of the data marts are segregated in anticipation of large volumes of data to be integrated in the future.

10.4.1 Household Surveys

This data mart describes the household surveys of mosquito nets ownership and usage data, as part of the World Malaria Report (WMR) 2010 [252]. The data is published annually by the WHO [251], and summarizes information received from
Figure 10.3. MalOLAP Constellation Schema for Data Marts. The constellation schema, also known as the galaxy schema, ties together all the fact tables and dimension tables contained within all data marts. The connection links join the fact tables to corresponding dimension tables. The fact tables are (partially) shaded in sky blue. Some dimensions are shaded in dim red. To aid in visualization, dimensions which are connected to comparatively large number of fact tables are shaded in different colors, along with their connection links. Detailed schemas for individual data marts are presented in Appendix E.
Figure 10.4. MalOLAP Constellation Schema for Lookup Tables. The constellation schema depicts all lookup tables, joined by connection links to corresponding dictionaries and dimension tables. The lookup tables are (partially) shaded in lavender. The lookup dictionaries are (partially) shaded in light brown. Detailed schemas for individual lookup tables are presented in Appendix E.
Figure 10.5. MalOLAP Enterprise Bus Matrix. The enterprise bus matrix for the data warehouse. An ‘X’ in a cell represents the correspondence of the dimension (column) with the data mart (row). Separate data marts can be implemented by different groups at different times, and can be plugged together and usefully coexist. The bus matrix is independent of the technology and the database platform. The coherent data marts share a uniform architecture of conformed dimensions and facts, which allow them to be fused into an integrated data warehouse.
106 malaria-endemic countries and other sources. It highlights continued progress made towards meeting the World Health Assembly (WHA) [250] targets for malaria to be achieved by 2015 [252]. The data mart is associated with dimensions location, date, source, and subgroup. Facts include percentage of households with at least one bednet (any net, ever-treated net, or ITN), percentage of population sleeping under bednets (total, less than five years old, and pregnant women), etc.

10.4.2 Vector Control Interventions

Also published as part of the WMR 2010 [252] data, this data mart contains operational coverage data for various vector control interventions (currently ITNs, IRS, and ACT), and is stored in three fact tables (one for each intervention). The data outlines the evolving financing situation for malaria control, the impact of these growing resources (primarily the increased coverages of WHO-recommended interventions), and the association between this rapid scale-up and substantial reductions in malaria burden [252]. The data mart is associated with dimensions location and date. Facts include percentage of coverages, number of bednets (ITNs and LLINs) sold or delivered, number of people protected by bednets and antimalarial treatment courses, number of treatment courses delivered (ACT), etc.

10.4.3 Rainfall Data from PACRAIN

This data mart stores an extensive collection of daily and monthly rainfall records, collected by schools across 936 sites, from various Pacific islands, atolls, and the United States, by the Schools of the Pacific Rainfall Climate Experiment (SPaRCE) [238], and stored in a comprehensive database named as the Pacific Rainfall database (PACRAIN) [236]. The PACRAIN project is sponsored by the National Oceanic and Atmospheric Administration (NOAA) [169]. Students at the schools (from elementary, middle school, high school, college, and trade school) record daily rainfall mea-
surements; the recorded data is then returned to the University of Oklahoma \[240\], and entered as a part of the PACRAIN database. Other data sources for the sites include French Polynesian Meteorological Service \[68\], National Institute for Water and Atmospheric Research, New Zealand \[168\], Taylor’s atlas of Pacific islands rainfall \[226\], and the National Climatic Data Center in U.S. \[233\]. The data mart is associated with dimensions **pacrain site, pacrain site data source, georeference, beginning of observation date, and last modification date.** The last two dimensions are aliases for the dimension **date**. Facts include **total rainfall (in millimeters)** and **length of observation (in hours)**. Other relevant information associated with the sites include latitude, longitude, elevation, terrain type, etc.

### 10.4.4 Georeferenced EIR Data from MARA

This data mart describes georeferenced annual *Plasmodium falciparum* entomological inoculation rates (EIRs) data from the Mapping Malaria Risk in Africa (MARA) collaboration \[150\]. The data was collected by a review study of an extensive search of the formal and informal literature on annual EIR across Africa from 1980 onwards \[94\]. The study, conducted primarily to investigate the spatial heterogeneity of malaria exposure in Africa, supports the idea of highly heterogeneous risk at the continental, regional, and country levels, and its implications for the rationalization of malaria control \[94\]. The values recorded include *P. falciparum*-infected bites per adult, per night indoors, using human biting rates averaged over a year. The data mart is associated with dimensions **location, land use, time interval, georeference, georeference source, seasonality meaning, sporozoite index method, biting rate method, and citation.** The **time interval** dimension is modeled using the **date** dimension, as shown in Figure \[E.7\]. The dimension **georeference source** indicates the source of information used by the sites to georeference each annual EIR (original references or citations, correspondence with authors, published maps and/or the GeoName dig-
ital gazetteer CD-ROM [72]). Seasonality meaning indicates either the number of months in which 75% of transmission occurred, or the length of the entire transmission season for each location. Land use indicates the land type of the study site (rural, irrigated rice, urban, or multiple). Biting rate method and sporozoite index method indicate the means used in obtaining the biting rate (exit traps, human bait, insecticide spray, or light traps), and the means of measuring the sporozoite index (dissection or ELISA), respectively. Citation indicates the corresponding literature. Facts include annual EIR, average annual biting rate, sporozoite index, percentage of the total annual EIR transmitted by An. gambiae s.l. freshwater species, An. funestus, and all other species, etc.

10.4.5 Weather Data from GSOD

This data mart stores a collection of daily data from over 9000 weather stations located around the world [78]. It describes the Global Surface Summary of Day (GSOD) data built monthly by the National Climatic Data Center (NCDC) [233]. The hourly data used in building these daily summaries are obtained from the Air Weather Service (AWS) Global Climatology Division. Over 8000 stations’ data are typically included for each month. Historical data are generally available for 1929 to the present, with data from 1973 to the present being the most complete. The dataset includes 18 daily surface meteorological elements, derived from each station’s synoptic (general summary) hourly observations. In deriving the summary of day data, a minimum of 4 observations for the day must be present (thus allowing for stations which report 4 synoptic observations per day). The data mart is associated with dimensions GSOD stations and date. Facts include mean values for temperature, dew point, sea level pressure, station pressure, visibility, wind speed, maximum values for sustained wind speed, wind gust, temperature, minimum temperature, precipitation amount, snow depth, etc. Other relevant information include indicators for
occurrences of fog, rain, snow, hail, thunder, tornado or funnel cloud, etc.

10.4.6 Insecticides

This data mart describes the common insecticides used in vector control, including products used for IRS, larvae control, LLINs, space sprays, etc. The data source is collected and created by R. Farlow Consulting, LLC for VECNet. The data mart is associated with dimensions location, intervention, active ingredient, chemical class, and manufacturer. Facts include the products’ trade name (common name).

10.4.7 Lookup Tables

In this section, we describe the two lookup tables of MalOLAP.

**Species** this lookup table specifies the species bionomics parameter values for use in VECNet applications that require bionomic input; it is associated with dimension location, and dictionary Species Parameters. All facts of the lookup table and entries of the dictionary are shown in Figure E.13. The lookup table appears in the constellation schema in Figure 10.4.

**Entomological Endpoints** this lookup table outlines the major measurable entomological endpoints impacted by vector control paradigms; it is associated with dimension intervention, and dictionary Entomological Endpoints Phases. All facts of the lookup table and entries of the dictionary are shown in Figure E.14. The lookup table appears in the constellation schema in Figure 10.4.

10.5 Design and Implementation

In this section, we describe the logical and physical design of the DW and the dimensional model (DM) used for MalOLAP. In particular, we discuss the design methodology, the data cubes, the OLAP operations, and the four DW components.

10.5.1 Design Methodology

To facilitate a rapid development, we use a bottom-up approach (as opposed to a time-consuming top-down approach which requires a global enterprise model)
and a centralized architecture (as opposed to federated or tiered architectures) which stores all data from various data marts into a single DW. Initially, we follow the source-driven (as opposed to analysis-driven) approach for the logical/conceptual design. This approach allows us to analyze and collect information from the heterogeneous sources to the DW. However, as the DW matures, we envision that the analysis-driven approach, which centers around the analysis requirements of the users, would be combined with the source-driven approach into an iterative development methodology. This, in turn, would ensure the correspondence of the analysis requirements with the available information in the DW.

Because of its ability to yield rapid turnarounds on deliverables, the iterative development methodology is advocated as a key to success for a DW implementation. In this method, the standard steps of the systems development life cycle (SDLC), which includes preliminary analysis, systems analysis and requirements definition, systems design, development, integration and testing, acceptance, installation, deployment, and maintenance, are followed. As we progressively advance the DW by incorporating new sources of data, many of these steps are undertaken multiple times in a series of iterative refinements. The goal is to allow the specifications and implementation to evolve gradually, as our end-users and other domain experts can continue to develop their requirements by interacting through MalOLAP, and can provide valuable feedback in the development process. Adjustments and enhancements of features to the data sets in previous iterations are made based on the collected feedback. Since the success of any DW is ultimately measured by the users, we consider the direct involvement of our end-users in the early phases of the SDLC as a critical success factor.
10.5.2 Data Cubes and OLAP Operations

The DM intuitively views the data stored in the DW as cubes of data. Each data mart can be modeled with one or more data cubes. Data points inside a cube represent the stored measurements (facts) for specific combinations of the associated dimensions. The multidimensional cube view of data promotes understandability by allowing data visualization in a concrete and tangible way [123]. Results of a multidimensional query is either a cell, a two-dimensional slice, or a multidimensional sub-cube [136]. Common OLAP operations on the multidimensional data stored in MalOLAP include:

- **Roll up and roll down**: these operations allow the user to navigate among levels of data ranging from the most summarized (up) level to the most detailed (down) level, along any hierarchical dimension. *Roll up*, also known as *aggregation*, or *drill up*, permits higher levels of summarization. *Roll down*, also known as *drill down*, is the opposite operation which permits to view more fine-grained levels of data. The design and granularity of the DW determines its ability to roll up or down. The summarization rule (for rolling up) may involve computing the sum, average, maximum, or minimum values along the hierarchical dimension, or applying a set of formulas, for example:

  \[
  \text{number of bednets} = \text{number of ITNs} + \text{number of LLINs}
  \]

  Both of these operations are depicted in Figure 10.6 based on some example (hypothetical) data.

- **Slice and dice**: these operations allow the user to access a DW through any of its dimensions; the process generally implies a systematic reduction of warehouse data in any logical combination. The *slice* operation selects a rectangular subset of a data cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension. The *dice* operation produces a sub-cube by selecting specific values of multiple dimensions. Both of these operations are depicted in Figure 10.7 based on some example (hypothetical) data.

In Section 8.4, we described the four separate and distinct components of a DW. Figure 10.8 depicts these four components as they are implemented in MalOLAP. In the following, we describe these components in detail.
Figure 10.6. Roll Up and Roll Down Operations. These operations allow the user to navigate among levels of data ranging from the most summarized (up) level to the most detailed (down) level, along a specific dimension. (A) Hypothetical data showing mosquito abundances for various locations (continents and countries), years, and Anopheles species.

(B) The data cube, derived from the data in A, shows the mosquito abundance facts (numbers in rounded rectangles). The cube is associated with three dimensions: species, year, and continent, which are displayed along the three axes, with data labels coming from A. Fact cells with different values of the continent dimension are distinctly color-coded for ease of visualization. (C) The roll down operation produces a more detailed view of data by rolling down one level along the hierarchical location (from continent to country). The mosquito abundances of continent Africa are rolled down to abundances for countries of Africa (Angola, Benin, Kenya, and Nigeria in this hypothetical scenario). Note that when rolled up, the entire cube in C represents one row of data in B (in this case, the topmost row, representing Africa).
Figure 10.7. Slice and Dice Operations. These operations permit to access a DW through any of its dimensions. (A) Hypothetical data showing mosquito abundances for various dates, times, and sites (mosquito collection sites). In figures B–D, fact cells with different values of the time dimension are distinctly color-coded for ease of visualization. (B) The data cube, derived from the data in A. (C) The slice operation selects a rectangular subset of the cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension. The mosquito abundances of all sites, for all dates, at 7PM are sliced out of the data cube. (D) The dice operation produces a sub-cube by selecting specific values of multiple dimensions. The abundances of sites Ifakara and Garki, for all dates, at 8PM and 10PM are diced out of the data cube.
Figure 10.8. MalOLAP Components. The four separate and distinct components of the data warehouse environment: operational source systems, data staging area, data presentation area, and data access tools. Each component serves specific functions, as described in Section 10.5. The modeling phases and/or implementation technologies used are listed at the bottom in the blue-shaded boxes for all components.
10.5.3 Operational Source Systems (OSSs)

The operational source systems (OSSs) refer to the external data storage systems (mostly flat files) from which the contents are ingested into MalOLAP. As described in Section 10.4, currently we include these OSSs: Household Surveys, Vector Control Interventions, Rainfall Data from PACRAIN, Georeferenced EIR Data from MARA, Weather Data from GSOD, Insecticides, Species, and Entomological Endpoints. After collecting the content of each OSS, we identify the dimensions and facts, and then decide on existing or new data marts to ingest the content. In this logical modeling phase, we pay particular attention to identify the correspondence between the data marts and the conformed dimensions, and populate the DW bus matrix (see Section 8.3.6). At the end of this phase, as each data mart has been logically modeled, the constellation schemas tie together all the fact tables, dimension tables, and lookup tables contained within all data marts (as depicted in Figures 10.3 and 10.4), and the enterprise bus matrix contains the coherent data marts sharing a uniform architecture of conformed dimensions and facts (as depicted in Figure 10.5).

As we perform the domain analysis for individual data marts, we analyze the contents and data type of each attribute in the OSSs’ source files. To ensure that each record can be uniquely identified within the data source (i.e., the referential integrity), we determine and assign primary keys for all dimensions and facts. If no such keys can be found, we consider additional candidate keys to generate the primary key, and try to avoid any system-generated control field.

In the subject area analysis phase, we identify overlapping data fields from the source files. This step is important because it directly translates to the identification of conformed dimensions and facts in our DW. Careful identification of synonyms (elements with different names but common meaning, e.g., region vs. continent for the dimension location), homonyms (elements with same names but different meanings, e.g., the fact biting rate appearing in two different OSSs) are also performed.
10.5.4 Data Staging Area

Once the content of each OSS has been modeled, we perform extract-transformation-load (ETL) processes on the content. We extract the content by reading and understanding the source data, and perform data transformations by data cleansing (done primarily by correcting misspellings, dealing with missing or null elements, and parsing into standard formats), and assign DW keys. Then, following the logical model and schemas, we load the content into a *PostgreSQL* RDBMS \[190\] by using a Python SQL toolkit named *SQLAlchemy* \[223\]. To adhere to the key architectural requirements for the data staging area, we keep the PostgreSQL instance off-limits to our users, and isolate it from our query and presentation services.

10.5.5 Data Presentation Area

In the data presentation area, following the constellation schemas, we organize and store the data from our *PostgreSQL* RDBMS into an open-source OLAP Python framework named *cubes* \[48\]. The main features of *cubes* include multidimensional analysis, star and snowflake schema preparation and abstraction, etc. It is supported with a HTTP OLAP server (Slicer), which provides API for multidimensional data browsing for the DW. It also supports *views*, which are used to provide logical abstraction of the data. While loading the content into *cubes*, we ensure that the data marts contain the most detailed (i.e., *atomic*) data.

10.5.6 Data Access Tools

Currently, our data access tools include the web-based GUI frontend developed in Django (see Section \[10.6\]). The frontend is made available for direct querying by MalOLAP users. Aggregation operations (for selected data marts) are also supported.
10.6 The cubes and Django Frameworks

In this section, we elaborate on the two frameworks: cubes \([48]\) and Django \([54]\). We use cubes in the backend to implement the DW, and Django in the frontend to support the web-based user interfaces. Figure \([10.9]\) shows the data flow and usage model for MalOLAP, depicting the role of both frameworks.

10.6.1 The cubes Framework

The cubes is an open-source OLAP Python framework that uses a logical model to describe the organization, analysis, and reporting of content in a DW \([48]\). Being independent of the physical implementation, it describes the data from the user’s or analyst’s perspective, emphasizing on the aspects of measurements (facts), aggregations, and reports.

The logical model in cubes creates an abstraction layer in the DW’s data presentation area (see Section \([10.5]\)) using the JavaScript Object Notation (JSON) file format \([110]\), which is a lightweight, text-based, language-independent data interchange format. The use of JSON allows the cubes model to be independent of the physical structure of the data. Features of the logical model include dimensions with multiple hierarchies, user-oriented metadata, dimension templates to define complex dimensions, localization of model and data, etc.

In cubes, the dimensions can be shared by multiple data cubes, and thus belong to the model space. It supports both hierarchical and non-hierarchical dimensions (see Section \([8.3.3]\)). For hierarchical dimensions, the levels of the hierarchy are defined in the logical model. Examples of hierarchical dimensions include date \((year-month-day)\), location \((continent-country-district-region)\), etc. Examples of non-hierarchical dimensions include source, subgroup, active ingredient, chemical class, etc.

The data cubes are usually specified by a list of dimensions, facts, and aggregations. It uses a set of mapping rules to map logical attributes in the model into
Figure 10.9. Data Flow and Usage for MalOLAP. Data from heterogeneous OSSs are collected, transformed through a series of ETL operations by using customized Python scripts, and loaded into a relational PostgreSQL database. The ETL scripts, relational database, and the cubes framework together constitute the backend. The cubes framework stores the facts and dimensions in JavaScript Object Notation (JSON) format. Upon receiving requests from the slicer module, the aggregation browser module provides the results to the slicer module as either cells (slice and dice) or aggregates. The frontend, developed with the Django web application framework, receives requests from users via various Web browsers. The requests are sent to the slicer module, and the retrieved results are presented to the users.
physical attributes in the data cube table. Each mapping rule consists of a key-value pair, where the key refers to the attribute name of the dimension, and the value refers to the column name of the physical table. The correspondence between a fact table and a dimension table is specified by a join mapping rule. Each join consists of references to a master table (fact table) and a details table (dimension table).

One of the advantages of using the cubes framework is its resilience to changes: if the metadata and the logical model are properly defined and used in an application, then most of the model changes can be handled by the application without any modifications. This allows, for example, to add or remove levels from a dimension without changing the DW’s reporting applications. As described in Section 10.5.1, this supports our iterative development methodology, as we progressively advance the DW by incorporating new sources of data.

10.6.2 The Django Web Framework

We use the Django web framework [54] in the frontend to build our web-based user interfaces [147]. Django is a free and open-source web application framework, written in Python, that encourages rapid software development and clean, pragmatic design, and facilitates the tasks of creating complex, database-driven websites. It is maintained by the Django Software Foundation (DSF), which is an independent non-profit organization.

Django follows the model-view-controller (MVC) software architectural pattern, which separates the representation of information from the user’s interaction with it. The MVC pattern promotes code reusability and separation of concerns. In this pattern, the model consists of application data, business rules, logic, and functions. A view may represent any form of output (e.g., textual, chart, diagram, etc.), and multiple views of the same data are possible. The controller mediates input, converting it to commands for the model or view.
Using the MVC pattern allows *Django* to adhere to the principles of code reusability, components pluggability, rapid development, and the DRY (Don’t Repeat Yourself) principle. *Django* is equipped with a Python-based object-relational mapper database, and also officially supports four popular databases as backends (PostgreSQL, MySQL, SQLite, and Oracle).

10.7 Results

In this section, we describe some results and screenshots of the MalOLAP web portal [147]. Figure 10.10 depicts the MalOLAP homepage [147]. We organize the data marts and lookup tables under different categories, which currently include Mosquito Species, Vector Control Interventions, Household Surveys, Weather & Climate, Factors of Transmission, and Insecticides. The navigation bars (on the top and left) allow navigation to individual data marts and lookup tables.

The Vector Control Interventions category consists of three interventions: (1) ITNs, (2) IRS, and (3) ACT. The Weather & Climate category consists of two weather-related data marts: (1) Rainfall Data from PACRAIN, and (2) Weather Data from GSOD. The two lookup tables, *Species* and *Entomological Endpoints*, equipped with dictionaries *Species Parameters* and *Entomological Endpoints Phases*, respectively, can be accessed by links on the left navigation bar.

Currently, MalOLAP supports both non-aggregated and aggregated (summarized) data. As an example of non-aggregated data, Figure 10.11 depicts a screenshot of the fact table *WMR IRS*, which has two dimensions (see Figure 10.3): location and date, both of which are hierarchical. In the figure, attributes region and country represent the location dimension, and attribute year represents portion of the date dimension. The fact table contains two facts: 1) % IRS Coverage, and 2) Number of people protected by IRS.

As an example of aggregated data, Figure 10.12 depicts a screenshot of the fact
Figure 10.10. Screenshot of MalOLAP Homepage. All data marts and lookup tables are contained within different categories, which include Mosquito Species, Vector Control Interventions, Household Surveys, Weather & Climate, Factors of Transmission, and Insecticides. The navigation bars (on the top and left) allow navigation to individual data marts and lookup tables.
Figure 10.11. Screenshot of Non-Aggregated Data from the Fact Table WMR IRS as an Example. The fact table WMR IRS has two dimensions (see Figure 10.3): location and date, both of which are hierarchical. In the figure, attributes region and country represent the location dimension, and attribute year represents portion of the date dimension. The fact table WMR IRS contains two facts: 1) % IRS Coverage, and 2) Number of people protected by IRS. The figure depicts only the first few records of the fact table.
table WMR Household, which has four dimensions (see Figure 10.3): location, date, source, and subgroup. The aggregated data, as plotted in the bar chart, displays averages of three facts from the fact table WMR Household: 1) % of Household with \( \geq 1 \) any net, 2) % of Household with \( \geq 1 \) ever-treated net, and 3) % of Household with \( \geq 1 \) an ITN. The aggregated averages are calculated on the basis of regions (continents), where region is an attribute of the hierarchical dimension location. Current regions include Africa, Americas, Eastern, Western Pacific, and South-East Asia. The tooltip on the chart shows 27.85 as the average value of % of Household with \( \geq 1 \) an ITN for the region Africa. Portions of the non-aggregated data are also displayed below the chart.

Figures 10.13 and 10.14 depict examples of a lookup table and a dictionary, respectively. In Figure 10.13, the Species lookup table is shown. It contains species bionomics parameter values for use in the VECNet applications that require bionomic input. Figure 10.14 shows the dictionary Species Parameters, which contains definitions and explanatory descriptions of terms that appear in the Species table.

Figure 10.15 shows an example of the roll down operation. As described in Section 10.5.2, the roll up and roll down operations allow the user to navigate among levels of data ranging from the most summarized (up) level to the most detailed (down) level, along any hierarchical dimension. The roll down operation produces a more detailed view of data by rolling down one level along a hierarchical dimension. As shown in the figure, we roll down along the hierarchical dimension location of the fact table WMR IRS, from the summarized attribute continent (with value = Africa) to the detailed attribute country (with value = Angola). Query parameters for the roll down operation are shown in the first yellow box. The number of records is reduced from 360 (for all countries) to 8 (for Angola) due to the roll down.
Figure 10.12. Screenshot of Aggregated Data from the Fact Table WMR Household as an Example. The fact table WMR Household has four dimensions (see Figure 10.3): location, date, source, and subgroup. The aggregated data, as plotted in the bar chart, displays averages of three facts from the fact table WMR Household: 1) % of Household with >= 1 any net, 2) % of Household with >= 1 ever-treated net, and 3) % of Household with >= 1 an ITN. The aggregated averages are calculated on the basis of regions (continents), where region is an attribute of the hierarchical dimension location. Current regions include Africa, Americas, Eastern, Western Pacific, and South-East Asia. The tooltip on the chart shows 27.85 as the average value of % of Household with >= 1 an ITN for the region Africa. Portions of the non-aggregated data, from which cubes calculated the averages, are displayed below the chart.
Figure 10.13. Screenshot of Data from the Lookup Table Species as an Example. The Species lookup table contains species bionomics parameter values for use in the VECNet applications that require bionomic input.
Figure 10.14. Screenshot of Data from the Dictionary *Species Parameters* as an Example. The dictionary contains definitions and explanatory descriptions of terms that appear in the *Species* lookup table.
Figure 10.15. Screenshot of Roll Down Operation from the Fact Table WMR IRS as an Example. The fact table WMR IRS has two dimensions (see Figure [10.3]: location and date. The roll down operation produces a more detailed view of data by rolling down one level along a hierarchical dimension. In this figure, we roll down along the hierarchical dimension location, from the attribute continent (with value = Africa) to the attribute country (with value = Angola). Query parameters for the roll down operation are shown in the first yellow box. As the second yellow box shows, the number of records, due to the roll down, is reduced from 360 (for all countries) to 8 (for Angola).
10.8 Discussion

In this section, we describe some miscellaneous issues faced during the design and implementation phases. As stated before, MalOLAP serves as a pilot project with the goal to be integrated into the VECNet CI analytical framework [241]. Currently, features such as the aggregation operation, the generation of graphical output, etc. are implemented as representative samples for some data marts. However, once fully implemented, MalOLAP would produce dynamic graphical output for every possible input combination.

As part of the VECNet CI, MalOLAP uses open-source software components for its various elements. All software used are freely available, mature, and widely used, and provide a low risk, maintainable, and portable design.

In some cases, the constellation schema, which in effect represents multiple star schemas, may not represent the best design technique for the modeled data [188]. As we strive to achieve an appropriate balance between traditional databases and dimensional modeling techniques, in future, these situations would be handled by considering our DW needs as well as the performance criteria, rather than reliance upon technical panaceas (i.e., the star schema).

In most cases, unlike other biological DW projects (e.g., BioMart [113, 112, 259]), transformation of existing content from third normal form (3NF) into a DM schema was not necessary, because the bulk of the data sources we worked with so far came from data files which are stored in flat files, spreadsheets, etc. (e.g., non-RDBMS data sources). However, for query performance and optimization issues, we took the leverage of using an RDBMS as a backend to our multi-dimensional schema (see Figure 10.9).
10.9 Future Work

As MalOLAP experiences continuous cycles of improvement, we continue to follow the iterative development methodology, and progressively advance the DW by incorporating new data sources. We have several new features under development, especially regarding content integration with other modeling tools in VECNet, and the ingestion of predictive data, generated primarily as outputs of agent-based micro-simulations, with priorities given to two models: OpenMalaria [177] and EMOD [61].

We are also working on extending MalOLAP to support advanced data browsing techniques such as faceted navigation of attributes and filters (e.g., with elastic lists for browsing multi-faceted data structures, as in [59]) and to improve the geospatial aspects of navigation (e.g., by including customized geographical maps for data browsing, as in [1]). To improve the historical data representation, the spatial and temporal dimensions are also likely to be re-modeled.

10.10 Summary

In this chapter, we described a multi-dimensional, online analytical processing (OLAP) data warehouse named MalOLAP, designed specifically to store, access, and analyze malaria-related data, with the goal to be integrated into the VECNet CI analytical framework [241]. We described the heterogeneous biological data sources, the logical and physical modeling phases, and other implementation details. We also presented the MalOLAP web portal [147]. We discussed some issues faced while developing the DW, and pointed to some future directions.

MalOLAP, once fully implemented, can help the VECNet CI to translate stakeholder questions into appropriate analyses and visualization of model predictions. The power of the DW emerges from integrated querying of the different data marts, and by structuring those queries according to the desired dimensions.
CHAPTER 11

CONCLUSION

11.1 Overview

This dissertation has described the design and implementation of an agent-based model (ABM) of one of the major malaria vectors, *An. gambiae*, and a multi-dimensional data warehouse (DW) for malaria-related data. We elaborate on our conclusions in the following sections.

11.2 An ABM of *An. gambiae* (Part 1)

In **Part 1** of this dissertation (Chapters 2–7), which concerns agent-based modeling of malaria, we presented a general background about ABMs, malaria, and the applicability of ABMs for malaria (Chapter 2). In Chapter 3 we described a biological core model of *An. gambiae*, and the non-spatial ABM that is built according to the specification of the core model. In Chapter 4 we described the verification & validation (V&V) issues of various ABMs developed within our research group, and showed how we achieved a complete dock for the mosquito populations. Next, in Chapter 5 we discussed the importance of a spatial representation for the ABM, and described the implementation of a spatial extension. Using the spatial ABM, we demonstrated the effects of using different landscape patterns, variation in the relative size and density of the habitats, the overall capacity of the system, and the effects of spatial heterogeneity of the landscapes. We also described a landscape generator tool for the ABM.
As the impact of vector control interventions to reduce and control malaria in sub-Saharan Africa are being investigated by numerous studies, in Chapter 6, we analyzed the effects of larval source management (LSM) and insecticide-treated nets (ITNs), both in isolation and in combination, by using our spatial ABM. We also compared our results to those reported by previous ABM-based studies. We discussed multiple definitions of ITN coverage, defined three coverage schemes, and simulated these schemes. We also recommended some guidelines for future ABM modelers, summarizing the insights and experience gained from our work.

In the last chapter (Chapter 7) of Part 1, we described another example application of our spatial ABM by integrating it with a geographic information system (GIS). By overlaying custom spatial variables (outputs of the ABM) on the GIS maps, we showed the utility of the application as the maps revealed important biological insights. We argue that since no model-based malaria study has yet shown such integration, the general workflow of the robust integration framework described in Chapter 7 would help in replenishing the vacuum of knowledge which currently endures in this area of research.

11.3 A DW for Malaria Data (Part 2)

In Part 2 of this dissertation (Chapters 8–10), we discussed data warehouses (DWs) and dimensional modeling (DM). In Chapter 8, we presented a general background of DW/DM. We described the characteristics, goals, and components of a DW, and the differences between DWs and operational source systems. We also described the basic DW/DM concepts and vocabulary, and the four-step dimensional design process. In Chapter 9, we described several biological DWs and data storage systems for malaria-related data, and discussed some data integration challenges in bioinformatics.

In the last chapter (Chapter 10) of Part 2, we described the design and imple-
mentation of a multi-dimensional DW, named MalOLAP, for malaria-related data. We showed its use as an efficient tool to search, retrieve, aggregate, and visualize data accumulated from heterogeneous data storage systems. We also discussed the operational goals of the DW in the context of the VECNet CI \[241\] project.

11.4 Contributions

This dissertation has described the following contributions:

- description of a biological core model of one of the major malaria vectors \textit{An. gambiae}
- design and implementation of an ABM of \textit{An. gambiae} from the core model
- verification & validation of different agent-based implementations
- design and implementation of a spatial extension of the ABM
- implementation of a landscape generator tool (\textit{VectorLand}) to facilitate the parameter specification for the spatial ABM
- assessment of impact of vector control interventions
- integration of the ABM with a GIS
- description of data warehouses (DWs) and dimensional modeling (DM)
- description of biological DWs and data storage systems
- design and implementation of a multi-dimensional DW for malaria-related data

11.5 Future Work

The work presented in this dissertation may be continued further along several lines of research, which include:

- examining the impact of new, novel, alternative vector control intervention tools (e.g., spatial repellents), both in isolation and in combination with existing tools (e.g., ITNs, IRS, etc.) by using the spatial ABM
- extending the landscape generator tools to improve the modeling of habitat-based landscapes
• supporting the VECNet CI project [241] by dimensional modeling of other types of historical data (e.g., malaria field study data, *P. falciparum* parasite rate (*PfPR*) summary data, etc.)

• dimensional modeling of the predictive data (output of micro-simulations, such as OpenMalaria [177] and EMOD [61]), and ingesting the data into the data warehouse

• implementing advanced data browsing techniques (e.g., faceted navigation of attributes and filters), geo-spatial aspects of navigation, and other user-friendly features in the data warehouse
APPENDIX A

FLOWCHART FOR THE JAVA IMPLEMENTATION OF THE ABM

In this Appendix, we present the detailed flowchart for the Java implementation of the ABM (described in Chapter 3).
Initialize variables:
- Number of initial adult = 1000
- Number of aquatic habitats = 5
- Number of aquatic habitats a female tries (at most) each day to lay eggs = 3
- Rainfall coefficient = 1.0
- Mean for number of eggs to lay in oviposition (fecundity) = 80
- Standard deviation for number of eggs to lay in oviposition (fecundity) = 12
- Mean for temperature-dependent growth rate of larvae = 1.0
- Standard deviation for temperature-dependent growth rate of larvae = 0.1

Initialize agent lists:
- List of all aquatic habitats: aquaticHabitatList
- List of all adult mosquitoes: adultsList
- List of all eggs in the system: eggsList
- List of all larvae in the system: larvaeList
- List of all pupae in the system: pupaeList

Initialize system:
- Create Aquatic Habitats with capacities 5000, 5000, 10000, 20000 & 30000
- Create a graph of the population vs. simulation time (tick count)
- Create 1000 initial adults:
  - Create 500 male adults with age 0, state EMERGE
  - Create 500 female adults with age 0, state EMERGE
Step 1: Kill existing adults
- Build a hash (temporary) to construct and save adult age-groups
- Each bucket in the hash represents one age-group
- For each bucket, keys are ages of mosquitoes, values are indices of adultsList
- Iterate adultsList and save each mosquito index to the correct age-group
- Iterate over each age-group, one at a time

Kill adults from this age-group by Age-Specific Death Rate (DMRAdult)
- For an age-group, DMRAdult depends on population size of the adult age-group, groupSize
- Set $a = 0.1$, $B = 25$, $s = 0.1$
- Calculate how many adults to kill ($tokill$) from this age-group today:

$$DMR_{Adult} = \frac{\alpha \times \frac{\text{groupSize}}{\text{Age} + 1} \times \beta}{1 + \alpha \times s \times B(\text{Age + 1}) - 1}$$

$$tokill = DMR_{Adult} \times \text{groupSize}$$

- for $tokill$ times do {
  - Get a random index (of adult mosquito) from this age-group
  - If duplicate (i.e. the adult mosquito already dead), get a random index again, continue until unique
  - Mark the adult mosquito as 'dead'
}

- Now actually kill those adult mosquitoes marked above as 'dead'
- Use an iterator to iterate over adultsList, removing the 'dead' ones
Step 2a: Kill Eggs
- Eggs are kept globally in eggsList
- Calculate how many eggs to kill (tokill) from eggsList:
\[
DMR_{Egg} = 0.1
\]
\[
tokill = DMR_{Egg} \times eggsListSize
\]
- for tokill times do {
  - Get a random index (of egg) from eggsList
  - If duplicate (i.e. the egg already dead), get a random index again, continue until unique
  - Mark the egg as 'dead'
}
- Now actually kill those eggs marked above as 'dead'
- Use an iterator to iterate over eggsList, removing the 'dead' ones
- Adjust global eggsCount by decrementing tokill

Step 2b: Kill Larvae from all aquatic habitats

Kill Larvae from this aquatic habitat aH
- Build a hash (temporary) to construct and save larvae age-groups
- Each bucket in the hash represents one age-group of larvae
- For each bucket, keys are ages of larvae, values are indices of allLarvaeList
- Iterate allLarvaeList and save each larva-index to the correct age-group
- Calculate One-Day equivalent Larval Population \( N_e \):
\[
N_e = \sum_{Age=0}^{MaxAge} Age \times Larva_{Age}
\]
- Iterate over each age-group, one at a time
Kill larvae from this age-group

- For an age-group, \( DMR_{Larva} \) depends on population size of the larva
- age-group, \( groupSize \)
- Set \( a = 0.1, R = 1.0 \)
- Carrying capacity of this aquatic habitat: \( C \)
- Using One-Day equivalent Larval Population \( N_e \), calculate how many larvae to kill (toKill) from this age-group:

\[
DMR_{Larva} = a \times e^{\frac{N_e}{groupSize}}
\]

\[
toKill = DMR_{Larva} \times groupSize
\]

- for toKill times do {
  - Get a random index (of larva) from this age-group
  - If duplicate (i.e. the larva already dead), get a random index again, continue until unique
  - Mark the larva as 'dead'
}

- Now actually kill those larvae marked above as 'dead'
- Use an iterator to iterate over allLarvaeList, removing the 'dead' ones
- Also, remove those larvae from the global larvae list, larvaeList

Step 2: Kill Pupae

- Pupae are kept globally in pupaeList
- Calculate how many pupae to kill (toKill) from pupaeList:

\[
DMR_{Pupa} = 0.1
\]

\[
toKill = DMR_{Pupa} \times pupaeListSize
\]

- for toKill times do {
  - Get a random index (of pupa) from pupaeList
  - If duplicate (i.e. the pupa already dead), get a random index again, continue until unique
  - Mark the pupa as 'dead'
}

- Now actually kill those pupae marked above as 'dead'
- Use an iterator to iterate over pupaeList, removing the 'dead' ones
- Adjust global pupaeCount by decrementing toKill
Step 3a: Process existing eggs

for each egg in eggsList do {
    if (egg.wait == 0) {
        o get the aquatic habitat aH the egg is in
        o increment larvaeCount of aH by 1
        o decrement eggsCount of aH by 1
        o add (this new larva) to global list larvaeList & also to aHLarvaeList
        o remove egg from the list eggsList
        o change state to LARVA, set wait to 0
        o set cumulativeLarvalStateDelay to 0
    }
}

Step 3b: Process existing larvae

for each larva in larvaeList do {
    if (larva.wait == 0) {
        o update cumulativeLarvalStateDelay:
            cumulativeLarvalStateDelay += (temperature * 0.000305 - 0.003285) * 24
        o if (cumulativeLarvalStateDelay >= larvalStateDelay)
            change state to PUPA,
            set cumulativeLarvalStateDelay to 0
        o get the aquatic habitat aH the larva is in
        o increment pupaeCount of aH by 1
        o decrement larvaeCount of aH by 1
        o add (this new pupa) to global list pupaeList
        o remove from the list larvaeList & from aHLarvaeList
        o change state to LARVA
        o set wait to 1
    }
}
Step 3c: Process existing pupae

for each *pupa* in *pupaeList* do {
  if (pupa.wait == 0) {
    o get the aquatic habitat *Al* the pupa is in
    o decrement *pupaeCount* of *Al* by 1
    o add (this new adult) to global list *adultsList*
    o remove pupa from the list *pupaeList*
    o change state to *EMERGE*
    o generate a random uniform *probability* from (0.0, 1.0)
      * if (*probability* <= 0.1) set *wait* = 1
      * else if (*probability* <= 0.9) set *wait* = 2
      * else set *wait* = 3
  }
}

Step 4: Process existing adults

Process each *adult* in *adultsList*

Is *adult* MALE?

yes

no

adult.wait == 0?

yes

no

if (adult.state == *EMERGE*) {
  change state to *MATE*
}

adult.state == OVIPOSITION?

yes

no

*availableEggs* == 0?

yes

Page 7

no

*availableEggs* == 0?

no

yes

New oviposition:

- Generate a random normal count of *maxEggs* from, with mean 80 and standard deviation 12
- Set *availableEggs* = *maxEggs*
- Try to lay *availableEggs* eggs

Page 7
Lay availableEggs eggs

numTryAquaticHabitatsPerDay = 3
laidEggs = 0

for i = 1 to numTryAquaticHabitatsPerDay times do {
  availableEggs = availableEggs - laidEggs
  if (availableEggs == 0) // No more eggs to lay
  else
    Select an aquatic habitat aH at random
    • Calculate One-Day equivalent Larval Population \( N_e \) in aH:
      \[
      N_e = \sum_{Age=0}^{MaxAge} Age \times \text{Larvae}_{Age}
      \]
    • totalBioMass = aH.eggsCount + oneDayEquivalentLarvaePopulation + aH.pupaeCount
    • normalizedBiomass = totalBioMass / (i * aH.carryingCapacity)
    • \text{potentialEggs} = \text{floor( maxEggs * potentialBiomass )}
    if (availableEggs >= potentialEggs) laidEggs = potentialEggs
    else laidEggs = availableEggs
    • lay laidEggs number of eggs in this aquatic habitat with equal male-female distribution:
      for (j = 0; j < laidEggs; j++) {
        if ((j & 1) == 0) egg.Sex = MALE;
        else egg.Sex = FEMALE
        • create the egg, attach to this aquatic habitat
        • add the egg to \text{eggsList}
      }
    • increment aH.eggsCount by laidEggs
  }

Update egg-count of the female:
if (availableEggs > 0) {
  availableEggs = availableEggs - laidEggs
  - force the female to stay in OVIPOSITION for another day; set \text{wait} = 1
} else { set availableEggs to 0
}
if (adult.state == OVIPOSITION)
    • change state to BLOODMEAL
    • generate a random uniform probability from (0.0, 1.0)
    • if (probability <= 0.7) set wait = 2
    • else set wait = 3

E

adult.state == BLOODMEAL?

adult.state == GRAVID?

E

 adult.state == BLOODMEAL?

yes

Change state to GRAVID

no

D

E

Any adult left?

C

D

availableEggs == 0?

yes

D

no

Page 8
Step all agents:

for each egg in eggsList do {
  • egg.age = egg.age + 1
  • egg.wait = egg.wait - 1
}

for each larva in larvaList do {
  • larva.age = larva.age + 1
  • larva.wait = larva.wait - 1
}

for each pupa in pupaeList do {
  • pupa.age = pupa.age + 1
  • pupa.wait = pupa.wait - 1
}

for each adult in adultsList do {
  • adult.age = adult.age + 1
  • adult.wait = adult.wait - 1
}

Print today's statistics:
• Print number of female adults to graph
• Print number of male adults to graph

Increment simulation day count

Maximum simulation day reached?

End
APPENDIX B

EFFECT OF TEMPERATURE ON VECTOR GROWTH AND DEVELOPMENT KINETICS

B.1 Overview

This appendix describes the theoretical background on the temperature-regulated rules and equations that we use in the models (both the core model and the ABM). Since malaria vectors are poikilothermic\(^1\), temperature is a critical variable in the growth and development kinetics of An. gambiae, and hence in malaria epidemiology modeling. In fact, temperature is one of the most influential parameters that can affect the rates of growth and development of mosquitoes and hence the vector abundance. For instance, in the range of 18°C to 26°C, a change of only 1°C in temperature can change a mosquito’s life span by more than a week \(^{51}\). Understanding this theoretical background is crucial to select the appropriate temperature model for the agents in our ABM.

The organization of this appendix is as follows: Section B.2 discusses some recent works on stochastic thermodynamic models. Section B.3 discusses the origin of historical models for organism development, including the Arrhenius plots and the Arrhenius equation. The Eyring Equation \(^{[B.3]}\) is derived next. Concepts of Gibbs free energy, entropy and enthalpy are then sequentially incorporated into the equation, yielding the final form of the Eyring Equation \(^{[B.6]}\).

\(^1\)A poikilotherm is a plant or animal whose internal temperature varies along with that of the ambient environmental temperature.
Section B.4 discusses the model developed by Sharpe and DeMichele [211], deriving Equation (B.10) as the final equation of development for this model. Section B.5 discusses the non-linear regression model derived by Schoolfield et al. [207]. Starting with Equation (B.10) from Section B.4, it re-formulates the model by using non-linear regression. Section B.6 discusses the model developed by Depinay et al. [51], and concludes with the major findings that are directly affected by temperature.

B.2 Literature Review

Sharpe and DeMichele [211] develop a stochastic thermodynamic model of poikilothermic development from the Eyring equation, assuming multiple activity states of the underlying developmental control enzymes. However, the model is not well-suited for non-linear regression. To alleviate this, the model derived by Schoolfield et al. [207] describes a new formulation of the Sharpe and DeMichele model using non-linear regression techniques. The model is partly based on Hultin’s formulation [103]. It discusses biological and graphical interpretation of the model parameters, and illustrates regression suitability with a typical data set.

Depinay et al. [51] presents a model simulating the *Anopheles* population dynamics by incorporating biological and environmental variables. It uses the enzyme kinetics model derived by Sharpe and DeMichele [211], based on absolute reaction rate kinetics of enzymes for the temperature-dependent developmental rates of eggs, larvae and pupae and the duration of the gonotrophic cycle, in the simplified form derived by Schoofield et al. [207]. It focuses on two abiotic factors (temperature and moisture) and three biotic factors (nutrient competition, predation or death by disease, and dispersal).

Bayoh and Lindsay [19] examine the influence of temperature on the survival of larval stages (larvae and pupae) of *An. gambiae* Giles sensu stricto and subsequent adult production. They observe groups of 30 mosquitoes in the laboratory at constant
temperatures (from 10 to 40 °C) from the first instar and observed until death or metamorphosis of the last individual.

B.3 Poikilothermic Development Models

Among numerous empirical formulations of organism development, the following two are more relevant to poikilothermic development [211]:

1. The day-degree or temperature summation model:
   - proposed by Candolle [49] and Reibisch [195]
   - approximates observed values within certain temperature limits
   - assumes that the rate of development is proportional to temperature:
     \[ k = b(T - T_a) \]
     where \( k \) is the rate of development, \( b \) is a constant, \( T \) is the absolute temperature and \( T_a \) is the temperature at the developmental zero

2. The non-linear temperature inhibition model:
   - derived by Johnson & Lewin [111] for high temperatures and by Hultin [103] for low temperatures
   - describes the inhibiting effect of either high or low temperatures on organism development

A model comprised of a linear response in mid-temperature ranges and non-linear temperature inhibition (as modeled by the above two formulations) in high- and low-temperature ranges is developed by Sharpe and DeMichele [211]. To deduce the formula for the transition rate constant, it uses the Arrhenius equation and the Eyring equation, as well as the concepts of entropy and enthalpy. However, the latter two originate from the concept of the Gibbs free energy, which is described below.

The Sharpe and DeMichele model [211] is not well-suited for non-linear regression. To alleviate this, the model derived by Schoolfield et al. [207] uses non-linear regression techniques.
B.3.1 Loglinear Models

Loglinear models postulate a linear relationship between the independent variables and the logarithm of the dependent variable. As shown in the following subsections, the temperature-dependent rate equations are usually plotted using the loglinear models. Specifically, Equation (B.2) explains why these equations bear the loglinear values in the ordinate axis and inverse temperature values in the abscissa.

B.3.2 The Arrhenius Model

Temperature governs the rate of a chemical reaction. In 1889, Swedish scientist Svante Arrhenius first provided a physical justification and interpretation for this. The Arrhenius plots and the Arrhenius Equation, both named after him, are used to define this relationship quantitatively.

At higher temperatures, the probability that two molecules will collide is higher. This higher collision rate results in a higher kinetic energy, which has an effect on the activation energy of the reaction. The activation energy is the amount of energy required to ensure that a reaction happens.

The extent of temperature inhibition on the rates of chemical reactions can be shown by an Arrhenius plot. It displays the logarithm of a rate (\(\ln(k)\), ordinate axis) plotted against inverse absolute temperature (\(1/T\), abscissa). For a single rate-limited thermally activated process, an Arrhenius plot gives a straight line, from which the activation energy and the pre-exponential factor can both be determined. The original Arrhenius Equation is as follows:

\[
k = Ae^{-E_a/RT}
\]  

where \(k\) is the rate constant, \(A\) is the pre-exponential factor, \(E_a\) is the activation energy, \(R\) is the gas constant and \(T\) is the absolute temperature. Taking natural
logarithm to Equation (B.1) yields:

\[ \ln(k) = \ln(A) - \frac{E_a}{R} \left( \frac{1}{T} \right) \]  

(B.2)

Thus, in an Arrhenius plot, the value of the “y-intercept” corresponds to \( \ln(A) \), and the gradient of the line equals to \(-E_a/R\). The gradient \((-E_a/R)\) represents the fraction of the molecules present in a gas which have energies equal to or in excess of activation energy at a particular temperature. The ordinate axis \((\ln(k))\) denotes the specific growth rate \((\text{time}^{-1})\) and the abscissa \(1/T \text{ (°K}^{-1})\) denotes the reciprocal absolute temperature.

Other terms of interest are interpreted as:

- the pre-exponential factor, \(A\): a constant of proportionality that describes a number of factors such as the frequency of collision between and the orientation of the reacting particles (often taken as constant across small temperature ranges)
- the activation energy, \(E_a\): this is the minimum energy needed for the reaction to occur, expressed in joules per mole
- the gas constant, \(R\): this comes from the Ideal gas law which relates the pressure, volume and temperature of a particular number of moles of gas: \(pV = nRT\) where \(p\), \(V\) and \(n\) are the absolute pressure, volume and amount of substance (usually measured in moles) of the gas, respectively; \(R\) is the gas constant \((8.314472 \text{ joules per °K per mole})\) and \(T\) is the absolute temperature

Thus, the Arrhenius Equation (B.2) shows the effect of a change of temperature on the rate constant, and thus on the rate of the reaction. For example, if the rate constant doubles, the rate of the reaction would also almost double.

B.3.3 The Eyring Equation

The Eyring equation, developed in 1935 by Henry Eyring, is trivially equivalent to the Arrhenius Equation (B.2) and relates the reaction rate to temperature. The
The basic form of the equation is:

\[ k = \frac{k_B T}{h} e^{-\frac{\Delta G^\dagger}{RT}} \] (B.3)

where \( k \) is the rate constant, \( k_B \) is Boltzmann’s constant \((1.380 6504(24) \times 10^{-23} \text{ J K}^{-1})\), \( h \) is Planck’s constant \((6.626 068 96(33) \times 10^{-34} \text{ J s})\), \( \Delta G^\dagger \) is the Gibbs energy of activation (the Gibbs free energy), \( R \) is the gas constant and \( T \) is the absolute temperature.

To determine whether a reaction will occur or not, both enthalpy and entropy changes are important. The Gibbs free energy establishes a relationship between enthalpy and entropy, and hereby accommodates these into the Eyring Equation (B.3).

### B.3.4 The Gibbs Free Energy, Entropy and Enthalpy

The *Gibbs Free Energy*, developed in the 1870s by the American mathematician Josiah Willard Gibbs, is a thermodynamic potential that measures the *useful* or process-initiating work obtainable from an isothermal, isobaric thermodynamic system. It is the maximum amount of non-expansion work that can be extracted from a closed system and can be attained only in a completely reversible process. When a system changes from a well-defined initial state to a well-defined final state, the Gibbs free energy, \( \Delta G^\dagger \), equals the work exchanged by the system with its surroundings, less the work of the pressure forces, during a reversible transformation of the system from the same initial state to the same final state \[155\]. The term *free* was attached to mean *available in the form of useful work*, for systems at constant pressure and temperature. The Gibbs Free Energy is defined as:

\[ G = U + pV - TS \]
which is the same as:

\[ G = H - TS \]  \hspace{1cm} (B.4)

where \( U, p, V, T, S \) and \( H \) denote internal energy (in joule), pressure (in pascal), volume (in \( m^3 \)), temperature (in \( ^\circ K \)), entropy (in joule per \( ^\circ K \)) and enthalpy (in joule), respectively.

*Entropy* is the quantitative measure of disorder in a system. It measures how much of the energy of a system is potentially available to do work and how much of it is potentially manifest as heat. The concept comes from thermodynamics, and *the second law of thermodynamics* can be stated as: *In any closed system, the entropy of the system will either remain constant or increase.* It is also (imprecisely) known as disorder, chaos, randomness etc. In an isothermal process, the change in entropy (\( \Delta S \)) is the change in heat (\( \delta Q \)) divided by the absolute temperature (\( T \)):

\[ \Delta S = \frac{\delta Q}{T} \]

The SI unit of entropy is joule per \( ^\circ K \).

*Enthalpy* is a thermodynamic property of a thermodynamic system. It can be used to calculate the heat transfer during a process taking place in a closed thermodynamic system under constant pressure (isobaric process). The enthalpy change \( \Delta H \) is equal to the change in the internal energy of the system, plus the work that the system has done on its surroundings. The change in enthalpy under such conditions is the heat absorbed by a chemical reaction [254].

### B.3.5 Incorporating Entropy and Enthalpy into Eyring Equation

For constant temperature (\( T \)), Equation \((B.4)\) can be written as:

\[ \Delta G = \Delta H - T\Delta S \]  \hspace{1cm} (B.5)
Free energy change is the net driving force of a chemical reaction - it determines whether the reaction will be spontaneous or not. Thus, once $\Delta G$ is calculated, spontaneity can be predicted as follows: if $\Delta G < 0$: The reaction is spontaneous if $\Delta G > 0$: The reaction is non-spontaneous if $\Delta G = 0$: The reaction is at equilibrium

Replacing $\Delta G$ from Equation (B.5) into Equation (B.3) yields:

$$k = k_B T \frac{e^{-\frac{\Delta H^\dagger - T \Delta S^\dagger}{RT}}}{h}$$

$$= k_B T \frac{e^{\left(\frac{\Delta S^\dagger}{R} - \frac{\Delta H^\dagger}{RT}\right)}}{h}$$

$$= k_B T \frac{e^{(\Delta S^\dagger - \Delta H^\dagger / T) / R}}{h}$$

Thus, for each possible transition (see Figure B.1), $k_i$, for $i = 1, 2, 3, 4$, we get:

$$k_i = \frac{k_B T}{h} e^{(\Delta S^\dagger - \Delta H^\dagger / T) / R} \quad (B.6)$$

B.4 The Sharpe and DeMichele Model

From the Eyring-Gibbs Equation (B.6), Sharpe and DeMichele derived a stochastic thermodynamic model of poikilothermal development.

B.4.1 Energy States

To calculate the probability that the control enzyme is in an active state, Sharpe and DeMichele defined three energy states and possible transitions between them (see Figure B.1):

- **Energy State 1**: predominates at low temperatures
- **Energy State 2**: represents the active enzyme configuration; predominates over the mid-temperature range
- **Energy State 3**: predominates at high temperatures

Other important assumptions made by the model are:
Figure B.1. Energy States of the Kinetic Model (redrawn from [211]). The directed arrows indicate possible transitions. *Energy State 1* predominates at low temperatures. *Energy State 2* represents the active enzyme configuration, and predominates over the mid-temperature range. *Energy State 3* predominates at high temperatures.
• at any temperature, the probability of being in any specific form is less than one, and the cumulative probability of being in energy state 1, 2 or 3 is equal to one

• the probability that the enzyme molecule at any given temperature $T$ can move from State 1 directly to State 3 is negligible; therefore, to move from State 1 to 3 and vice versa, the enzyme must pass through State 2

• transitions between states are randomly distributed with a mean transition rate specified by the form of the Eyring-Gibbs Equation (B.6), with transition rate constants $k_i$

B.4.2 Exponential Distribution of Transition Times

If no transition between states of the enzyme occurred during time interval $\Delta t$, the probability of a transition occurring in the next time interval is not changed. Thus, the time between transitions becomes exponentially distributed. This implies that the probability of a transition occurring during a very short time interval $\Delta t$ is $k_i \Delta t$. As $\Delta t$ becomes infinitely small ($\Delta t \rightarrow 0$), the probability of more than one transition occurring during the time interval becomes infinitely small. Therefore, the probability of no transitions occurring during $\Delta t$ becomes approximately $\{1 - k_i \Delta t\}$.

B.4.3 Probability Calculations

The probability of the enzyme being in state 1 at time $t + \Delta t$, $P_1(t + \Delta t)$, can be calculated by summing up the probability of being in State 1 at time $\Delta t$ (i.e. $(1 - k_1 \Delta t)$) and the probability of moving from State 2 to State 1 at time $\Delta t$:

$$P_1(t + \Delta t) = P_1(t)(1 - k_1 \Delta t) + P_2(t)k_2 \Delta t$$

Re-arranging, we get the rate of change of the probability of being in State 1:

$$\frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = -k_1 P_1(t) + k_2 P_2(t)$$
Taking the limit as \( \Delta t \to 0 \):

\[
\frac{dP_1(t)}{dt} = -k_1 P_1(t) + k_2 P_2(t)
\]

Similarly, the rates of change of the probability of being in States 2 and 3 become:

\[
\frac{dP_2(t)}{dt} = k_1 P_1(t) - (k_2 + k_3) P_2(t) + k_4 P_3(t)
\]

\[
\frac{dP_3(t)}{dt} = k_3 P_2(t) - k_4 P_3(t)
\]

For steady-state conditions, all rates of change become 0; thus, equating \( \frac{dP_i(t)}{dt} = 0 \) for \( i = 1, 2, 3 \), and then solving for \( P_2 \), we get:

\[
P_2 = \frac{1}{1 + \frac{k_2}{k_1} + \frac{k_3}{k_4}}
\]

Substituting the Eyring-Gibbs Equation (B.6) for \( k_1, k_2, k_3, k_4 \) yields:

\[
P_2 = \frac{1}{1 + \frac{k_B T \exp\left(\frac{\Delta S^\ddagger_2 - \Delta H^\ddagger_2/T}{R}\right)}{\exp\left(\frac{\Delta S^\ddagger_1 - \Delta H^\ddagger_1/T}{R}\right)} + \frac{k_B T \exp\left(\frac{\Delta S^\ddagger_3 - \Delta H^\ddagger_3/T}{R}\right)}{\exp\left(\frac{\Delta S^\ddagger_1 - \Delta H^\ddagger_1/T}{R}\right)}}
\]

\[
(B.7)
\]

At equilibrium in a given temperature, only the differences in entropy of activation and enthalpy of activation need be considered. For all the three states, these terms are defined in Table B.1. Using these activation terms, Equation (B.7) can be further simplified as:

\[
P_2 = \frac{1}{1 + \exp\left(\frac{\Delta S_L - \Delta H_L/T}{R}\right) + \exp\left(\frac{\Delta S_H - \Delta H_H/T}{R}\right)}
\]

\[
(B.8)
\]

\( P_2 \) represents the probability that the developmental enzyme will be in the active state and thus affect the developmental process.
TABLE B.1

ENTROPY AND ENTHALPY OF ACTIVATION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta S_L$</td>
<td>Difference in entropy of activation</td>
<td>State 1 and State 2</td>
</tr>
<tr>
<td>$\Delta H_L$</td>
<td>Difference in enthalpy of activation</td>
<td>State 1 and State 2</td>
</tr>
<tr>
<td>$\Delta S_H$</td>
<td>Difference in entropy of activation</td>
<td>State 2 and State 3</td>
</tr>
<tr>
<td>$\Delta H_H$</td>
<td>Difference in enthalpy of activation</td>
<td>State 2 and State 3</td>
</tr>
</tbody>
</table>

To calculate the rate of development, Sharpe and DeMichele [21] define four additional terms, which we describe in Table B.2. The last term, $k'_2$, assumes no enzyme inactivation, and is described by the Eyring-Gibbs Equation (B.6). Then, the rate of development, $R_D$, becomes:

$$R_D = \varepsilon_c k'_2 P_2$$

Substituting for $k'_2$ from Equation (B.6) and $P_2$ from Equation (B.8):

$$R_D = \frac{\varepsilon_c k_B T e^{(\Delta S^\neq_A - (\Delta H^\neq_A / T)) / R}}{1 + e^{(\Delta S_L - \Delta H_L / T) / R} + e^{(\Delta S_H - \Delta H_H / T) / R}}$$  \hspace{1cm} (B.9)

The unknown thermodynamic constants $\varepsilon_c$ and $\Delta S^\neq_A$ can be summarized by $\phi$, where:

$$\phi = \Delta S^\neq_A + \ln(k_B \varepsilon_c / h)$$

Taking the exponential:

$$e^\phi = e^{\Delta S^\neq_A} k_B \varepsilon_c / h$$

279
TABLE B.2

ENZYME, REACTION AND RATE CONSTANT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_c$</td>
<td>Total concentration of the control enzyme</td>
</tr>
<tr>
<td>$\Delta S_A^#$</td>
<td>Entropy of activation of the reaction</td>
</tr>
<tr>
<td>$\Delta H_A^#$</td>
<td>Enthalpy of activation of the reaction</td>
</tr>
<tr>
<td>$k_2'$</td>
<td>Rate constant for development</td>
</tr>
</tbody>
</table>

Replacing $e^\phi$ in Equation \[B.9\]:

$$R_D = \frac{T \cdot e^{(\phi - \Delta H_A^\# / T) / R}}{1 + e^{(\Delta S_L - \Delta H_L / T) / R} + e^{(\Delta S_H - \Delta H_H / T) / R}}$$  \[B.10\]

which is the final equation for development.

B.5 The Schoolfield et al. Model

The original formulation of Sharpe and DeMichele is not well-suited for non-linear regression, which is needed to fit any kinetic model to observed growth and development rate. The model derived by Schoolfield et al. [207] uses non-linear regression techniques. It also discusses the biological and graphical interpretations of the parameters.

The model starts from Equation \[B.10\] by denoting the mean development rate at temperature $T$ as $r(T)$:

$$r(T) = \frac{T \cdot e^{(\phi - \Delta H_A^\# / T) / R}}{1 + e^{(\Delta S_L - \Delta H_L / T) / R} + e^{(\Delta S_H - \Delta H_H / T) / R}}$$  \[B.11\]

It defines three new thermodynamic parameters to replace three parameters in Equa-
tion (B.11): $\rho(25^\circ C)$, $T_{1/2_L}$ and $T_{1/2_H}$.

$\rho(25^\circ C)$ relates the standard reference temperature (25°C) at which most poikilothersms experience little low or high temperature enzyme inactivation. 25°C is used as a standard reference temperature in many scientific disciplines. $\rho(25^\circ C)$ is defined as:

$$\rho(25^\circ C) = 298 * e^{(\phi - \Delta H^\neq_A/298)/R}$$

where $\phi$, $\Delta H^\neq_A$ and $R$ are as defined before. Solving for $\phi$, we get:

$$\phi = \frac{\Delta H^\neq_A}{298} + R * \ln \frac{\rho(25^\circ C)}{298} \quad (B.12)$$

Substituting $\phi$ from Equation (B.12) into the numerator of Equation (B.11), the new numerator becomes:

$$T * e^{\frac{\Delta H^\neq_A}{298} + R * \ln \frac{\rho(25^\circ C)}{298} - \frac{\Delta H^\neq_A}{298}}$$

$$= T * e^{\frac{\Delta H^\neq_A}{298}} * \frac{\rho(25^\circ C)}{298} * e^{-\frac{\Delta H^\neq_A}{RT}}$$

$$= \frac{T * \rho(25^\circ C)}{298} * e^{-\frac{\Delta H^\neq_A}{RT}} \left(\frac{1}{298} - \frac{1}{T}\right) \quad (B.13)$$

$T_{1/2_L}$ and $T_{1/2_H}$ were defined by Hultin [103] as the ratio of difference in enthalpy to difference in entropy at low- and high-temperature inactivation, respectively:

$$T_{1/2_L} = \frac{\Delta H_L}{\Delta S_L}$$

$$T_{1/2_H} = \frac{\Delta H_H}{\Delta S_H}$$
Solving for $\Delta S_L$ and $\Delta S_H$:

$$\Delta S_L = \frac{\Delta H_L}{T_{1/2_L}}$$

$$\Delta S_H = \frac{\Delta H_H}{T_{1/2_H}}$$

Replacing $\Delta S_L$ and $\Delta S_H$ into the denominator of Equation (B.11), the new denominator becomes:

$$1 + e^{\frac{\Delta H_L}{R}\left(\frac{1}{T_{1/2_L}} - \frac{1}{T}\right)} + e^{\frac{\Delta H_H}{R}\left(\frac{1}{T_{1/2_H}} - \frac{1}{T}\right)}$$  \hspace{1cm} (B.14)

Assembling the new numerator and denominator from Equations (B.13) and (B.14) into Equation (B.11), we get:

$$r(T) = \frac{\rho_{(25^\circ C)}\frac{T}{298}e^{\frac{\Delta H_L}{R}\left(\frac{1}{298} - \frac{1}{T}\right)}}{1 + e^{\frac{\Delta H_L}{R}\left(\frac{1}{T_{1/2_L}} - \frac{1}{T}\right)} + e^{\frac{\Delta H_H}{R}\left(\frac{1}{T_{1/2_H}} - \frac{1}{T}\right)}}$$  \hspace{1cm} (B.15)

which is the final form of the modified equation.

B.6 The Depinay et al. Model

Depinay et al. [51] present a model simulating the *Anopheles* population dynamics by incorporating biological and environmental variables. The model focuses on two abiotic factors (temperature and moisture) and three biotic factors (nutrient competition, predation or death by disease, and dispersal). In this appendix, only the effect of temperature is discussed.

In this model, the mosquito life cycle has four stages: three immature stages (egg, larva, pupa) occurring in a water body and then the mature stage (flying adult). An adult female disperses from the water body and begins a cycle which is maintained throughout the rest of her life, alternating between obtaining a bloodmeal...
and ovipositing in a water body.

Temperature is included as a critical regulator of growth and development within each stage, in determining the end of one stage and the beginning of the next and in regulating the length of the gonotrophic cycle. The model uses the enzyme kinetics model derived by Schoolfield et al. (see Section B.5) to realize the temperature-dependent developmental rates of eggs, larvae and pupae and the duration of the gonotrophic cycle, with the basic assumption that poikilothermic development is regulated by a single control enzyme whose reaction rate determines the development rate of the organism.

B.6.1 Cumulative Development

The model directly uses Equation (B.15) to define \( r(T_{tk}) \), the developmental rate per hour at temperature \( T \) (°K), where \( T_{tk} \) is the mean temperature (°K) over the time interval \( k \). All other parameters of Equation (B.15) are defined as above.

At time step \( t_n \) of \( t_0, t_1, \ldots, t_n \), the development within each of the four stages, during the time step \( \Delta t_k = t_k - t_{k-1} \), is defined by:

\[
d_k = r(T_{tk}) \times \Delta t_k \tag{B.16}
\]

The cumulative development (of each of the three immature stages and the length of the adult gonotrophic cycle) depends only on temperature. For each time step \( t_n \), it is defined as:

\[
CD(t_n) = \sum_{k=1}^{n} d_k \tag{B.17}
\]

with \( d_k \) defined as in Equation (B.16).

To allow variability (10%) in the cumulative development time, a normal random variable \( G \) is defined. A stage is considered completed and the next stage begins
when:

\[ CD(t) > CD_f = 1 + G(0, 0.1) \]  \hspace{1cm} (B.18)

The model uses Equation (B.15) to produce three curves by varying the equation parameters. They find that all three curves provide similar fits to the An. gambiae relevant published data, and select the middle one for further analysis.

However, as noted by the authors, these different curves have important implications for vector population dynamics, and reinforce the need for more data for these species, particularly at the temperature extremes (low and high). Since any number of curves might fit the data, in order to fit an optimal curve, more data for the extreme temperatures is needed.

The thermal death point for An. gambiae is 40°C, which is the reported temperature in most small pools. To model the effect of varying water temperature on mortality, the model considers a daily larval mortality of 10%, 50% and 100% for a maximum water temperature rise of 1, 2 and 3°C, respectively, above the thermal death point.

B.6.2 Results

As an example run, the model simulates over a 20-month period on a small cluster of six houses, each with five residents, and a total of three oviposition sites: a semi-permanent pool \( P_1 \), and two temporary pools \( P_2 \) and \( P_3 \). Each female mosquito chooses at random among oviposition sites and among houses and residents at different points in her gonotrophic cycle. Temperature inputs were obtained from data on Kilifi, on the coast of Kenya.

The first set of simulations consist of 300 eggs and 10 adults, with all six houses but only pool \( P_1 \) present. It considers the stochastic effect allowed in the cumulative development time, length of the initial gonotrophic cycle and number of eggs
oviposited per female. The resulting *An. gambiae* adult abundance (mean) curve shows similarities to several published trends in the literature, specifically in that there are relatively low levels of mosquitoes throughout the year, with fluctuations due to competition and/or predation and several high peaks in short time intervals.

To analyze the effects of temperature, two additional temperature curves, one with $2^\circ$C increase and the other with $2^\circ$C decrease in the actual temperatures, are used. It reveals that:

- with increasing temperature, the level and number of peaks are increased, and the egg-to-adult development time is shortened, thus producing more mosquitoes
- the $2^\circ$C rise increases *An. gambiae* adult abundance by 15%, and the $2^\circ$C drop decreases it by 17% overall

At the end, the model admits temperature as an important factor for the adult abundance curve and, particularly, to the occurrence of the initial peak after a drought period (simulated in the 20-months period), which might be critical for control purposes.

B.7 Summary

In this appendix, we described the effect of temperature on the growth and development kinetics of *An. gambiae*. We discussed some recent stochastic thermodynamic models, including the origin of historical models for organism development, the Arrhenius plots, and the Arrhenius equation. We derived the Eyring Equation (B.3) from the Arrhenius equation, and incorporated the concepts of Gibbs free energy, entropy and enthalpy into the Eyring Equation.

We then discussed the Sharpe and DeMichele [211] model, and the non-linear regression model derived by Schoolfield et al. [207]. Lastly, we discussed the Depinay et al. [51] model, analyzed the effects of temperature, and concluded with the major findings of the model.
APPENDIX C

ADDITIONAL MATERIALS FOR CHAPTER 6

In this appendix, we present additional materials for Chapter 6.
Figure C.1. The Landscapes Digitized from the GN-LSM Study [86]. The 40 × 40 grid-based landscapes, digitized and reproduced from the GN-LSM study [86], by using our landscape generator tool, VectorLand. Each landscape contains 70 aquatic habitats (blue circles), and 20 houses (black house icons). Within each landscape, the houses are arranged either diagonally, horizontally, or vertically. For each arrangement, seven scenarios of LSM are shown; from left to right: NOCTRL (no LSM), T1, T2, T3, C1, C2, C3. T1, T2 and T3 refer to targeted removal of aquatic habitats within 100, 200 and 300 m of surrounding houses, accounting for 4, 17 and 28 of 70 habitats, respectively. C1, C2 and C3 refer to non-targeted, random removal of the same numbers of aquatic habitats as the corresponding targeted interventions.
Figure C.2. The Landscape Digitized from the GN-ITN Study [85]. The 40 × 40 grid-based landscape, digitized and reproduced by using our landscape generator tool, VectorLand. It contains 90 aquatic habitats (blue circles) which are randomly distributed, and 50 houses (black house icons) which are arranged diagonally.

Figure C.3. Full One-Year Results Showing the Impact of LSM, Applied in Isolation, on Abundance, Using an Absorbing Boundary. We compare the results with the GN-LSM study [86]. For details about the LSM scenarios used in the sub-figures, see legend of Figure 6.5.
Figure C.4. Full One-Year Results Showing the Impact of LSM, Applied in Isolation, on Abundance, Using a Non-absorbing Boundary. We compare the results with the GN-LSM study [86]. For details about the LSM scenarios used in the sub-figures, see legend of Figure 6.6.
Figure C.5. Impact of ITNs, One-Year Results, Using Partial Coverage with Single Chance for Host-seeking. Full one-year results showing the impact of ITNs (applied in isolation). Each row represents a specific coverage (C) value for ITNs (e.g., C = 0.8). Each column represents a specific repellence (R) value for ITNs (e.g., R = 0.5). Within each sub-figure, each color-coded plot represents a specific mortality (M) value for ITNs (e.g., M = 0.25), with mortality (M) color keys at the bottom of the figure. For other details, see Figures 6.1 and 6.8.
Figure C.6. Impact of ITNs, One-Year Results, Using Partial Coverage with Multiple Chances for Host-seeking. Full one-year results showing the impact of ITNs (applied in isolation). Each row represents a specific coverage (C) value for ITNs (e.g., C = 0.8). Each column represents a specific repellence (R) value for ITNs (e.g., R = 0.5). Within each sub-figure, each color-coded plot represents a specific mortality (M) value for ITNs (e.g., M = 0.25), with mortality (M) color keys at the bottom of the figure. For other details, see Figures 6.2 and 6.8.
Figure C.7. Impact of ITNs, One-Year Results, Using Complete Coverage. Full one-year results showing the impact of ITNs (applied in isolation). Each row represents a specific coverage (C) value for ITNs (e.g., C = 0.8). Each column represents a specific repellence (R) value for ITNs (e.g., R = 0.5). Within each sub-figure, each color-coded plot represents a specific mortality (M) value for ITNs (e.g., M = 0.25), with mortality (M) color keys at the bottom of the figure. For other details, see Figures 6.3 and 6.9.
APPENDIX D

ADDITIONAL MATERIALS FOR CHAPTER 7

In this appendix, we present additional materials for Chapter 7.
Figure D.1. Clipped Water Sources and Villages Projections for Kenya.

The figure on the left shows different water source features (rivers, wetlands, and other types of water-points) clipped within Kenya. The figure on the right shows villages clipped within Kenya.
Figure D.2. Polygon Creation Process in ArcGIS. A new shapefile is created for the desired polygon. The boundary for the polygon is created in the editing mode, and the area of the polygon is calculated using the *Calculate Areas* tool using the ArcGIS software [4]. Finally, the polygon area is highlighted in the editing mode.
Figure D.3. Clipped Habitats within the Selected Polygon. A new shapefile is created for the desired polygon. The boundary for the polygon is created in the editing mode, and the area of the polygon is calculated using the Calculate Areas tool using the ArcGIS software [4]. Finally, the polygon area is highlighted in the editing mode.
Figure D.4. Data Conversion to Raster Format, Part 1. The figure shows the value and count of house features in the attribute table after raster conversion. From these attributes, the count of features per grid cell is generated.
Figure D.5. Data Conversion to Raster Format, Part 2. The figure shows all the features converted in raster format within the polygon area for the ABM. The interpretation of each ID, as supplied by the ABM, of each feature and count of features per grid cell is also linked.
APPENDIX E

ADDITIONAL MATERIALS FOR CHAPTER 10

In this appendix, we present additional materials for Chapter 10.
Figure E.1. Detailed Dimensions Location, Date, Source, and Subgroup. These dimensions are attached to fact tables Household Surveys and Operational Coverages of interventions (ITNs, IRS, and ACT), as shown in Figures E.2, E.3, and 10.3.

Figure E.2. The Fact Table Household Surveys (of Mosquito Nets Ownership and Usage) with Dimensions Location, Date, Source, and Subgroup. All facts of the fact table are shown. The fact table appears in the constellation schema in Figure 10.3.
Figure E.3. The Fact Tables for Operational Coverages of Three Interventions ITNs, IRS, and ACT with Conformed Dimensions Location and Date. All facts of the fact tables are shown. The fact tables appear in the constellation schema in Figure 10.3.
### PacRain Site Dimension
- PacRain Site Key (PK)
- Site Identifier

### Georeference Dimension
- Georeference Key (PK)
- Longitude
- Latitude

### PacRain Site Data Source Dimension
- PacRain Site Data Source Key (PK)
- Data Source

### Beginning of Observation Date Dimension
- Date Key (PK)
- Beginning of observation date
  **alias for:** Date

### Last Modification Date Dimension
- Date Key (PK)
- Last modification date
  **alias for:** Date

---

**Site Identifier:** uniquely identifies a site; the first two characters indicate the data source for the site:
- **FR** French Polynesian Meteorological Service
- **NZ** National Institute for Water and Atmospheric Research, New Zealand
- **SP** Schools of the Pacific Rainfall Climate Experiment
- **TA** Taylor’s An Atlas of Pacific Islands Rainfall
- **US** National Climatic Data Center, United States

---

**Site Terrain Type:**
- 0 = atoll; 1 = coastal; 2 = coastal with orographic influence; 3 = inland; 4 = inland with orographic influence; 5 = unspecified
- Site Elevation
- Group
- Site Name
- Site Flags
- Site Time Zone

Figure E.4. Detailed Dimensions *PacRain Site*, *PacRain Site Data Source*, *Georeference*, *Beginning of Observation Date*, and *Last Modification Date*. The last two dimensions are aliases for dimension *Date*. Some additional notes are also listed. Most of these dimensions are attached to fact table *PacRain Rainfall*, as shown in Figures E.5 and I0.3.
Figure E.5. The Fact Table *PacRain Rainfall* with Dimensions *PacRain Site*, *PacRain Site Data Source*, *Beginning of Observation Date*, and *Last Modification Date*. All facts of the fact table are shown. The fact table appears in the constellation schema in Figure 10.3.

Figure E.6. Detailed Dimensions *Location*, *Land Use*, *Time Interval*, *Georeference*, *Georeference Source*, *Seasonality Meaning*, *Sporozoite Index Method*, *Biting Rate Method*, and *Citation*. These dimensions are attached to the fact table *Georeferenced EIR*, as shown in Figures E.8 and 10.3. The *Time Interval* dimension is modeled using the *Date* dimension, as shown in Figure E.7. Some additional notes are also listed.
Figure E.7. Modeling Time Interval by Using Date. The Time Interval dimension, which is attached to the fact table Georeferenced EIR (see Figures E.6 and E.8), is modeled using the Date dimension.

Figure E.8. The Fact Table Georeferenced EIR with Dimensions Location, Land Use, Time Interval, Georeference, Georeference Source, Seasonality Meaning, Sporozoite Index Method, Biting Rate Method, and Citation. All facts of the fact table are shown. The fact table appears in the constellation schema in Figure 10.3.
<table>
<thead>
<tr>
<th>GSOD Station Dimension</th>
<th>Date Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSOD Station Key (PK)</td>
<td>Date Key (PK)</td>
</tr>
<tr>
<td>Station Number</td>
<td>Month</td>
</tr>
<tr>
<td>WBAN Number</td>
<td>Day</td>
</tr>
<tr>
<td>WBAN: Weather Bureau</td>
<td>Year</td>
</tr>
<tr>
<td>Air Force</td>
<td></td>
</tr>
<tr>
<td>Navy</td>
<td></td>
</tr>
</tbody>
</table>

**Indicators**: broken into six attributes:
- 1 = yes, 0 = no/not reported
- Fog (‘F’ - 1st digit)
- Rain or Drizzle (‘R’ - 2nd digit).
- Snow or Ice Pellets (‘S’ - 3rd digit).
- Hail (‘H’ - 4th digit)
- Thunder (‘T’ - 5th digit)
- Tornado or Funnel Cloud (‘T’ - 6th digit)

Figure E.9. Detailed Dimensions *GSOD Station* and *Date*. These dimensions are attached to the fact table *GSOD Weather*, as shown in Figures [E.10](#) and [10.3](#). Some additional notes are also listed.
Figure E.10. The Fact Table *GSOD Weather* with Dimensions *GSOD Station* and Date. All facts of the fact table are shown. The fact table appears in the constellation schema in Figure 10.3.
Figure E.11. Detailed Dimensions Location, Intervention, Active Ingredient, Chemical Class, and Manufacturer. These dimensions are attached to the fact table Insecticides, as shown in Figures E.12 and 10.3.

Figure E.12. The Fact Table Insecticides with Dimensions Location, Intervention, Active Ingredient, Chemical Class, and Manufacturer. The single fact of the fact table is shown. The fact table appears in the constellation schema in Figure 10.3.
Figure E.13. The Lookup Table *Species* is Associated with Dimension *Location*, and Dictionary *Species Parameters*. All facts of the lookup table and entries of the dictionary are shown. The lookup table appears in the constellation schema in Figure 10.4.

Figure E.14. The Lookup Table *Entomological Endpoints* is Associated with Dimension *Intervention*, and Dictionary *Entomological Endpoints Phases*. All facts of the lookup table and entries of the dictionary are shown. The lookup table appears in the constellation schema in Figure 10.4.
In this appendix, we present a digital library portal designed earlier as a prototype for the Vector Ecology and Control Network (VECNet) project [241]. We emphasize that this work represents an early prototype that we developed to explore several digital library features using an open-source software named *Greenstone* [82]. While Greenstone offered quick development accompanied by an agile learning curve, it lacked the benefits of detailed customization support and several other dynamic features (e.g., implementing searches based on complex search criteria). The current *VECNet Digital Library Metadata Catalog* [244], being developed as a general repository providing storage, access, and advanced search for malaria-related data in many forms, uses the open-source digital asset management software named *Fedora Commons*, which provides a core data repository exposed as web-based services [64].

In recent years, digital libraries have been extensively used for management and access of digital content, internal information assets, scholarly communications, e-journals, e-prints, e-books, data sets, etc. The collections span a wide range of knowledge domains, which includes cultural, heritage, historical and special collections, museums, biodiversity, e-governance, government policies, plans, procedures, rules and regulations, archiving and preservation, etc.

Greenstone is a suite of software for building and distributing digital library collections [82]. It is produced by the New Zealand Digital Library Project at the University of Waikato [234], and is developed and distributed in cooperation with UNESCO [239] and the Human Info NGO [104]. It provides a convenient way of orga-
nizing information and publishing it on the Internet in the form of a fully-searchable, metadata-driven digital library.

A listing of digital collections, built using Greenstone, can be found in [63]. The collections range from humanitarian information to computer science technical reports, demonstration collections of Chinese and Arabic documents, etc.

We built three sample collections in our digital library portal:

- the VECNet Digital Library Demo collection,
- the Garki Project Demo collection, and
- the WHOPES Demo (the WHO Pesticide Evaluation Scheme) collection

In each collection, we cataloged and stored sample representative documents. Greenstone’s Librarian Interface helped to create, modify, add, or delete collections. The Administration Module enabled maintenance and administration services for authorized library administrators. It also allowed to add new users, summarized the collections in the system, and provided technical information on the installation, etc.

The following figures depict some screenshots from the digital library.
Figure F.1. The Greenstone Librarian Interface (GLI). GLI is an interactive Java program. It helps in building collections, and includes a metadata editor, which lets the digital librarian to create digital content, gather similar content, examine extracted metadata, set up shortcut in the Librarian interface, etc.
Figure F.2. The Format Tab in the Greenstone Librarian Interface (GLI). It allows the digital librarian to modify metadata of specific digital contents.
Figure F.3. Homepage for the Prototype Digital Library Portal. It shows the three sample collections: the VecNet DL collection, the Garki DL collection, and the WHOPES DL (the WHO Pesticide Evaluation Scheme) collection. While the early prototype digital library offered quick and agile development, it lacked the benefits of detailed customization support and several other dynamic features.
Figure F.4. Search with Keyword Malaria in the Prototype Digital Library Portal. A search in the VECNet DL collection with the keyword “Malaria” retrieved 10 documents.
DIVIDE AND CONQUER: A FOUR-FOLD DOCKING EXPERIENCE OF AGENT-BASED MODELS

S. M. Niau Ariffin
Gregory J. Davis
Steve Kurtz
James E. Gentile
Ying Zhou
Gregory R. Madey

Department of Computer Science & Engineering
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA

ABSTRACT

Verification and validation (V&V) techniques are used in agent-based modeling (ABM) to determine whether the model is an accurate representation of the real system. Docking is a form of V&V that tries to align multiple simulation models. In a previous paper, we described the docking process of an ABM that simulates the life cycle of *Anopheles gambiae*. Results showed that the implementations were docked for adult but not for aquatic mosquito populations. In this paper, following the ‘Divide and Conquer’ paradigm, we compartmentalize the simulation world to prohibit the propagation of errors between compartments. Using four separate implementations that sprang from the same core model, we describe a series of docking experiments, analyze the results, and show how they lead to a successful dock. The complete four-fold docking encompasses verification between the four implementations, as well as validation against the core model with respect to these implementations.

1 INTRODUCTION

Figure F.5. A Sample Document Retrieved from the VECNet DL Collection in the Prototype Digital Library Portal. While the early prototype digital library offered quick and agile development, it lacked the benefits of detailed customization support and several other dynamic features.
Figure F.6. Search with Keyword Garki in the Prototype Digital Library Portal. A search in the Garki DL collection with the keyword “Garki” retrieved 17 documents.
Figure F.7. Search with Keyword WHOPES in the Prototype Digital Library Portal. A search in the WHOPES DL collection with the keyword “WHOPES” retrieved 10 documents.
Figure F.8. A Sample Document Retrieved from the *WHOPES DL* Collection in the Prototype Digital Library Portal. While the early prototype digital library offered quick and agile development, it lacked the benefits of detailed customization support and several other dynamic features.
APPENDIX G

P-SAM: A POST-SIMULATION ANALYSIS MODULE

G.1 Introduction

Agent-based models (ABMs) can produce large volumes of textual output, potentially in the range of hundreds of gigabytes. In most cases, these output contain inherent logical structures that can be naturally expressed in terms of abstract mathematical notions such as graphs, relations etc. Moreover, the simulation user is often interested in visualizing these structures in the forms of data plots, time-series analysis, visual graphs and the like. To understand the simulation results and patterns exhibited by the agents, it is crucial to be able to effectively analyze this voluminous textual output, and to produce the desired visualization with ease. Appropriate analysis and visualization also play important roles in verification and validation (V&V) of ABMs. However, simulation modelers often invest the majority of their resources (time and money) on model development and programming, with little attention paid to analysis of simulation output data.

We develop a software module, called P-SAM (Post-Simulation Analysis Module), to analyze and visualize the post-simulation output for ABMs, with special emphasis on biological simulation models. P-SAM is written in the Perl programming language.

\[1\] We would like to thank Paul Brenner, Associate Director, High Performance Computing Group of the Center for Research Computing (CRC) at the University of Notre Dame, for his support to improve P-SAM performance.

\[2\] Portions of this appendix appeared in Arifin et al. \[7\]; results have been reported in Kennedy et al. \[117\] and Lane-deGraaf et al. \[130\].
P-SAM differs from conventional statistical computation software by emphasizing the visualization part that arises from the interaction between abstract entities present in the textual output, with the goal to automate post-simulation analysis tasks for ABMs.

As a case study, this appendix describes the application of P-SAM to a biological simulation model named LiNK that analyzes the spread of disease among macaque monkeys in the Indonesian island of Bali. Reported results [117, 130] indicate the importance of using P-SAM to perform V&V of the LiNK model by allowing internal validity checking and tracing the model entities.

The major strengths of P-SAM lie in allowing visualization of certain network structures, and saving those structures for further analysis. For example, a pathogen transmission graph (Section G.4.4) allows the user to visually track the transmission record of a pathogen. This is particularly helpful with respect to the validation and interpretation of simulation output.

Analyzing large volumes of textual output poses the challenges of huge memory and runtime requirements. To alleviate these, P-SAM uses the technique of serialization (Section G.3.3). It builds the visualization structures as soon as the output data is available and then writes them to files in specific formats (e.g., the dot format for visual graphs). This allows the user to load the structures from files in minimal amount of time and hereby increases the response time.

We envision P-SAM to be useful to other types of ABMs that produce large volumes of textual output. These may include different types of agents (e.g., humans, mosquitoes, monkeys etc.) present in biological models that simulate, for example, the spread of diseases. P-SAM is specially suited for these types of simulations since it can be used to identify the exact chains of transmission, to check whether any transmission link exists between multiple agents, and so on.

The organization of this appendix is as follows: Section G.2 discusses some of the
previous works involving simulation output analysis, describing statistical analysis and other visualization and analysis tools in general. Section G.3 briefly discusses the LiNK Model, followed by the P-SAM architecture. Section G.4 describes some of the analysis and visualization results performed by P-SAM on LiNK, and Section G.5 briefly analyzes P-SAM performance with respect to LiNK. Finally, Section G.6 discusses future directions and Section G.7 concludes.

G.2 Literature Review

In this section, we mention some of the previous works involving simulation output analysis. We begin with statistical analysis and describe its limitations when used as the single tool of analysis. We then point to literature describing historical reasons for inadequate and inappropriate cases of output data analyses. Some recent works on visualization and analysis tools are also discussed.

G.2.1 Statistical Analysis

There are well-known statistical software tools that focus on conventional statistical analysis. These can be used to analyze, and visualize (to some extent) the simulation output.

Simulink [215] is an environment for multidomain simulation and model-based design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries to design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing. R [192] is a popular language and environment for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques that include linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, etc.

SPSS (Statistical Package for the Social Sciences) [222] is among the most widely
used programs for statistical analysis in social science. It is used by market researchers, health researchers, survey companies, government, education researchers, marketing organizations, and others. The base software includes descriptive statistics (cross tabulation, frequencies, ratio statistics etc.), bivariate statistics (means, t-test, ANOVA, correlation), nonparametric tests, prediction for numerical outcomes (linear regression) and prediction for identifying groups (factor analysis, cluster analysis), etc. STATPerl [224] is a free statistical software based on Perl, packaged with source codes for various statistical analysis and an inbuilt Perl interface. Users can add new analysis and edit existing ones.

However, as noted in [132], simulation studies that rely solely on a few statistical estimates for output analysis run the risk of making erroneous inferences about the system, because the statistical estimates, based on random variables, may have large variances. Law [131] described several historical reasons of why output data analyses have not been conducted in an appropriate manner and presented a survey of statistical analyses for simulation output data of a single simulated system, concluding with a discussion of how developments in simulation languages, computer graphics, and computer execution speed may affect the future of output analyses.

With regard to output analysis, Law [132] categorized simulations as *terminating* (a simulation for which there is a natural event \( E \) that specifies the length of each run) and *non-terminating* (a simulation with no such natural event; also called *steady-state* simulation), and discussed statistical analysis methods for both categories. He concluded by listing three pitfalls in output analysis: analysis involving formulae-based runs that assume independence and hence might result in gross underestimation of variances and standard deviations, failure to have a warm-up period for non-terminating simulation analysis, and failure to determine the statistical precision of output statistics by the use of a confidence interval.

Law [133] presented techniques for building valid and credible simulation models,
and discussed the difficulty in using formal statistical techniques to validate simulation models. In particular, he described the limitation of classical statistical tests to conduct a specific form of validation, known as results validation, which compares the model and system output data with those from the corresponding real-world system. Since the output processes of real-world systems and simulations are nonstationary (the distributions of the successive observations change over time) and auto-correlated (the observations in the process are correlated with each other), classical statistical tests based on independent, identically distributed (IID) observations are not directly applicable [131, 133].

Goldsman [79] emphasized simulation output analysis to be one of the most important aspects of any simulation study, pointing out to some of the issues and techniques relevant to conducting valid analysis. Seila [209] reviewed some methods for analyzing data produced by simulations for estimating parameters of stationary output processes. The techniques include some variations of the batch means method, sequential methods, standardized time series estimators, methods based upon Hoeffding’s inequality, quantile estimation, and multivariate estimation methods.

G.2.2 Visualization and Analysis Tools

Bell and O’Keefe [21] showed the usefulness of VIS (Visual Interactive Simulation), which is a simulation methodology for experimental analysis that allows users to suspend the execution of the simulation model, change one or more parameters and resume model execution with the help of a graphic display. To examine the effectiveness of VIS to model experimentation, they performed a task-based behavioral experiment with fifty one subjects, provided with VIS, to solve a case study based around the allocation of trucks in a mining operation. Results showed that the subjects performed worse relative to a known solution obtained through detailed formal experimentation, but performed well compared to solutions they provided prior to
use of the model, and the use of animated display was not associated with correct solutions but was associated with more efficient use of the VIS. They suggested the need for improvement in the design of interaction within the VIS software.

Kurkowski et al. [129] discussed an open-source (C++ and OpenGL-based) visualization and analysis tool, named iNSpect (interactive NS-2 protocol and environment confirmation tool), for use with NS-2 (the Network Simulator 2) wireless simulations. They emphasized the need of a visualization tool to understand the large amount of data produced during network simulations. Describing the default (existing) tool called NAM (Network Animator), they showed how its limitations necessitated iNSpect for wireless simulations, and how iNSpect handled three areas of NS-2 based simulation research: (1) validation of the mobility model’s output and the node topology, (2) validation of new versions of the NS-2 simulator itself, and (3) statistical and visual analysis of the results of NS-2 simulations.

The Stanford Microarray Database (SMD) [50] provides a wide array of web-accessible tools for processing, analyzing, visualizing and sharing microarray data in a research environment. The users, having access to data from thousands of microarrays, need effective tools to locate microarrays of interest. To accomplish this, SMD provides two different search forms (basic and advanced) to find relevant data. The users are presented with a variety of flexible options which allow them to find data based on specific researcher who entered the microarray data, keywords, text searches of experiment descriptions or pre-grouped selections of microarrays.

Gollub et al. [80] described how SMD serves as a resource for the entire scientific community, by making its source code open and providing full public access to data published by SMD users, along with many tools to explore and analyze those data. They discussed, for example, a data visualization tool, named Array Color, that provides a simplified view of the ratio data for a given microarray, allowing the user to quickly examine the microarray. In addition to the graphical display, the
Array Color tool provides simple analysis of variance (ANOVA) calculations. Hubble et al. [101] described their implementation of the GenePattern microarray analysis software package into the SMD code base that provides access to many new analysis tools and allows users to directly integrate and share additional tools through SMD.

The OpenScience project [235] hosts numerous open-source software packages with simulation analysis and visualization tools to be used in versatile scientific domains. These include Packmol for molecular dynamics (MD) simulations, AGM Build for interactive model preparation for molecular dynamics simulations, Scilab for numerical computations and relevant simulation, OpenFOAM (Field Operation and Manipulation) etc. A list of the popular projects can be found at [235].

Simbios, the NIH Center for physics-based simulation of biological structures [212], provides infrastructure and software to help biomedical researchers. SimTK, the Simbios biosimulation toolkit [214], provides a collection of technologies to build Simbios applications for a wide variety of domains, ranging from molecules to whole organisms. The major biological projects include RNA folding, protein folding, myosin dynamics, neuromuscular dynamics, and cardiovascular dynamics.

G.3 Design and Implementation

P-SAM was initially developed for a GIS-aware agent-based model named LiNK and named LiNKStat (Statistics builder for LiNK). In this section, we briefly discuss LiNK, followed by the P-SAM architecture.

G.3.1 The LiNK Model

LiNK is an interdisciplinary project which models pathogen transmission amongst long-tailed macaque monkeys on Bali, Indonesia. Macaques on Bali exist in distinct populations: within temple sites and dispersing (roaming). LiNK models each population separately while offering the capability for appropriate macaques to move
between populations and within varying landscapes. LiNK aims to address specific research questions such as the potential rates and routes of pathogen transmission in macaques across the island and the impact of the pathogen life history parameters on this transmission, paying careful attention to the role of landscape on pathogen transmission. For details, see [117].

The model consists of a display of Bali with temple sites and macaques, along with a display of the contents of the temples. LiNK can simulate a wide array of pathogens through various pathogen parameters, such as infectivity, virulence, latency, etc. Macaques are simulated as agents, with each macaque having its own properties (e.g., location, sex, age, etc.). They can also move through their environment, interact with other macaques, transmit pathogens, reproduce, and die.

G.3.2 P-SAM Architecture

The core P-SAM architecture (see Figure G.1) consists of two programs called the writer and the reader. The writer first serializes the simulation output; then, the reader builds the interactive visualization structures. Both of these are described below.
Figure G.1. The P-SAM Architecture.
TABLE G.1

PERL EXTENSION MODULES USED IN P-SAM

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tk/Tcl</td>
<td>To provide access to the Tk library, and build the GUI.</td>
</tr>
<tr>
<td>Graph</td>
<td>To create abstract graph data structures, and use the graph.</td>
</tr>
<tr>
<td>GraphViz</td>
<td>To visualize the graphs generated by Graph.</td>
</tr>
<tr>
<td>Devel::Size</td>
<td>To report the memory usage of the graphs and the GUI.</td>
</tr>
<tr>
<td>Devel::NYTProf</td>
<td>To profile the source code (both statement profiling and subroutine profiling).</td>
</tr>
<tr>
<td>Devel::Profile</td>
<td>To profile the source code (basic profiling).</td>
</tr>
</tbody>
</table>
G.3.3 The Writer

The writer analyzes and serializes the simulation output. Serialization, also known as deflating or marshalling, is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file, a memory buffer, or transmitted across a network connection link to be rebuilt later in the same or another environment. When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object. The opposite operation, i.e. extracting a data structure from a series of bytes, is called deserialization.

Several available Perl modules provide serialization mechanisms (e.g., FreezeThaw, Data::Dumper, Storable etc.). However, for LiNK, we chose to use the Graph module due to the following reasons:

- the pathogen transmission pattern of LiNK can be most naturally expressed with the notion of a (mathematical) graph structure
- Graph allows the recorded information to be easily saved and later retrieved for visualization

The writer takes the LiNK file as its input (produced as output from the LiNK simulation) and serializes it (after analysis) into three separate files: Infection Data, Roaming Infection Data, and Summary Data. The first two are written in the DOT format, which is a plain-text graph-description language to describe graphs usable by both humans and computer programs. DOT allows hierarchical or layered drawings of directed graphs. The last one records the summary information about the LiNK file and the P-SAM processing.

G.3.4 The Reader

Once analysis and serialization are complete, the reader allows visualization by building the Graphical User Interface (GUI). It reads in the aforementioned serialized
DOT files, and projects the information into the *Infection Statistics* and *Roaming Infection Statistics* tabs for *Infection Data* and *Roaming Infection Data*, respectively (see Section G.4).

### G.3.5 Advantages of Using Perl

Both the *writer* and the *reader* are written in Perl [237], and hence offer the following advantages:

- **Portability:** Perl does not use system-specific features and maintains the portability of an interpreted language while achieving nearly the speed of a compiled language. Thus, P-SAM can be run on any platform.

- **String processing and Regular Expression Support:** The major processing task of P-SAM is to handle large volumes of textual output from the LiNK simulation. This is the most compelling reason of choosing Perl, which has highly versatile regular expression support, seamlessly integrated into the language.

- **Reusable Code repository:** P-SAM uses several Perl extension modules (see Table G.1) from CPAN [45], which has a huge collection of free and reusable Perl code. Use of these modules ensures tested error-free code, avoiding the need to reinvent the wheel.

### G.4 Post-Simulation Analysis and Visualization for LiNK

P-SAM works on relations defined by the user. The user identifies different relations involving relevant entities (e.g., agents) in the simulation output file and specifies how much satellite data should accompany each relation. P-SAM then builds the visualization structures separately for each type of relations and outputs those in the form of visual graphs, data plots etc. Also, structures that require more time and memory to be built can be saved in different formats (e.g., image files, relational database records, etc.).

The visualization process enables the user to visually analyze the following features built from the simulation output. LiNK uses a unique naming convention for
each macaque with natal temple number concatenated with an id concatenated with a gender identifier.

G.4.1 Infection Statistics

An infection event occurs when a macaque transmits the pathogen to another macaque inside a temple, allowing self-transmission. The Infection Statistics tab (see Figure G.2) shows all the initially infected macaques (top-left) and the macaques that did not infect any other macaque (top-right) in the course of the whole simulation. It lists all macaques that took part in infection events (the leftmost vertical list), and populates the bottom-left frame with details of all infection events, listing the time-step, infected macaque and the temple where each event took place. To allow interactive probing, when the user clicks on an individual macaque on the vertical list, it dynamically populates the bottom-right frame with details of all infection events for the macaque.

G.4.2 Roaming Infection Statistics

A roaming infection event occurs when a macaque transmits the pathogen to another macaque outside the temples, allowing self-transmission. Each roaming infection event is accompanied by location information, such as the latitude and longitude, as well as the landscape in which the infection took place, such as in City, Forest, Rice Field, River, Road, and Coast.

The Roaming Infection Statistics tab (see Figure G.3) is similar to the Infection Statistics tab, except it does not contain any initially infected macaques (as roaming). It lists all macaques that took part in roaming infection events (the leftmost vertical list), and populates the top frame with details of all roaming infection events, listing the time-step, infected macaque, temple, latitude, longitude and landscape. When the user clicks on an individual macaque on the vertical list, it dynamically populates
the bottom frame with details of all roaming infection events for the macaque.

G.4.3 Birth and Death Statistics

In a birth event, a mother macaque gives birth to an infant. A death event records the time, place and cause of deaths. LiNK simulates three different causes of deaths: aging, dispersal and pathogen.

The Birth and Death Statistics tab (see Figure G.4) lists all birth events with the corresponding time-steps (left), and all death events with the corresponding time-steps, causes of deaths and the location temple (right).

G.4.4 Pathogen Transmission Graphs

The major strengths of P-SAM lie in allowing visualization of certain network structures, and saving those structures for further analysis (e.g., by direct printing). A pathogen transmission graph is an example of one such network structure that allows...
Figure G.3. P-SAM Roaming Infection Statistics Tab.
Figure G.4. P-SAM Birth and Death Statistics Tab.
the user to visually track the transmission record of pathogen. This is particularly helpful with respect to the validation and interpretation of simulation output.

Figure [G.5] shows an example of a pathogen transmission graph with nodes representing macaques and edges representing infection events. The topmost node (27.2969.0) is parsed as a female macaque with temple 27 as its natal temple and 2969 as its id. Infection events are listed with the time-step and location where the infection occurred. Starting at the top, macaque 27.2969.0 infected macaque 27.2775.0 at time-step 1, in temple 27. Macaque 27.2775.0 went on to infect three other macaques (infecting macaque 27.2753.0 twice at time-steps 11 and 12), and was also reinfected by macaque 27.2870.1. Autoinfection (reinfecting oneself) is possible, as indicated by macaques 27.2863.0 and 27.2805.1.

G.4.5 Summary Statistics

The Summary Statistics tab (see Figure [G.6]) lists summary information about the LiNK file and the P-SAM processing. The Input/Output Statistics frame (top-left) lists several original parameters from the LiNK file, including acquired immunity, clearance time, virulence, infectivity, random seed, initially infected temples etc.

The Line count and Event Statistics frame (top-right) lists various useful information obtained after P-SAM finishes processing the LiNK file. These include the counts for initially infected macaques, infections, roaming infections, the total number of infected macaques (separately at temples and as roaming), and the total number of infected males and females. It also summarizes the total number of death events, categorized by the cause of death with separate counts.

The Temple Statistics frame (bottom-left) summarizes the number of unique infections occurring at each temple. The Runtime and Memory Statistics frame (bottom-right) reports the memory and runtime consumed by P-SAM.
Figure G.5. Example of a Pathogen Transmission Graph. Nodes represent macaques, and edges represent infection events. Loops represent auto-infection.
Figure G.6. P-SAM Summary Statistics Tab.
G.5 P-SAM Performance

LiNK output files can reach sizes in the tens or hundreds of gigabytes, with terabytes of uncompressed output data generated so far. This poses P-SAM with challenges of large memory and runtime requirements. Using serialization, P-SAM alleviates some of these issues by building the visualization structures as soon as the data is available, and then writing them to files for later retrieval. This allows the user to load the structures from files in minimal time, and thereby increases the response time.

P-SAM runtime is dominated by the number of infection events and their degree of proliferation in the original LiNK simulation. In collaboration with CRC [36] at the University of Notre Dame, we achieved some performance improvements, by working iteratively over three major phases. As evident from Figure [G.7] the run-time scales best (with respect to input file-sizes) in Phase 3. This was done primarily by profiling and code optimization, which are described below.

G.5.1 Profiling

Software profiling is a form of dynamic program analysis that investigates a program’s behavior using information gathered as the program executes with the purpose of determining which sections of a program to optimize (i.e. to increase its overall speed, decrease its memory requirement, etc.). A profiler is a performance analysis tool that measures the frequency and duration of function calls and collects extensive performance data.

We use two different profilers, Devel::Profile and Devel::NYTProf, both available at CPAN [45]. The use of two profilers enables us to compare the profiles and identify hotspots with increased confidence. Devel:: Profile is a simple profiler that collects information about the execution time and about the subroutines in the code. This information can be used to determine the duration and calling frequency of the
Figure G.7. P-SAM Performance. For each phase, the seven data points represent average values of output files, obtained by varying parameter settings of the LiNK model. The same seven sets of parameters are used for all phases. Phase 1 bypasses the GUI. Phase 2 eliminates unnecessary existence checks for infection hash used in the Writer, thus reducing search penalties for macaque nodes in the hash. Finally, Phase 3 optimizes some hash operations by using sets.
subroutines. Devel::NYTProf is a powerful feature-rich profiler, and is the fastest reported so far. It offers profiling with finer granularity, i.e. per-line statement profiling, per-subroutine statement profiling, per-opcode profiling, and per-block statement profiling. It can perform inclusive and exclusive timing of subroutines with sub-microsecond resolution and generate richly annotated, cross-linked html reports.

G.5.2 Code Optimization

Profiling is a useful tool for optimizing code, but it can only point to the potential location of the problem and not actually resolve those. Since most programs spend most of the execution time in a few small hotspots, after the hotspots are identified by profiling, certain optimization techniques can help improve the performance.

For Perl, the combination of logic sequence and generated bytecode may have drastic effect on performance. To partially minimize this, several checks and practices may be helpful [30]. These include using references for large arrays and hashes, avoiding excessive function calls in loops, using short circuit logic, etc.

After analyzing the profiles, we eliminate unnecessary existence checks for infection and roaming infection hashes used in the Writer. This greatly reduces the search penalties for macaque nodes in the graphs (which are stored as hashes). We also replace some hash operations by using sets, which provide faster subroutines for operations like set union, set intersection, etc.

G.6 Future Work

We plan to modify and enhance P-SAM to be able to perform the following:

**Multi-run Analysis** Automaton of simultaneous analysis of multiple data-sets for parallel comparison and complex processing. Some initial steps have already been done during performance improvement (Section G.5).

**Generalization** Decoupling from specific domain of application in order to apply P-SAM in a broader context.
To achieve generalization, we plan to investigate the most commonly-used (or, desired) visualization structures for other biological simulations which produce large volumes of textual output. For example, in addition to building the routes of pathogen transmission for macaque agents, P-SAM can be extended for the malaria epidemiology model with mosquito agents for malaria transmission (e.g., see [6]). This may be helpful, for example, to answer the following questions:

- To find all mosquito agents, across their heterogeneous populations, in specific states in the mosquito life-cycle, with interaction relationships to other mosquito agents (e.g., mating between a pair of male-female)
- To find parent-child relationships from a gravid female mosquito to its eggs
- To categorize agents by generations, age-cohorts, age-specific mortality rates, geographic locations, aquatic habitats, etc.
- To perform the above, where relevant, for other types of agents (e.g., humans)

Some of the following may help to achieve these goals:

- **Preprocessing**: In order to categorize the events in the input file, P-SAM can decompose a large file into several smaller files according to similar events, sort the smaller files, and then perform the analysis with the type of event known a priori.

- **Data Mining** and **Causal Data Analysis**: these areas may prove useful in discovering high-level macro patterns (besides summary statistics) from the simulation output.

- **Relational Database (RDBMS)**: For fast future retrieval, structures (that require longer runtime and memory to be built) can be stored in an RDBMS as database records.

G.7 Conclusion

In this appendix, we described a software module, called P-SAM, developed for simulation output analysis and visualization. We demonstrated its application to a biological simulation model named LiNK [139] that produces voluminous textual outputs (in the range of hundreds of gigabytes). We showed how P-SAM can analyze
and visualize logical structures (e.g., pathogen transmission graphs) inherent in LiNK outputs, thus helping V&V of the model by allowing internal validity checking, and tracing the model entities. We also described the P-SAM architecture, its major strengths, and how it copes with the huge memory and runtime requirements. We concluded with future directions to generalize P-SAM for other biological simulations.
BIBLIOGRAPHY


193. R Farlow Consulting LLC. 156 Cardinal Cove, Burkeville, TX 75932, USA.


