The Exploratory Advanced Research Program

A Primer for Agent-Based Simulation and Modeling in Transportation Applications

EXPLORATORY ADVANCED RESEARCH

U.S. Department of Transportation
Federal Highway Administration

Foreword

Agent-based modeling and simulation (ABMS) methods have been applied in a spectrum of research domains. This primer focuses on ABMS in the transportation interdisciplinary domain, describes the basic concepts of ABMS and the recent progress of ABMS in transportation areas, and elaborates on the scope and key characteristics of past agent-based transportation models, based on research results that have been reported in the literature. Specifically, the objectives of this primer are to explain the basic concept of ABMS and various ABMS methodologies scoped in the literature, review ABMS applications emerging in transportation studies in the last few decades, describe the general ABMS modeling frameworks and commonly shared procedures exhibited in a variety of transportation applications, outline the strength and limitation of ABMS in various transportation applications, and demonstrate that ABMS exhibits certain comparable modeling outcomes compared to classical approaches through a traveler's route choice decisionmaking process example.

The target audiences of this primer are researchers and practitioners in the interdisciplinary fields of transportation, who are specialized or interested in social science models, behavioral models, activity-based travel demand models, lane use models, route choice models, human factors, and artificial intelligence with applications in transportation.

Monique R. Evans
Director, *Office of Safety Research and Development*

Debra S. Elston
Director, Office of Corporate Research, Technology, and Innovation Management

Notice

Quality Assurance Statement

Technical Report Documentation Page

1. Report No. FHWA-HRT-13-054	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A Primer for Agent-Based Simulation and Modeling in Transportation Applications		5. Report Date November 2013
		6. Performing Organization Code:
7. Author(s) Hong Zheng, Young-Jun Son, Yi-Chang Chiu, Larry Head, Yiheng Feng, Hui Xi, Sojung Kim, Mark Hickman		8. Performing Organization Report No.
9. Performing Organization Name and Address University of Arizona, 1401 E University Blvd, Tucson, AZ 85721		10. Work Unit No.
		11. Contract or Grant No. DTFH61-11-H-00015
12. Sponsoring Agency Name and Address Office of Safety Research and Development and Office of Corporate Research, Technology, and Innovation Management Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296		13. Type of Report and Period Covered Final Report, May 2011-May 2014
		14. Sponsoring Agency Code HRTM-30
15. Supplementary Notes FHWA Agreement Officer's Technical Representative (AOTR): Dr. C. Y. David Yang, Office of Safety R&D		
16. Abstract Agent-based modeling and simulation (ABMS) methods have been applied across a spectrum of domains within transportation studies. Different paradigms for ABMS in transportation exist; in general, ABMS has strong roots in the individual-based travelers' model in the activity-based travel demand domain. In the distributed system domain, ABMS is commonly seen as a method, known as *multiagent systems*, for a distributed autonomous system. Recently, transportation-related applications leveraging ABMS have continued to grow. This report attempts to clarify the concept of ABMS and summarize variant paradigms that have been studied in the transportation field. It will do this by distinguishing similarities or differences of the specified problems, model capabilities, strengths and weaknesses of ABMS scoped in different applications, and through a comprehensive review of ABMS approaches that have been seen in transportation studies. The report also seeks to connect the individual-based ABMS with the transportation problems viewed in the social science paradigm. This is achieved by trying to apply ABMS characterized by social science rules to study behavioral decisions of individual travelers. This exploratory study is demonstrated in an example of travelers' route choice decisions, which features a bottom-up, rather than a conventional top-down, approach to formulate the mechanism of an individual traveler's complex route choice behavioral process as a collaborative and reactive result of the traveler's mindset and the network environment integrated in an ABMS.		
17. Key Words Agent-based modeling and simulation, transportation, route choice, transportation planning.		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 75	22. Price N/A
Form DOT F 1700.7 (8-72)		Reproduction of completed page authorized	

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

Table of Contents

List of Figures

List of Tables

List of Symbols

$>$	Greater than
$\leq$	Less than or equal to
$\neq$	Not equal to
$\propto$	Proportional to
Σ	Summation symbol
Π	Product symbol
$\alpha_n, \alpha_{0n}, \alpha_{1n}, \alpha_{2n}, \alpha_{jn}, \alpha_{in}$	Dirichlet distribution parameter set
B	Beta function
$\mathbf{d}_n$	Vector of minimum travel time variables
d_{jn}	Binary variable, equal to 1 if the traveler perceives that the jth route takes the minimum travel time (TT_{min}^n) on the nth day, and 0 otherwise
e	Natural exponential constant, equal to 2.71828
E	Expectation
ε	Route Choice Threshold
$\in$	Element of
f	Function
f_r	Proportational value of travelers who replan their trips in TRANSIMS
g	Data likelihood function
j	Route
n	Day
θ	Gumbel distribution parameter
p_{in}	Subjective probability that the i^{th} route takes the minimum travel time on the nth day
$\mathbf{p}_n$	Vector of subjective probabilities
S_i	Number of iterations of the outer-loop in DaySim
TT_j^n	Travel time of jth route on nth day
TT_{min}^n	Minimum travel time of all routes on nth day
γ_{jn}	Perception error term

List of Acronyms and Abbreviations

General Terms

2D	two dimensional
3D	three dimensional
ABMS	agent-based modeling and simulation
ABS	agent-based simulation
ABSS	agent-based social simulation
ACT-R	Adaptive Control of Thought—Rational
aDAPTS	Agent-based Dynamic Activity Planning and Travel Scheduling
AI	artificial intelligence
API	application programming interface
BBN	Bayesian belief network
BDI	belief–desire–intention
BPR	Bureau of Public Roads
CARTESIUS	Coordinated Adaptive Real-Time Expert System for Incident Management in Urban Systems
CAS	complex adaptive systems
CEMDAP	Comprehensive Econometric Microsimulator for Daily Activity Travel Patterns
CEMSELTS	Comprehensive Econometric Microsimulator for Socio-economics, Land Use, and Transportation System
CTMRGS	Cooperative Traffic Management and Route Guidance System
DAI	distributed artificial intelligence
DaySim	Person Day Activity and Travel Simulator
DTA	dynamic traffic assignment
DYNASMART-P	Dynamic Network Assignment-Simulation Model for Advanced Roadway Telematics (Planning version)
DynusT	Dynamic Urban Systems for Transportation
EMME	equilibre multimodal, multimodal equilibrium
FIPA	Foundation for Intelligent Physical Agents
GA	genetic algorithm

GMU	George Mason University
HOV	high-occupancy vehicle
HUT	Helsinki University of Technology
ILUTE	Integrated Land Use, Transportation, Environment
ISP	information service provider
MABS	multi-agent based simulation
MALTA	Multiresolution and Loading of Transportation Activities
MAS	multi-agent systems
MASON	Multiagent Simulator of Neighborhoods
MATSim	Multi-Agent Transport Simulation Toolkit
MITSIM	MIcroscopic Traffic SIMulator
NASA	National Aeronautics and Space Administration
NP-hard	Non-deterministic Polynomial-time hard
O-D	origin–destination
OpenAMOS	Open Activity-Mobility Simulator
PCATS	Prism Constrained Activity Travel Simulator
PopGen	population generator
PopSyn	population synthesizer
Repast	Recursive Porous Agent Simulation Toolkit
RL	reinforcement learning
SAAS	social aspects of agent systems
SACSIM	Sacramento Activity-Based Travel Demand Simulation Model
SimAGENT	Simulator of Activities, Greenhouse Emission, Networks, and Travel
SOCSIM	social simulation
SOV	single-occupant vehicle
TASHA	Travel Activity Scheduler for Household Agents
TRACK-R	TRaffic Agent City for Knowledge-Based Recommendation
TRANSIMS	Transportation Analysis and Simulation System
TRYS	Tráfico, Razonamiento y Simulación
TRYSA2	Tráfico, Razonamiento y Simulación Autonomous Agents
UE	user equilibrium

Introduction

Agent-based modeling and simulation (ABMS) has been widely applied across a spectrum of disciplines by both researchers and practitioners. Examples of these disciplines include ecology, biology, business, economic science, computer simulation, social sciences, political science, policy, and military studies. Knowledge and applications of ABMS continue to expand and accumulate through rapid and in-depth research and development.

ABMS has been applied to a broad range of domains in transportation. These applications primarily fall into two methodological paradigms: individual-based models that study personal transportation-related activities and behavior, and computational (or system) methods that study a collaborative and reactive transportation system that exhibits intelligence by modeling a collection of autonomous decisionmaking of subsystem entities called *agents*. The former is closely related to the models for travelers' activities. The latter is typically scoped as a computational method in a distributed artificial intelligence (DAI) system, or a complex adaptive system (CAS), which is a powerful technique for simulating dynamic complex systems to observe emergent behavior. In research literature, it is common to see transportation studies crossing the boundary of the two categories but scoped with the same (or similar) term, *agent-based*, thus leading to conceptual confusion.

The goal of this primer is to review the historical aspects and the ongoing developments of ABMS in the interdisciplinary transportation areas, summarizing and clarifying the scope and key characteristics of past agent-based studies and to shed light on future potential research. Another scientific focus of this primer is to document a research effort that attempts to establish the relationship between the classical and ABMS-based route choice models. This effort aims to answer a scientific inquiry: Because both classical econometric models and ABMS are plausible in modeling individuals' route choice decision behaviors, there supposedly exist certain conditions and contexts at which both modeling paradigms exhibit comparable results. This inquiry, as an important step toward a better understanding of the classical econometric method and ABMS-based approaches, sheds light on the path forward for the development of a holistic modeling framework.

The objectives of this primer are to:

- Explain the basic concept of ABMS and various ABMS methodologies scoped in the literature.
- Review ABMS applications that have emerged in transportation studies in the past few decades.
- Describe the general ABMS modeling frameworks and commonly shared procedures exhibited in a variety of transportation applications.

- Outline the strength and weakness of ABMS in various transportation applications.
- Demonstrate, through a traveler's route choice decisionmaking process example, that ABMS exhibits certain modeling outcomes compared with classical approaches.

In reviewing ABMS applications in transportation, this primer serves to present and summarize the ABMS approaches that have been practiced in the transportation paradigm in the past few decades and depict the concept of ABMS as discussed in the literature. The applications described in this primer represent major recognized transportation systems and platforms that have been leveraging ABMS, rather than comprehensively reviewing products.

This primer is organized by chapters and sections. In chapter 1, fundamental concepts of ABMS are introduced, and both benefits and challenges of this methodology are presented. In chapter 2, the authors discuss how to model learning, and interactions in general, through an agent-based method derived from the individual-based human behavioral perspectives in the social science paradigm. In chapter 3, the authors briefly introduce several agent-based simulation software toolkits. These toolkits have the standardized modules, processes, libraries, and programming language that could be used conveniently to develop an agent-based model. In chapter 4, the authors discuss agent-based behavioral models that have been studied by the transportation community. Most of those models are individual-based models, in which agents are individual travelers. Those models have strong roots in activity-based transportation models. In chapter 5, the authors review agent-based system modeling in transportation problems where agents are intelligent, distributed, and autonomous subsystems. These subsystems (agents) interact with each other and model complex holistic performance and emergent behavior of the overall transportation system. In chapter 6, the authors seek to demonstrate that ABMS could be viable to model the individual traveler's route choice decisionmaking process, when compared with classical methods. In chapter 6, a behavioral model leveraging the social science methodology, called *belief–desire–intention* (BDI), is established in a bottom-up framework. This model attempts to formulate the mechanism of a traveler's complex route choice behavioral process as a collaborative and reactive result of the traveler's mindset and the network environment.

The target audiences of this primer are researchers and practitioners in the interdisciplinary field of transportation, who specialize or have an interest in social science models, behavioral models, activity-based transportation models, lane-use models, human factors, and artificial intelligence (AI) models with applications in transportation.

CHAPTER 1:
Agent-Based Modeling and Simulation

Basic Concepts

ABMS is a modeling approach for simulating the actions and interactions of autonomous individuals, with a view to assessing their effects on the system as a whole. An essential idea of ABMS is that many phenomena, even complex ones, can be understood as systems of autonomous agents that follow rules of interaction.[1] Repetitive, competitive interactions between agents are major features of ABMS, which rely on the power of computers to explore dynamics out of the reach of pure mathematical methods. In a traditional discrete event simulation, entities follow a sequence of processes, which are defined from the top-down system perspective. In contrast, ABMS defines the local behavior rules, usually simple, of each entity from a bottom-up perspective. In accordance, simulation results reveal the emerging behaviors of a system as a whole, based on the behavior formations of the underlying entities. The main roots of ABMS are in modeling human social and organizational behavior and individual decisionmaking.[2]

What Is an Agent?

There is no universal definition of the term *agent*, as agent could refer to different components when studying different objectives in different paradigms. Some may consider any type of distinguished parts of a program (e.g., model, system, or subsystem), or any type of independent entity (e.g., organization, firm, or individual people), to be an agent. As shown in figure 1, the agent is programmed to react to other agents and the computational environment in which it is located,[3] with a behavior rule ranging from primitive reaction decisions to complex adaptive AI.[4]

According to Macal and North,[4] an agent in a typical ABMS could be defined as follows:

- Identifiable, self-contained, and discrete with a set of characteristics and rules governing its behaviors and decisionmaking capability. The discreteness requirement implies that an agent has a boundary, and one can easily determine whether something is part of an agent, is not part of an agent, or is a shared characteristic.
- Situated, living in an environment with which it interacts along with other agents. Agents have protocols for interaction with other agents, such as for communication, and the capability to respond to the environment. Agents have the ability to recognize and distinguish the traits of other agents.
- Goal-directed, having goals to achieve (not necessarily objectives to maximize) with respect to its behaviors. This allows an agent to compare the outcome of its behavior relative to its goals.
- Autonomous and self-directed. An agent can function independently in its environment and in its dealings with other agents, at least over a limited range of situations that are of interest.
- Flexible, having the ability to learn and adapt its behaviors based on experience, which requires some form of memory. An agent may have rules that modify its rules of behavior.

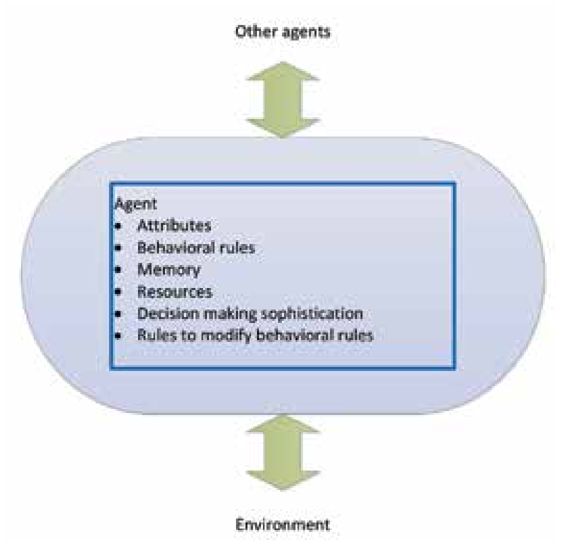


Figure 1. An agent. (**Modified from Macal & North**[4])

Scope of ABMS

Agent-based models consist of agents that interact within an environment. Agent-based modeling has been called by various names in the broad base of its applications, which could refer to completely different methodologies. In a computing scientific domain (e.g., AI or distributed autonomous systems), agent-based modeling typically refers to a computational method and simulation for studying the actions and interactions of a set of autonomous entities. It is also called a *multi-agent system* (MAS) or *agent-based system*. In non-computing–related scientific domains (e.g., ecological science or life science), Agent-Based models usually refer to the individual-based models.

In social sciences, agent-based modeling could refer to an actor in the social world. In recent years, in agent-based social simulation (ABSS) that mimics social phenomena, the concept of autonomous agents has become well-known. Davidsson[5] classifies research areas in ABSS into social aspects of agent systems (SAAS), multi-agent–based simulation (MABS), and social simulation (SocSim). This classification depends upon different combinations of focus areas, which include agent-based computing, computer simulation, and social science. First, SAAS focuses more on social science and agent-based computing and includes the study of norms, institutions, organizations, cooperation, and competition, among others. Second, research in the intersection between computer simulation and agent-based computing is referred to as *MABS* and uses agent technology for simulating any phenomena other than social phenomena. Third, SocSim is in the intersection between social science and computer simulation and corresponds to the simulation of social phenomena on a computer that uses typically simple models of the simulated social entities, such as cellular automata. In transportation research and applications, which are the focus of this primer, the keyword of *agent-based modeling* is mostly seen referring to an individual-based model and simulation or an autonomous computing method.

Although agent-based modeling is a diverse research paradigm applied in completely different ways in a large and widely spread scientific field, all eventually tie together in the domain of agent-based computing.[6] The term *ABMS* in this primer covers all semantics referenced above.

It is worth noting that an agent-based system could also be a software method, such as defined by the Institute of Electrical and Electronics Engineers' Foundation for Intelligent Physical Agents, an international agent standard. The ABMS agents in this primer exclude the software method, in which agents (including mobile agents) are lightweight software proxies that roam over the World Wide Web and perform various functions.[6]

Backgrounds of ABMS

ABMS evolved from AI and computer science but is now being developed independently in research centers throughout the world. The history of ABMS can be traced back to John Von Neumann, who conceived and developed a device later known as *cellular automata*.

In the 1970s, John Conway developed the Game of Life, a two-dimensional (2D) cellular automata shown in figure 2.[7] A cell has two states, *alive* and *dead*; the state of a cell depends on the state of the neighbors of the previous time step. Conways's game engendered great interest in the emergence of complexity from simple rules.

Interest continued to grow and diversify in the 1990s with the appearance of various tools, particularly Swarm and NetLogo in the mid-1990s and Recursive Porous Agent Simulation Toolkit (Repast) and AnyLogic in 2000.

In the mid-1990s, Joshua Epstein and Robert Axtell[9] developed *Sugarscape*, an artificially intelligent ABSS, which captures fundamental concepts of social sciences. At each grid point on a plane, sugar grew at a constant rate. A set of agents, with a fixed, randomly determined level of vision and metabolism, find and eat sugar on the sugarscape. If sugar at one place was exhausted, the agent then moved to a new location where it had the maximum

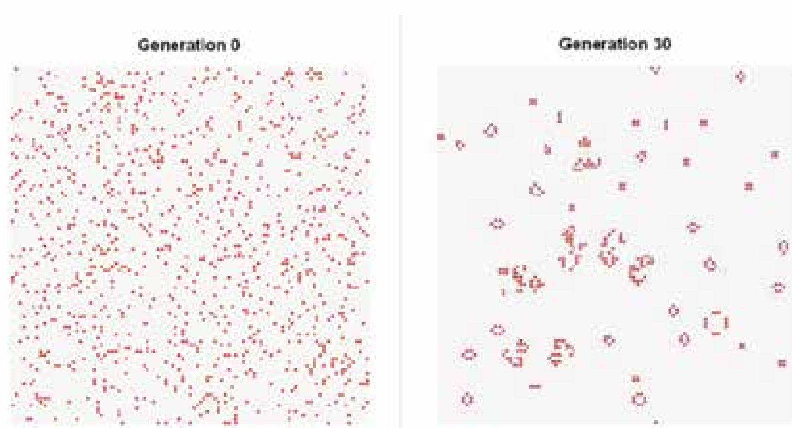


Figure 2. Example of "Game of Life." (**Source: Macal & North**[8])

sugar within its vision. This simple system of rules led to migration phenomenon. More rules created additional interesting results. Epstein and Axtell added spice, a second resource similar to sugar. This showed that with barter economies, agents had a higher chance of survival. They added sex and, when there was sufficient sugar, the agents would reproduce. This led to age pyramids, tribal growth, and other demographic features. Other rules led to combat and other evocative results. Sugarscape showed how simple rules could create a complex society in a bottom-up manner, as shown in figure 3, and inspired further growth in agent-based modeling. In the late 1990s, computer power advanced significantly, and ABMS became widespread.

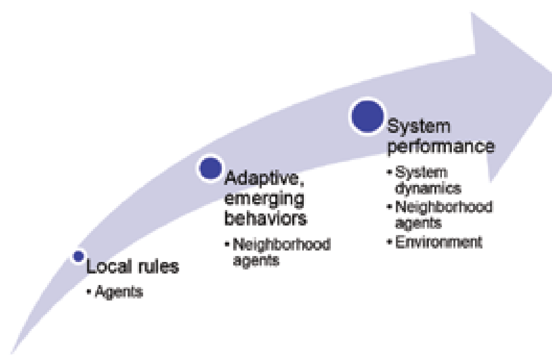


Figure 3. A bottom-up approach to ABMS.

The field of CAS is sometimes referenced as providing the historical roots of ABMS. CAS draws its primary inspiration from biological systems and is concerned mainly with how complex adaptive behavior, as shown in figure 4, emerges in nature from the interaction among autonomous agents.[10] One of the fundamental contributions made to the field of CAS, and to ABMS as well, was John Holland's identification of the four properties—aggregation, nonlinearity, flows, and diversity—and three mechanisms—tagging, internal models, and building blocks—that compose all CAS. Essentially, these items have aided in defining and designing ABMS as they are known today.[10]

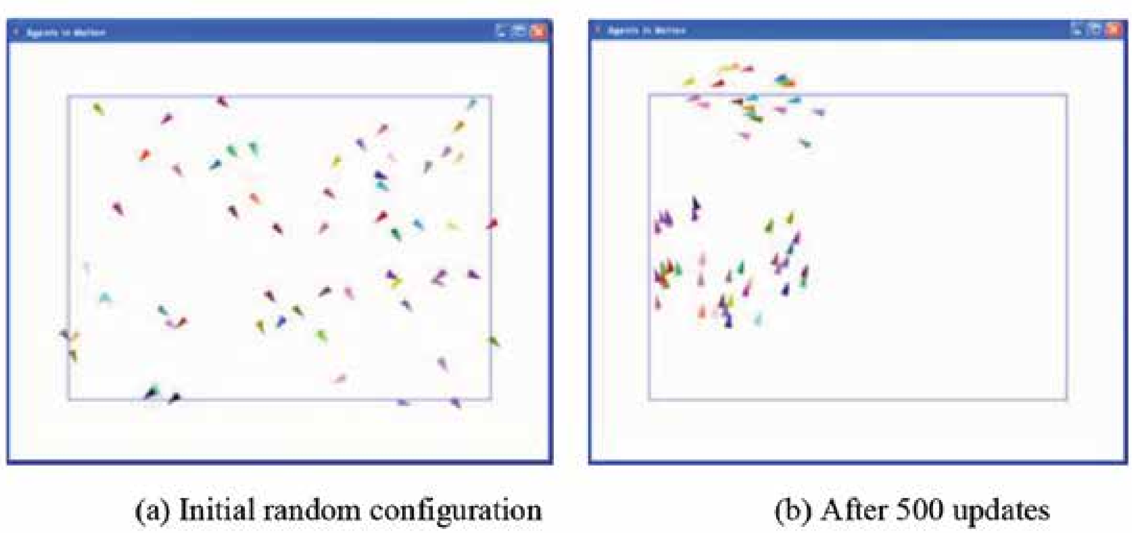


Figure 4. Simple local rules result in complex system behaviors. (**Source: Macal & North**[4])

The Need for ABMS

The ABMS approach allows one to represent and analyze a complex problem (e.g., system dynamics) beyond the reach of mathematics or traditional modeling tools. Advances in database technology (allowing a finer level of granularity) and computational power allow one to compute large-scale microsimulation models that would not have been possible even recently.[11] This feature of ABMS has contributed to the field of computer simulation by providing a new paradigm for the simulation of complex systems with many interactions between the entities of the system.[5] In microsimulations, the structure is viewed as emergent from the interactions between the individuals, whereas in macrosimulations, the set of individuals is viewed as a structure that can be characterized by a number of variables. Bonabeau[2] summarized the benefits of ABMS over other modeling techniques as follows:

- ABMS captures emergent phenomena.
- ABMS provides a natural description of a system.
- ABMS is flexible.

Specifically, ABMS is superior in modeling the following situations:

- The interactions between agents are complex, nonlinear, discontinuous, or discrete.
- Space is crucial, and agents' positions are not fixed.
- Population is heterogeneous, and each individual possesses different characteristics.
- The topology of the interactions is heterogeneous and complex.
- Agents exhibit complex behavior, especially involving learning, interactions, and adaptation.

Challenges in ABMS

Despite the substantial benefits of ABMS discussed in the previous section, there are several challenges associated with ABMS. Samuelson[12] pointed out that many complex ABMS deal with sufficiently sensitive issues, in which validation becomes problematic, and this difficulty increases as models become more complex. In addition, simulating detailed behaviors of underlying agents could be extremely intensive in computation and therefore become time-consuming.[2] Although the computing power is growing at an impressive pace, the high computational requirements of ABMS remain a problem when it comes to modeling extremely large systems.

In a similar vein, Jennings[13] identified two major drawbacks associated with ABMS:

- The patterns and the outcomes of the interactions are inherently unpredictable.
- Predicting the behavior of the overall system based on its constituent components is extremely difficult (sometimes impossible) because of the strong possibility of an emergent behavior.

Another issue of ABMS in the social science field is that it often involves human agents with potentially irrational behavior, subjective choices, and complex psychology. All of these factors are difficult to measure, quantify, calibrate, and sometimes justify.[2]

ABMS Applications

Practical ABMS is actively being applied in many areas. Examples of applications include the modeling of organizational behavior and psychology,[14] team working,[15] supply chain management and logistics,[16] consumer behavior,[17] social networks,[18] distributed computing, transportation management,[19] and environmental study.[20] In these applications, and in the example shown in figure 5, the system of interest is simulated by capturing the behavior of individual agents and their interconnections. Agent-based modeling tools can be used to test how changes in individual behaviors will affect the system's emerging overall behavior.

ABMS has also been applied to various domains in social and society studies, including population dynamics,[21] the spread of epidemics,[22] biological applications, civilization development,[23] and military applications.[24]

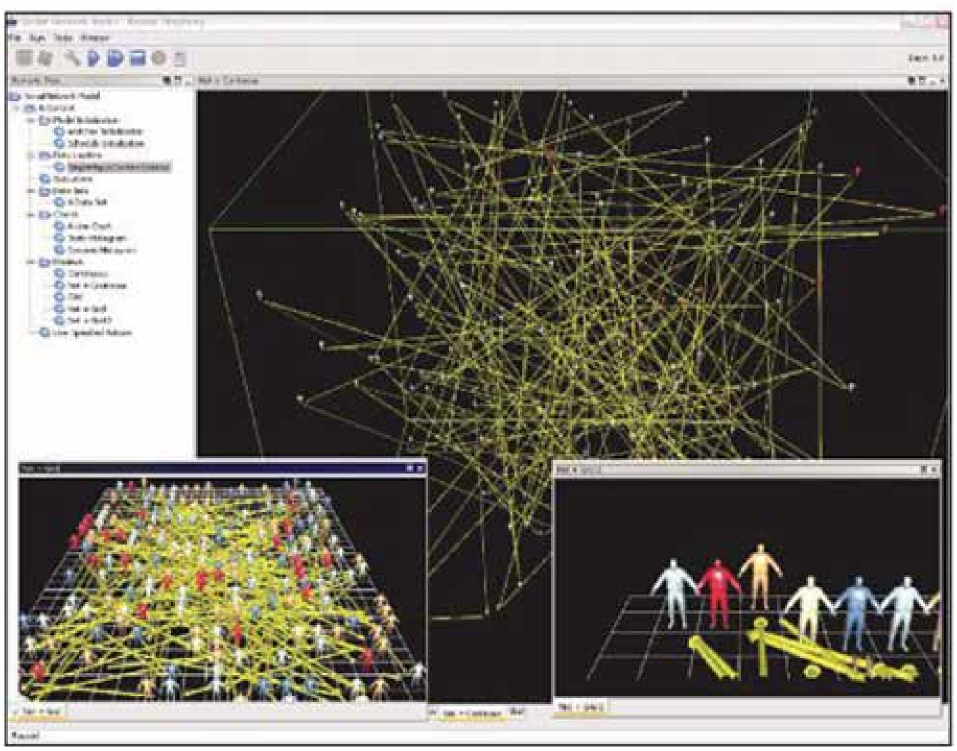


Figure 5. Autonomous agents interact over a self-organizing social network in a Repast simulation of social influence. (**Source: Macal & North**[4])

In their review, Macal and North[4] classified these ABMS applications into two categories, as follows:

- *Small, elegant, minimalist models*—Minimalist models are based on a set of idealized assumptions, designed to capture only the most salient features of a system. These are exploratory electronic laboratories in which a wide range of assumptions can be varied over a large number of simulations.

- *Large-scale decision-support systems*—Decision support models tend to be large-scale applications, designed to answer a broad range of real-world policy questions. These models are distinguished by including real data and having passed some degree of validation testing to establish credibility in their results.

A short list of ABMS applications summarized by Macal and North is presented in table 1.

An Example of ABMS in the Supply Chain

For this section, the researchers use an example from Macal and North[11] to illustrate the general components of an agent-based model, and general methods of modeling an agent-based system, in the supply chain context.

Consider a generic supply chain system consisting of five stages: factories, distributors, wholesalers, retailers, and customers. Each stage is modeled by a set of individual entities, or agents, which interact with each other to form a supply chain network, as shown in figure 6.

For simplicity, the suppliers are ignored. There is only one commodity, no transformation of goods is made, and no assembly of materials into products is

Table 1. Agent-based modeling applications. (**Source: Macal & North**[4])

Business and Organizations	Society and Culture
• Manufacturing operations	• Ancient civilizations
• Supply chains	• Civil disobedience
• Consumer markets	• Social determinants of terrorism
• Insurance industry	• Organizational networks
Economics	Military
• Artificial financial markets	• Command and control
• Trade networks	• Force-on-force
Infrastructure	Biology
• Electric power markets	• Population dynamics
• Transportation	• Ecological networks
• Hydrogen infrastructure	• Animal group behavior
Crowds	• Cell behavior and subcellular processes
• Pedestrian movement	
• Evacuation modeling	

required. The flows of goods and information in the form of orders between stages (agents), as well as physical shipments, are included in the model. The flows of payments and the complexities of pricing, negotiation, and financial accounting that this would entail are not included in this simple model.

Local rules, or behaviors, of the agents that model the flows of goods are specified as follows:

- The customer places an order with the retailer.
- The retailer fills the order immediately from its respective inventory if it has enough inventories in stock.
- The retailer routinely makes an order, and receives a shipment from the upstream wholesaler, to keep its inventory at a certain desired level. The retailer needs a specific local ordering rule to decide how much to order from the wholesaler. The rule could be based in part on estimating future customer demand by using a demand-forecasting rule.
- Similarly, each wholesaler receives a shipment from the upstream distributor, forecasts future demand by the downstream retailer, and places an order with the distributor. This process continues up the chain to the factory who decides on how much to put into new production.

The goal of the agents at each stage (retailer, wholesaler, and factory) is to manage their inventory at an optimized level such that their net costs are minimized (or net gains are maximized). When inventories are low, there is a chance of losing profits because of running out of stock. When inventories are large, agents have to maintain high inventory holding costs. Agents control their inventory level by following their local rules and by processing local information, such that their own benefits are maximal. Local rules dominate the operating mechanism of the system, because none of the agents can access global information (e.g., information other than an agent's neighborhood), none of the agents has a global view of the supply chain, and none of the agents is interested in optimizing the system as a whole. Through local rules of individual entities in this supply chain system, it is expected to observe the emergent behavior, which ultimately could exhibit an equilibrium state for agents at each stage.

In this example, *local information* refers to the experienced order histories that have been received from neighborhood agents (upstream and downstream), *local rules* are the set of rules to maintain a desired inventory level, and *emergent behavior* refers to the equilibrium for each agent.

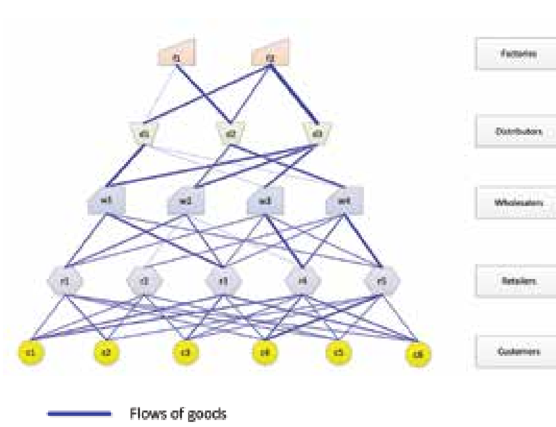


Figure 6. A typical supply chain network and its agents. (**Source: Macal & North**[4])

CHAPTER 2
Agent-Based Methods in Human Decisionmaking

Human Decision–Behavior Modeling Framework

According to Lee et al.,[25] human decision behaviors have been studied in various disciplines, such as AI, psychology, cognitive science, and decision science. Lee et al.[25] classified these models theoretically into three major categories: an economics-based approach, a psychology-based approach, and a synthetic engineering-based approach. Each approach exhibits strengths and limitations. For example, models of the economics-based approach have a solid theoretical foundation, based mainly on the fundamental assumption that the decisionmakers are rational.[26–29] One major limitation, however, is the lack of capability to capture the nature of the human cognition process. To overcome this limitation, researchers proposed models using a psychology-based approach.[30–33] Although these models account for human cognition, they generally examine human behaviors under simplified and well-controlled laboratory conditions. The synthetic engineering-based models, which supplement economics- and psychology-based models, engage a range of engineering methodologies and technologies to assist in reverse-engineering and representing human behaviors in complex and realistic environments.[34–40] Human decisionmaking models in this category consist of engineering techniques used to implement submodules; however, given the possible interactions between submodules, the complexity of such comprehensive models makes it difficult to validate them against real human decisions. Lee et al.[25] proposed a novel, comprehensive model of human decisionmaking behavior, effectively integrating engineering-, psychology-, and economics-based models.

According to Lee et al.,[25] three popular synthetic engineering-based models are Soar, Adaptive Control of Thought–Rational (ACT-R), and BDI. Soar and ACT-R have their theoretical bases in the unified theories of cognition—an effort to integrate various disciplines to capture a single human cognition.[36] Thus, Soar and ACT-R use concepts that correspond to the real mechanisms of those brain activities involved in information processing, including activities of reasoning, planning, problem-solving, and learning. In contrast, the core concepts of the BDI paradigm, originally proposed in folk psychology, allow the use of a programming language to describe human reasoning and actions in everyday life.[41] The BDI paradigm has been successfully applied in many medium-to-large-scale software systems, including an air-traffic management system.[42]

Models of Learning

Lee and Son[43] provided a comprehensive review of various models of human learning behaviors, part of which is summarized in this section. Researchers have conducted extensive research on applying various machine learning algorithms and models, such as statistics, neural networks, and control theory, to mimic human learning behavior. For example, statisticians have introduced Bayesian models as a way to understand how a human being deals with uncertainties. Although many researchers have studied various Bayesian belief network (BBN) methods, such as Bayesian methods,[44,45] quasi-Bayesian methods,[46,47] and non-Bayesian methods,[48,49] a major obstacle to practical implementation of a BBN is the difficulty in constructing an accurate model, especially when the training data is limited. To tackle this problem, Niculescu et al.[50] introduced a framework for incorporating general parameter constraints into estimators for the parameters of a BBN. In a similar vein, Djan-Sampson and Sahin[51] used Scatter Search as a heuristic for identifying the best structure of a BBN. In spite of the above efforts, identifying a BBN structure is still a difficult task compared with other learning techniques. In addition, there is a gap between the BBN learning model and actual human learning, as most of the existing models focus more on finding the best solution (optimal behavior).

Lee and Son[43] described another attempt to develop a human-like learning machine, known as *reinforcement learning* (RL), which was adopted initially in the domain of animal learning psychology that concerned learning by trial and error. Later, in the 1980s, RL was implemented in some of the earliest work in the field of AI. In this field, RL was used in cognitive models that simulate human performance during problem-solving and skill acquisition.[52–55] RL was also used in the human error-processing system.[56] As such, the RL technique was demonstrated to be successful in mimicking human behavior in some sorts of simple problems, especially when prior knowledge is limited. Although BBN training is a non-deterministic polynomial-time hard (NP-hard) problem, training in RL could also be performed relatively easily based on the recursive mathematical formula. A major drawback of RL, however, is its difficulty in complex problems, because the states (which can be exhaustive for complex problems) and actions need to be clearly defined beforehand. As a result, if the environmental factors change, they need to be redefined accordingly. In addition, as compared with the BBN method, the RL method is more limited in its ability to take into account more extensive prior knowledge.

Models of Interactions

Kim et al.[57] discussed various models of human interactions, some of which are summarized in this section. Thomas[58] defined five types of human interactions for conflict management: collaboration, compromise, competition, accommodation, and avoidance. Given a pair of interacting participants, each one selects an interaction type based on his or her goal and willingness to help the other party. Collaboration entails achieving the mutual goal of both participants, in which they are willing to share their knowledge to satisfy their interests. In contrast, competition emphasizes each party's own interest, in which no information is exchanged, and the interests of the other party are predicted based on a win–lose paradigm. Collaboration and competition constitute major concerns in individual and collective behaviors, and these two types of interactions have attracted much attention from researchers. In accordance, several agent-based modeling methods have been applied to address these two issues.[59] For example, Danielson[60] employed an evolutionary game as a theoretical framework to model the collaboration between agents. In a similar vein, Macy and Flache[61] explored collaboration between agents in mixed-motive games by using an RL approach. With regard to competition, Read[62] employed differential equations to represent competition interactions, whereas Pan et al.[63] employed decision rules to represent competitive behaviors between agents.

In a compromise interaction, both parties tended to exchange part of their resources to maximize each party's outcome. Zlotkin and Rosenschein[64] introduced a unified negotiation protocol to illustrate the negotiation process in multiagent systems. When one or both participants are either assertive or cooperative, the interaction type may become accommodation or avoidance. In an accommodation interaction, although the participants may have different personal goals, the more unassertive one sacrifices his or her interests for the other's interests. In this case, the more unassertive one acts as an information sender while the other one is an information receiver. In the case when the conflicting issue is trivial, or both parties believe that the costs outweigh the benefits of conflict resolution, they will choose to avoid interaction, which will resolve the conflict situation. Although there is limited research in modeling accommodation and avoidance interactions among the agents, Kim et al.[57] addressed all five interaction types under the extended BDI framework.[25]

CHAPTER 3
Agent-Based Software Toolkits

Recently, an increasing number of modeling toolkits have become available to facilitate agent-based modeling and applications. Each software toolkit has a variety of characteristics, and many efforts have been attempted to review and compare these toolkits.[65–69] These toolkits are in general integrated tool suites, designed to simplify the construction of agent-based models and the development of agent applications. There is no universal definition of an *agent toolkit*. According to review articles, an agent toolkit could be defined as a software package, application, or development environment that provides modules with a sufficient level of abstraction to allow them to implement agents with desired attributes, features, and rules.[69]

Serenko and Detlor[69] provided a good summary on why agent-based toolkits are needed in general:

- They provide a certain level of abstraction in which programmers can develop their objects.
- They incorporate some features of visual programming, which saves much time and makes development easier, more attractive, and enjoyable.
- They offer run-time testing and debugging environments.
- They allow programmers to reuse classes (definition of objects) created by libraries or other programmers.

Railsback et al.[68] classified ABMS platforms into two categories. The first category follows the framework and library approach, which includes most of the commonly used ABMS toolkits, such as AnyLogic, Repast, Swarm, and Multiagent Simulator of Neighborhoods (MASON). These tools in general provide a framework, which is a set of standard concepts for designing and describing ABMS models. A library of software that implements the framework is also available as a simulation tool. For example, in AnyLogic, a model is constructed with one or more active object classes. A Java application programming interface (API) is provided to guide the use of state charts, variables, functions, and other miscellaneous tools. The second category consists of approaches designed to provide a high-level platform that allows people to build and learn from simple agent-based models. The NetLogo family belongs to this category. In addition, offering a different review, Macal and North[70] distinguished ABMS toolkits based on their simulation scalability. Although AnyLogic, Repast, Swarm, and MASON can be applicable to large-scale agent development environments, NetLogo was designed to fit the agent-based prototyping environment that runs on desktops. Railsback et al.[68] reviewed six ABMS platforms: NetLogo, MASON, Swarm, Repast, Ascape, and AnyLogic. The characteristics of the six toolkits are summarized in table 2 and use the results from both reviews mentioned previously. In terms of animation capabilities, the most updated versions of MASON, AnyLogic, and NetLogo offer both geographic information system and three-dimensional (3D) capabilities.

Table 2. Comparison of ABMS software toolkits.
(**Sources: Macal & North**[70] **and Railsback et al.**[68])

Platform	Scalability	Execution Speed	Programming Language	Primary Domain	Web site
NetLogo	desktop computing	intermediate	NetLogo	social and natural sciences	www.ccl.northwestern.edu/netlogo/
MASON	large-scale	fast	Java	social complexity, physical modeling, AI/machine learning	www.cs.gmu.edu/~eclab/projects/mason/
Swarm	large-scale	slow	Objective-C; Java	general purpose	http://alumni.media.mit.edu/~nelson/research/swarm/
Repast	large-scale	fast	Java; Python; C++	social sciences	http://repast.sourceforge.net/
Ascape	large-scale	fast	Java	general purpose	http://ascape.sourceforge.net
AnyLogic	large-scale	fast	Java	general purpose, distributed simulation	www.anylogic.com

In the following subsections, the authors briefly summarize the characteristics of the six ABMS toolkits individually. Most of the reviews come from Allan,[65] Lytinen and Railsback,[66] and Railsback et al.[68]

NetLogo

NetLogo is a multiagent programming language and modeling environment for simulating natural and social phenomena. Authored by Uri Wilensky[71] in 1999, it has been in continuous development ever since at the Center for Connected Learning and Computer-Based Modeling. It is designed for both research and education and is used across a wide range of disciplines and education levels.[71] Although the primary purpose of NetLogo has been to provide a high-level platform that allows one to build and learn from simple agent-based models, it now contains many sophisticated capabilities (behaviors, agent lists, and graphical interfaces).

NetLogo includes its own programming language that is simpler to use than is Java or Objective-C, an animation display automatically linked to the program, and optional graphical controls and charts. Its programming language includes many high-level structures and primitives that reduce programming efforts. NetLogo runs on the Java virtual machine; thus, it works on all major platforms (Mac, Windows, and Linux) and runs as a standalone application, or from the command line. NetLogo also provides a classroom participatory-simulation tool called *HubNet*. Models and HubNet activities can be executed as Java applets in a Web browser. NetLogo provides an error checker and makes it easy to develop and try code in small steps, but it lacks integrated development environment features, such as a stepwise debugger. Reproducibility may be a concern for some scientific users, because NetLogo does not provide immediate access to the algorithms implementing its primitives. NetLogo is free and open source.

Wilensky[72] summarized that there are four types of agents in NetLogo: turtles, patches, links, and the observer. Turtles are agents that move around in the world, a 2D space divided into a grid of patches. Each patch is a square piece of ground over which turtles can move, and both turtles and patches have coordinates. Links are agents that connect two turtles. The observer does not have a location, but one can imagine it as looking out over the world of turtles and patches.

The NetLogo environment enables exploration of emergent phenomena. In addition to its comprehensive documents and tutorials, NetLogo also comes with a comprehensive library of models and a large collection of pre-written simulations that can be used and modified. These include models in a variety of natural and social science domains, such as economics, biology, physics, chemistry, psychology, system dynamics, medicine, mathematics, and computer science.[73]

Railsback et al.[68] commented that NetLogo is suitable for developing models that are compatible with its paradigm of short-term, local interaction of agents and a grid environment, and not extremely complex. It is even recommended for developing prototyping models that may be implemented later by using lower level platforms; starting to build a model in NetLogo can be a quick and thorough way to explore design decisions. Its intermediate execution speed may not be a significant limitation for many applications, especially compared with the potential reduction of programming time. On one hand, with its heritage as an educational tool, NetLogo stands out for its ease of use and excellent documentation. On the other hand, its simplified programming environment restricts experienced programmers when making a detailed or large-scale model. For instance, it requires having all code in one file and enforces less organizational discipline than is required in Java or Objective-C and thus can be cumbersome for large models.

MASON (MultiAgent Simulator of Neighborhoods)

MASON is a joint effort between George Mason University's (GMU) Evolutionary Computation Laboratory and the GMU Center for Social Complexity. MASON is a single-process, discrete-event multiagent simulation library core in Java, designed to support many agents relatively efficiently, or be the foundation for large custom purpose Java simulations. It is designed to provide more than enough functionality for many lightweight simulation needs. MASON contains both a model library and an optional suite of visualization tools in 2D and 3D. MASON is open source software licensed under the Academic Free License, Version 3.0.[74] For more information on the software, see http://cs.gmu.edu/~eclab/projects/mason/.

MASON was designed as a smaller and faster alternative to Repast, with a focus on computationally demanding models with many agents executed over a variety of iterations. Design appears to have been driven largely by the objectives of maximizing execution speed and assuring complete reproducibility across hardware.[65]

Luke[75] summarized that MASON has the following features:

- Models are fully separated from visualization. One can run a model without visualization or with various different kinds of visualization, and switch among them.
- One can run multiple MASON models in parallel in the same process, and they will not touch each other.
- It is written in Java to make it easy to run in heterogeneous computer environments, with careful improvements of standard Sun classes and library.
- It has a high-quality random number generator.
- Models are largely duplicable, meaning that different simulation runs with exactly the same parameters will produce the same simulation results, even on different machines.
- It is modular and consistent. There is a high degree of separation and independence among elements of the system.
- It is not an easy toolkit for Java but demands significant Java knowledge on the part of its users.

Swarm

Swarm was the first reusable software tool created for ABMS. It was developed at the Santa Fe Institute in 1994 and was specifically designed for artificial life applications and studies of complexity.[65] Swarm was originally developed for multiagent simulation of CAS. Until recently, the project was based at the Santa Fe Institute, but its development and management is now under control of the Swarm Development Group, which has a wider membership to sustain the software. For more information on Swarm, see http://www.swarm.org.[65]

Railsback et al.[68] summarized Swarm's design as follows:

> *Swarm was designed as a general language and toolbox for ABMS, intended for widespread use across scientific domains. The developers started with a general conceptual approach with respect to agent-based simulation software. Key to Swarm is the concept that the software must both implement a model and, separately, provide a virtual laboratory for observing and conducting experiments on the model. Another key concept is designing a model as a hierarchy of swarms, a swarm being a group of objects and a schedule of actions that the objects execute. This is similar to the concepts of context and project now included in Repast Simphony. One swarm can contain lower level swarms whose schedules are integrated into the higher level swarms; simple models have a lower level model swarm within an observer swarm that attaches observer tools to the model.*

Allan[65] reviewed the structure of the Swarm system as follows:

> *In the Swarm system, the fundamental component that organizes the agents of a Swarm model is a swarm. A swarm is a collection of agents subject to a schedule of events. The swarm represents an entire model: It contains the agents as well as the representation of time. Swarm supports hierarchical modeling whereby an agent can be composed of swarms of other agents in nested structures. In this case, the higher level agent's behavior is*

defined by the emergent phenomena of the agents inside its swarm. This multilevel model approach offered by Swarm is very powerful. Multiple swarms can be used to model agents that themselves build models of their world. In Swarm, agents can themselves own swarms, models that an agent builds for itself to understand its own world.

Swarm is a powerful and flexible simulation platform; however, these virtues come at a price. In practice, Swarm has a very steep learning curve. It is necessary to have experience in Objective-C, and possibly Java, to be familiar with object orientation methodology and be able to learn some Swarm code.[65]

Repast (Recursive Porous Agent Simulation Toolkit)

Railsback et al.[68] summarized that Repast development has been driven by several objectives. The initial objective was to implement functionalities similar to Swarm in Java but without adopting all of Swarm's design philosophy and without implementing swarms. The additional objective of making it easier for inexperienced users to build models has been approached in several ways, including a built-in simple model, as well as interfaces through which menus and Python code can be used to begin model construction.[68]

Repast was started as a Java implementation of Swarm but diverged significantly from Swarm. It focuses on modeling social behavior, in the social science domain, and offers support tools for social network modeling. Repast Toolkit Version 3 can be considered as a specification for agent-based modeling services or functions. There are three concrete implementations of this conceptual specification: Repast for Java (Repast J), Repast for the Microsoft.Net framework (Repast.Net), and Repast for Python Scripting (Repast Py). Repast J is the reference implementation that defines the core services. In general, Repast developers recommend that basic models be written in Python by using Repast Py, because of its visual interface, and that advanced models be written in Java with Repast J or in C# with Repast.Net. Repast 3 is available on virtually all modern computing platforms including Windows, Mac OS, and Linux. The platform support includes both personal computers and large-scale scientific-computing clusters.[76]

The Repast Simphony version, or RepastS (the current version released in 2010), uses a new conceptual approach and is a different platform from previous versions. Part of the Simphony version is ReLogo, which is based on NetLogo as it includes many of NetLogo's primitives and its graphical interface tools. NetLogo and ReLogo share a common goal of enabling novice programmers to develop agent-based models.[66] The developers of Repast claim that Repast Simphony models can be developed in several different forms, including the ReLogo dialect of Logo, point-and-click flowcharts translated into Repast Simphony models, Groovy, or Java—all of which can be fluidly interleaved.[77] Repast Simphony provides all the core functionalities of RepastJ or Repast.Net, as well as implementation in Java. RepastJ, Repast.Net, and RepastPy have been superseded by Repast Simphony and are no longer being developed.[65]

Ascape

Ascape is a framework for developing and analyzing agent-based models and was developed by Miles Parker of the Brookings Institution Center on Social and Economic Dynamics, which also developed the well-known Sugarscape model.[65]

Ascape follows some of the ideas behind Swarm; however, it is somewhat easier to develop models with Ascape than with Swarm. Ascape is a high-level framework supporting complex model design, while end-user tools make it possible for non-programmers to explore many aspects of model dynamics. Ascape is written entirely in Java and runs on any Java-enabled platform.[78]

Ascape is released under a Berkeley Software Distribution standard open-source license and thus is free to use and redistribute. The Ascape distribution includes a number of other open-source libraries. Ascape is a research-oriented software, and direct support is not provided, although the Ascape forum serves as a main venue for information exchange and makes use of limited support resources.[78]

In terms of Ascape's applicability to simulation in the social sciences, Ascape is able to implement complex social mechanisms. Allan[65] commented that, "Like Swarm, the only restriction would be finding a programmer with sufficient skills to code the model."

AnyLogic

AnyLogic is claimed to be the only tool that supports three major simulation modeling methodologies in place today: system dynamics, discrete-event, and agent-based modeling.[79] It provides a visual language that simplifies development of agent-based models significantly. For example, Unified Modeling Language states that charts are used to define agent behaviors, action charts are designed to define algorithms, environment objects are used to help describe the agent environment and collect statistics, and events are used to describe occasional or time-certain occurrences. These constructions allow users to describe almost all the behavioral aspects of agents. In addition, users can always write specific Java code to model something more specific or unanticipated. Agent-based models can be combined seamlessly with discrete-event and system dynamics models. The agents themselves may be included inside system dynamics stock and flow diagrams or flow charts. In other words, AnyLogic provides a highly friendly environment for multiparadigm modeling, which combines different simulation methods within one model in various ways: hierarchical, series hand-off, or parallel.

An agent-based model built in AnyLogic usually contains a main class and one or multiple agent classes. Complying with the object-oriented programming principles, each type of agent is modeled independently as an active object class in which all the agent's capabilities are defined. Along with the basic Java API, AnyLogic provides a library named Agent (AnyLogic 6 Engine API) to support communication between agents as well as interactions between agents and (continuously or discretely spaced) environment. External libraries and packages can also be added as dependencies and used in AnyLogic models. For example, Lee et al.[80] have used the Java matrix package, JAMA, developed by Hicklin et al.[81] to improve the matrix computing efficiency. In addition, AnyLogic provides a variety of statistical tools, such as dataset, bar chart, time plot, and histogram. These components are usually added into the main class for demonstrating analysis results during the simulation run. If needed, various add-on modules allow users to perform various types of experiments, such as parameter variation, Monte Carlo, or optimization.

As a widely used commercial software product for agent-based simulation, AnyLogic has been improving not only its functional aspects, but also its exportability and animation capabilities. For example, models can dynamically read and write data to spreadsheets or databases during a simulation run, as well as charting model output dynamically. Furthermore, external programs can be initiated from an AnyLogic model for dynamic communication of information, and vice versa. The most current version, AnyLogic 6.7, allows the integration of AnyLogic models with external Java applications. An exported model can also be freely installed and run on many computers with supported operating systems. From release 6.5 and on, AnyLogic has been supporting both 2D and 3D animations. The current release 6.7 has made AnyLogic 3D animation compatible with Java applets. In other words, users can publish their models with 3D animation on the Web, and remote users are able to view and navigate them in the 3D scene from their own Web browsers. It should be noted, however, that high-quality animation demands considerable computer memory that may slow down the model's execution speed.

AnyLogic offers different versions for three types of operating systems: Microsoft, Mac, and Linux.[82] As previously mentioned, once compiled, the simulation can be run on any Java-enabled operating system. The AnyLogic Web site shows many examples of models that have been developed for a diverse range of applications, including the study of social, urban, and ecosystem dynamics (e.g., a predator-prey system); planning of healthcare schemes (e.g., the impact of safe syringe usage on HIV diffusion); computer and telecommunication networks (e.g., the placement of cellular phone base stations); the location of emergency services and call centers; and pedestrian dynamics. There are also online video tutorials.

CHAPTER 4
Agent-Based Transportation Modeling Platforms

Basic Structure of Agent-Based Transportation Platforms

There are several well-known agent-based transportation modeling platforms, including, but not limited to, Transportation Analysis and Simulation System (TRANSIMS),[83] Multi-Agent Transport Simulation Toolkit (MATSim),[84] Sacramento Activity-Based Travel Demand Simulation Model (SACSIM),[85] Simulator of Activities, Greenhouse Emissions, Networks, and Travel (SimAGENT),[86] Open Activity-Mobility Simulator (OpenAMOS),[87] and Integrated Land Use, Transportation, Environment (ILUTE).[88] (Some systems promote themselves as ABMS and some do not; the authors classify them, herein, in the same category because they exhibit similar structure in the same modeling paradigm.)

Most of these agent-based models are individual-based models. They have roots in the activity-based travel demand models and are commonly characterized by a similar feature. Naturally, they all exhibit similar architecture, more or less, which combines two transportation components—travel activities and network loading—into an integrated microsimulation platform. An agent in those models stands for a human/person/traveler in general.

It should be noted that these individual-based transportation systems also share some similarities with the simulation-based dynamic traffic assignment (DTA) approaches (e.g., Dynamic Network Assignment-Simulation Model for Advanced Roadway Telematics (Planning version; DYNASMART-P),[89,90] Dynamic Urban Systems for Transportation (DynusT),[91-93] and Dynameq[94,95]). The similarities are summarized as follows:

- Both systems use some sort of simulation as a network loading method to measure travel time (and accessibility). Travel time then feeds back into the travelers' route choice components to revise the routes.
- Both systems (could) run iteratively to accomplish a convergence and consistency between the travelers' route choice decisions and the network-wide traffic performance.

The dissimilarities between a simulation-based DTA approach and an agent-based transportation model are recognized as follows:

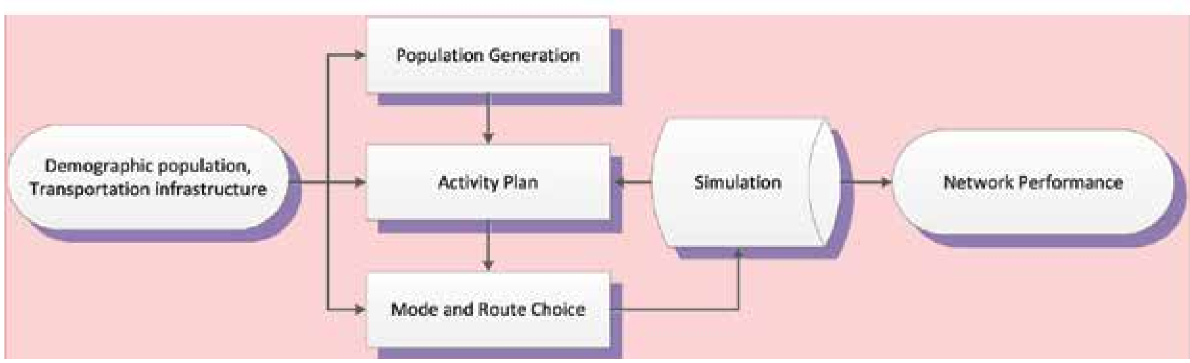
- Simulation-based DTA feeds updated travel time into travelers' route choice decision only. An agent-based model provides the feedback of travel time to a multidimensional decision domain, including not only travelers' route choice decisions but also a set of activity decisions, like activity location, schedule, and change of participation agenda.
- Because the decision domain of the agent-based decisions is much more complex than a simulation-based DTA, an agent-based system usually adopts heuristic rules in feedbacks to achieve approximate convergence and consistency. Thus, compared with an agent-based model, simulation-based DTA may spend more computational resources to achieve equilibrium between the network loading and assignment result.

The existing agent-based transportation system in today's literature in general has the distinguishing feature of integration combining three components: travelers' activity decisions (multidimensional), travelers' route decisions, and microsimulation. Although the same decisions have been well-studied in two modeling paradigms, that is, activity-based travel demand models and DTA, the two differ in the types of feedback they utilize. In the case of an activity-based travel demand model, feedback is between travelers' activity decisions and traffic simulation. For simulation-based DTA, feedback is between travelers' route choice decision and traffic simulation. In this regard, an agent-based transportation system tries to have feedback among the three components together.

In summary, most existing agent-based transportation systems follow a structural design similar to that shown in figure 7:

- An agent represents an individual person or traveler and is associated with the individual demographic and travel characteristics.
- The system generates demand, or an activity travel plan, for each individual agent based on the demographic characteristics.
- Activity plans are revised, enhanced, and finalized such that all plans meet spatial (facility) and temporal (schedule) constraints.
- The activity plans are fed into a microsimulation to produce the transportation results network-wide.
- Network performance is a source of feedback to both activity plan and route choice decisions. Agents revise the activity travel plan and route choice decision, such that both decisions are optimized simultaneously.

What follows is a brief review of a selected number of agent-based transportation platforms, with a focus on model structure, inputs and outputs, and functionality. The description of review is based on scientific papers in the published literature in this document and does not suggest any level of development, maturity of application, or readiness to use, unless otherwise noted. The overall framework for all platforms exhibits a certain similarity, that is, microsimulation of the activities of agents (or individual human/person/traveler/driver–vehicle unit); however, the design



© Hong Zheng

Figure 7. Model structure of an agent-based simulation approach.

process, functionalities, agent activities, data structures, and emphasized policy decisions may vary substantially.

TRANSIMS (TRansportation Analysis and SIMulation System)

TRANSIMS is an open-source, integrated transportation modeling and simulation toolbox for regional transportation system analyses. It is designed to assist transportation planners in analyzing accurate and complete information on traffic impacts, congestion, and pollution. It was originally developed by the Los Alamos National Laboratory for the U.S. Department of Transportation and has been continuously supported and improved. At present, the Travel Model Improvement Program sponsors the research and development of TRANSIMS.

TRANSIMS is designed based on microsimulation that adopts the activity-based approach to generate individual's activities instead of conventional trip-based origin–destination (O-D) matrices. It builds synthetic populations based on census and survey data, estimates activities for all individuals and households, plans multimodal trips satisfying those activities, selects routes for those trips, and runs a microsimulation of all vehicles (including transit) over the entire transportation system. The goal is to gain detailed traffic data in a given study area to support traffic, travel demand, and transportation policy analyses.

The fundamental concept of TRANSIMS is to simulate individual travelers. The overall model framework is structured as follows:[83]

- A synthetic population is created from demographic data.
- Another module of TRANSIMS generates synthetic activities (e.g., sleeping, eating, working, and shopping) and activity locations for each synthetic individual.
- Each individual in the simulation then plans that individual's transportation mode and routing choices.
- The plans of each individual are fed into a transportation microsimulation.
- The overall traffic performance then feeds back to the activity generator and route planner, leading to partial revision of individual activity and route decisions, using heuristic rules accordingly.

In this modeling framework, because the processes are conducted independently and sequentially, consistency of the output attributes produced by each process needs to be secured. TRANSIMS relaxes partial inputs of each module and then utilizes feedback loops to achieve consistency between the modules, as shown in figure 8. Such feedback schemes, however, may not necessarily converge, and the effect for TRANSIMS may occur in particular where new attributes of modules are generated heuristically.[96]

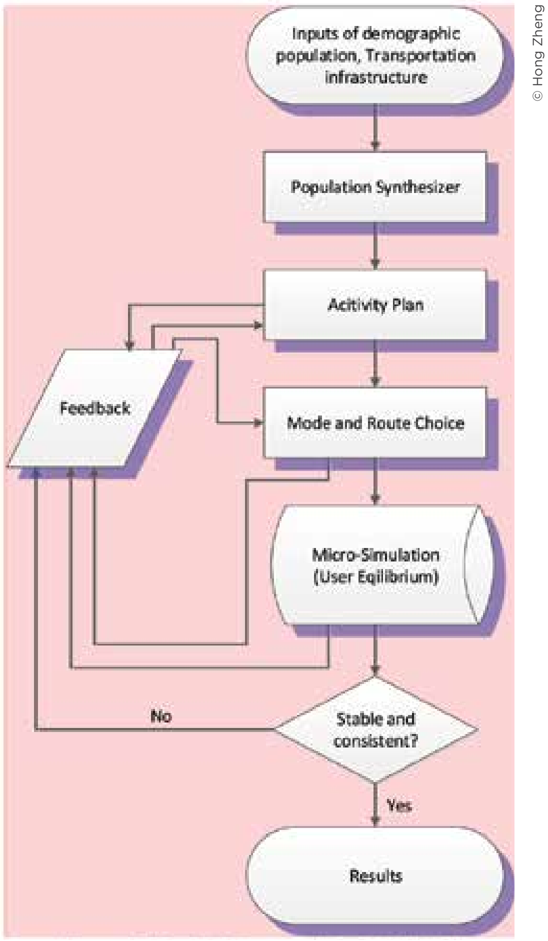


Figure 8. Model diagram in TRANSIMS.

The route choice planner in TRANSIMS attempts to assign the time-dependent fastest route to individual travelers. It implements Dijkstra's time-dependent algorithm to solve the fastest route. This solution requires accurate time-dependent, link travel-time information, which is provided by the microsimulation in TRANSIMS. The produced traffic performance measures then feed back to the route planner, which revises travelers' route choice decisions, and thus loops in an iterative feedback framework between routing and simulation. The difficulty of this approach is that, because no congestion information is available, the system cannot initially predict the fastest route. This problem is addressed via an iterative relaxation approach, that is, one generates an initial set of routes, runs it through the microsimulation, re-plans a fraction (f_r) of the trips, runs the microsimulation again, and so on, until a certain convergence criterion is verified. Choosing a f_r too large can result in oscillations that never converge; choosing a f_r too small can cause limited improvement in each iteration, which leads to slow convergence.[97]

The traffic simulation component in TRANSIMS is microscopic, consisting of car-following and lane-changing models. The car-following model in TRANSIMS is based on cellular automata, that is, a road is composed of cells, and each cell can either be empty or occupied by exactly one vehicle.[98] The length of the cell is defined by the jam density. Because movement has to be from one cell to another in one direction, velocities have to be integer numbers between zero and maximal velocity. The interval (i.e., time step) of simulation is 1s. During each time step, the velocity of each vehicle is updated by predefined rules based on the gap. The vehicle position is then updated in accordance with the updated vehicle velocity. The lane-changing models in TRANSIMS are rule-based.

Because TRANSIMS utilizes the detailed microsimulation to simulate vehicle movements, it is able to produce traffic flow dynamic characteristics that are consistent with the macroscopic traffic flow properties. The cost is high demand of computation. Although TRANSIMS has simplified microscopic rules, the running time is still expensive because of the detailed level of vehicle movement for each individual vehicle. Compared with MATSim, TRANSIMS comes with a significantly prolonged run time, particularly for large-scale scenarios. The issue is due to the difference in philosophy of the system architecture design—TRANSIMS seeks to model traffic flow dynamic characteristics more than to conserve computational time, but MATSim does the reverse.

MATSIM (MultiAgent Transport Simulation)

MATSim is a microsimulation platform implemented as a Java application. It also adopts the activity-based approach to generate and simulate individuals' activities. *Agent* stands for the individual travelers, and *agent behavior* refers to an individual's daily activity travel plan and route choice.

MATSim designs two layers: the *physical layer,* which simulates the physical world where the agent (or traveler) moves, and the *mental layer*, in which the agents generate strategies, including routes, mode choice, and daily activity plans.(99) It is clear that the mental layer refers to model logics that create an agent's daily activity decisions, and the physical layer is a microsimulator. Outcomes of the two layers feedback to each other in an iterative manner, producing the traffic simulated on the roadway network at a microscopic level. The overall system architecture of MATSim is similar to TRANSIMS, except that the details of implementation vary.

The mental layer is the conventional activity model counterpart, which generates the agents' activity-travel patterns, composed of two components: activity agenda and mode or route choice. The activity generation module creates a 24-hour activity agenda plan for each agent. The quality of an activity plan is measured by a score, quantitatively measured by the sum of the utilities of all performed activities, as well as the travel disutilities for trips connecting one activity location to another.(100)

The physical layer is the microsimulation. MATSim does not adopt car-following and lane-changing as detailed as in TRANSIMS. Instead, it utilizes a spatial queue model to measure traffic dynamics and queue spillovers in the traffic simulation.(101) The purpose of this design is (a) to capture the queue spillovers (rather than point-queue model) at a fine-resolution level by simulating each individual traveler and (b) to make the simulation as simple as possible by modest computation. In the spatial-queue simulation, roadways are essentially represented as first-in–first-out queues, with the additional restrictions that (a) vehicles have to remain for a certain time on the link, corresponding to the free-flow travel time and (b) links possess storage capacity, and once saturated, no more vehicles can enter the link. To speed up the computation, MATSim utilizes parallel computation of the spatial queue model in microsimulation. Because the queue model needs less data and computing resources, it runs much faster than does TRANSIMS.

A noticeable shortcoming of the spatial-queue model is that the traffic dynamics may not be realistic, and the speed of the backward wave may not be modeled correctly. A vehicle that leaves the head of a link immediately opens a new space at the tail of the link into which a vehicle can enter, indicating that the backward wave speed is roughly the length of the link per time step. This effect is especially significant during congestion, not only losing appropriate flow dynamics propagated along the space and time dimensions but possibly resulting in an overestimation of travel time. This is the tradeoff elected by the MATSim system architecture design, favoring fast running time but at the cost of simplifying the traffic flow model.

MATSim runs its activity plan, microsimulation, activity re-plan, microsimulation, and so on, iteratively. The goal of the iterative loop is to find the stationary state of the system, where an agent cannot improve its score by revising the plan. The overall simulation system consists of the following steps:

- A set of initial plans has to be generated.
- The plan selection mechanism of the agent database chooses one plan per agent for execution.
- Run simulation to execute the plans, produce a new travel time for each trip, and re-score the plans.
- A subset of the agents is chosen for plan adjustment or new plan generation by external strategy modules.
- Run external strategy modules, and each agent is updated with a new or revised plan.
- Run mode and route choice module to produce a route for each agent.
- If stop criterion is satisfied, then stop; otherwise, go to simulation.

The external strategy module that performs the activity re-plan is called *planomat*.[102] Planomat applies a genetic algorithm (GA) to revise activity plans in a multidimensional domain, including location choice, mode choice, and the choice of the activity pattern. The major drawbacks of the GA are (a) the solution quality is unknown; and (b) it requires an expensive run-time to obtain a fine solution. After the activity re-plan is executed by planomat, the mode and route choice module then connects a set of activities located at different places by specifying the trip characterized by the choice of the mode of transportation and specific routes. Finally, MATSim is designed to run millions of agents in a metropolitan area. In this regard, MATSim is computationally fast.

OpenAMOS (Open Activity-Mobility Simulator)

OpenAMOS has its roots in Activity-Mobility Simulator and PCATS (Prism-Constrained Activity Travel Simulator). Today, OpenAMOS is the open source, activity-based, travel-demand model within SimTRAVEL, which is an integrated urban continuum framework. SimTRAVEL integrates the following four components: PopGen (a population generator), UrbanSim (used as the urban simulation system), OpenAMOS (for activity-travel demand), and Multiresolution and Loading of Transportation Activities (MALTA; used for dynamic traffic assignment).

PopGen is primarily a population synthesizer (PopSyn). By using zonal socioeconomic data and household travel survey data, PopGen generates a synthetic population, matching it with the census attribute distribution at both the household level and person level. This step generates a basic skeleton around which the complete activity–travel agenda of a person will be formed.

In the next step, OpenAMOS uses microsimulation to generate daily activity travel patterns for the individual travelers synthesized by PopGen. The simulator assumes a sequential history and time-of-day dependent structure. Implementation of microsimulation approaches usually entails the generation of synthetic households and their associated activity–travel patterns to achieve forecasts with desired levels of accuracy.

The sequential approach segments the entire daily activity–travel pattern into various components, as follows:[103]

- Activity type choice models.
  - Home-based and non-home-based.
  - Workers and non-workers.
- Activity duration models.
  - Workers and non-workers.
  - By activity type.
- Activity location choice models.
  - Home-based versus non-home-based.
  - Workers and non-workers.
  - By activity type.
- Mode choice and mode transition models.
  - Home-based and non-home-based.
- Initial departure timing models.
  - Workers and non-workers.
- Initial location models.
  - Workers and non-workers.

Each component can be estimated as a multinomial logit model. Multinomial logit models of activity type choice are estimated by the standard maximum-likelihood method. These steps finalize the blocked attributes (e.g., start time, end time, type, and location of each fixed activity) of an individual's activity–travel pattern. The open attributes of an individual's activity–travel pattern are estimated by a microsimulator, called *PCATS*.[87] PCATS recognizes that the speed of travel is finite and the time available for travel is limited; thus, the individual's trajectory in time and space is necessarily confined within Hagerstrand's prism.[104]

In recent years, OpenAMOS was integrated with a DTA simulation platform, called *MALTA*. In this framework, MALTA serves as a dynamic network loading engine: It takes as input the auto trips produced by OpenAMOS and replicates the dynamics in network traffic flow along the temporal dimension. To a certain degree similar to TRANSIMS and MATSim, the network performance produced by MALTA feeds back into OpenAMOS, and many decisions in OpenAMOS—which are in part based on the travel time information (e.g., travel mode choice, destination choice, etc.)—are revised in OpenAMOS and PCATS in the next iteration. Therefore, in the integrated framework, network performance simulated by MALTA not only feeds back into the traveler's route choice decision, but also feeds back into the traveler's activity–travel pattern decisions. This feature is the linchpin of most existing individual-based agent models characterized by integration of travel activities and traffic simulation.

SACSIM (Sacramento Activity-Based Travel Demand Simulation Model)

SACSIM is a regional travel forecasting model used by the Sacramento Area Council of Governments. Within SACSIM, an integrated activity-based, disaggregate econometric model, the Person Day Activity and Travel Simulator (DaySim), simulates each individual's full-day activity and travel schedule.[85,105]

In SACSIM, a representative PopGen (i.e., DaySim's PopSyn component) creates a synthetic population comprised of households drawn from the region's U.S. Census Public Use Microdata Sample and allocated to parcels. In SACSIM, two decision choices are modeled separately. Long-term choices (e.g., work location, school location, and auto ownership) are simulated for all members of the population. DaySim then creates a 1-day activity and travel schedule for each person in the population, including a list of their tours and trips on each tour. DaySim consists of a hierarchy of multinomial logit and nested logit models.

SACSIM is integrated with the traffic assignment in an analytical framework. DaySim uses network performance measurements to model a person's activity and travel patterns, which are then loaded to the network to determine congestion and network performance for the next iteration. The model achieves equilibrium when the network performance used as input to DaySim matches the network performance resulting from assignment of the resulting trips.

The SACSIM equilibrium procedure is solved analytically by a Frank–Wolfe-like algorithm, which contains two loops. The inner loop is designed to achieve equilibrium of the traffic assignment and runs assignment for four time periods. Each period employs multiclass equilibrium assignment, with classes composed of a single-occupant vehicle (SOV), a high-occupancy vehicle (HOV) not using median HOV lanes, and HOVs using median HOV lanes, solved by a Frank–Wolfe similar convex combination method. The outer loop generates demand O-D flow matrices. In the i^{th} outer loop, DaySim is run on a subset of the synthetic population, consisting of the fraction $1/s_i$ (i.e., $100/s_i$ percent) of the households; the subset is selected by proceeding uniformly every s_i households. When the i^{th} outer loop reaches equilibrium (i.e., the inner loops reach convergence), link volume is combined in a convex combination with the prior outer loop link volume. This is the preloading method intended to prevent link volume oscillation between outer loops. In each outer loop, a fraction of $1/s_i$ population is synthetic until, after s_i iterations of outer loop, all households are synthetic.

ILUTE (Integrated Land Use, Transportation, Environment)

ILUTE is a comprehensive urban transportation system maintained and promoted by a consortium of Canadian universities. The objective is to develop models to capture interactions between urban land use, travel demand, the transportation system, and environmental impacts, and to address high-resolution policy analysis in a variety of transportation and urban planning contexts. ILUTE inherently is an integrated land use (urban) transportation system. Land use and transportation are fundamentally linked: The land use pattern directly determines travel needs, activity agenda type and participation, and viability of alternative travel modes. Transportation, in turn, influences land development and location choices of residents and firms. ILUTE is designed to capture interactions and policy effects between land development and transportation.

ILUTE is a disaggregated, behavioral approach to study the urban transportation system by simulating activities of individual objects (agents) as those activities evolve. In ILUTE, the agents could be various entities, including persons, families, households, houses, buildings, firms, or road and transit networks. ILUTE simulates system evolution in the long run, usually on a month-to-month or year-to-year basis. This is accomplished by simulating one typical day of the system as a means to capture the state of that simulated month or year.

Microsimulation simulates the behavior and state of agents (objects) in the system. Simulating individual objects means knowing when new objects are generated in the system and when they leave. For persons, it simulates the processes of birth and death as well as the process of in-migration and out-migration. For firms, it simulates firms opening, relocating, and closing. For households, it simulates member change, property change (auto ownership), and relocation. This structure exhibits a certain similarity to the Sugarscape model in the social science domain. ILUTE aims to simulate the emergent land-use transportation interactions in the long term.

To model long-term system evolution, ILUTE extends classic market theory to the microsimulation framework to study market-based decisionmaking and interactions between various suppliers and consumers in a variety of markets. The markets studied by ILUTE include the residential housing market, marriage market, commercial real estate market, land market, labor market, vehicle market, and travel market.

The design of ILUTE embodies the following fundamental principles:

- Microscopic simulation-based.
- Agent-based design and implementation.
- Household- and firm-based.
- Individual persons are synthesized with a set of attributes consistent with frequency distributions exhibited in census data.
- Integration of a set of prototype models, including a land use model; an activity-based travel demand model; urban economics, housing, and labor market models; and an auto ownership, population demographics, and emission model.

Salvini and Miller[88] introduced an early structure of ILUTE as follows:

> *The 'behavior core' of the model has four inter-related components: land use, location choice, auto ownership, and activity travel. ILUTE enables researchers to capture the complex interactions that occur within an urban system. The transportation system, for example, is one of many interconnected factors affecting the quality of 'life' within the simulated system.*

As an integrated full-feedback model, ILUTE allows higher-level decisions (e.g., residential mobility) to influence lower-level decisions (e.g., daily travel behavior) and vice versa.

In the ILUTE system, travel activities are quantified as short-term measurements. These reciprocally feedback to an individual's or agent's high-level behavior and long-term decisions in categories such as residential mobility, location choice, market clearing, and economic decisions.

The housing market module models the high-level decision of residential mobility, which consists of three steps: mobility decision, a search process, and a bid. Mobility desires are generally triggered by a stress manager, but may also be triggered on a random basis. Once the desire to enter the housing market has been triggered, it is the role of the housing market module to accomplish the search and bidding process. In the ILUTE structure, a multinomial logit model was applied to make location choice decisions, and a two-level constrained and directed search process was applied to find the final bid price.[106]

The *auto transaction module* is a properly estimated empirical model that uses a nested logit equation with calibrated real-world parameters.[107,108] The module uses attributes of the households, owners, drivers, and their current vehicle bundle, and also uses the attributes of each vehicle to determine whether the household will maintain its existing auto ownership level or will purchase, dispose of, or trade a vehicle.[109]

In the latest version, ILUTE contains an activity-based travel demand component, called *Travel Activity Scheduler for Household Agents* (TASHA), to generate the household and person activity–travel patterns and to perform scheduling. It uses a set of activity episode frequencies and a probability density function to generate activity participation agenda (e.g., work, school, shopping, other, and free time) and then searches a feasible start time and duration time by joint probability density functions. In such a manner, activities occur in space and time and have various scheduling dependencies. ILUTE then applies a tour-based method to generate a set of activity chains (with specific modes) for each individual. The tour-based model explicitly accounts for the car-conflicting constraints. A clean-up algorithm is applied to reflect final scheduling and fine-tuning just before or during execution of the schedule.[110] TASHA is designed so that it interfaces readily with a variety of network assignment models. As reported in Miller et al.,[111] it can be used with either Equilibre Multimodal, Multimodal Equilibrium (EMME) for road and transit assignments, or MATSim for road assignments.[112]

SimAGENT (Simulator of Activities, Greenhouse Emissions, Networks, and Travel)

SimAGENT is a system that features PopGen + CEMSELTS + CEMDAP + traffic assignment. In this structure, PopGen generates a synthetic population of individuals and households. The Comprehensive Econometric Microsimulator for Socioeconomics, Land Use, and Transportation System (CEMSELTS) simulates and generates long-term demographic attributes for persons and households. The Comprehensive Econometric Microsimulator for Daily Activity travel Patterns (CEMDAP) generates the daily activity travel plan for each individual, and the traffic assignment component determines a traveler's route choice decision and measures the overall network performance. In recent years, the SimAGENT model system has been applied to the Southern California region by the Southern California Association of Governments.

PopGen is used to generate a synthetic population of individuals and households such that the distributions of socioeconomic and demographic attributes in the synthesized population match the known population distributions. In SimAGENT, a synthetic population is generated based on a set of control variables whose known (census) distributions drive the population synthesis process.(113)

The generated synthetic population serves as the input into the module of CEMSELTS. *CEMSELTS* is a microsimulation capable of modeling medium- and long-term socioeconomic choices of individuals and households. All of the variables that can be simulated by CEMSELTS are stripped away from the synthetic population generated by PopGen and replaced with simulated values from CEMSELTS. This step is to provide a rich set of socioeconomic inputs for activity-based modeling and create a system where long- and medium-term attributes (e.g., worker and student, work and school location, work duration, residential location, and auto ownership) are sensitive to household and person demographic characteristics.(113) In the current design, sets of rule-based probability models and logit models are used within CEMSELTS to determine the medium- to long-term demographic attributes for each person and household.

With well-defined demographic attributes of households and persons, daily activity–travel patterns are generated by CEMDAP. *CEMDAP* is a micro-simulation engine that simulates activity–travel patterns of all individuals in the region for a 24-hour period along a continuous time axis. CEMDAP generates an activity–travel plan for each individual in two steps: (1) generate mandatory activity and schedule (e.g., work and school activity participation and timing) and (2) generate the full daily activity agenda and schedule.(86)

The output of CEMDAP generates a set of time-dependent trip interchange matrices of O-D among the traffic analysis zones, which feed into the traffic assignment component to determine routes and the overall network performance. At the current stage, different traffic simulation models are under investigation to be used in integration with the SimAGENT, including TRANSIMS and MATSim.(86)

CHAPTER 5
Agent-Based: A System Paradigm Applied in the Transportation Field

Multi-Agent System—A Computational Method For The Distributed Systems

This section reviews ABMS in another methodological domain, that of AI, in which ABMS is viewed as one of the powerful computing technologies. Other than the transportation systems discussed in chapter 4, which are individual based models, that is, models that treat each individual person or traveler as an agent, ABMS scoped in this section is regarded as a method in system modeling. More specifically, the common feature found in such studies is that the inherent distribution allows for a natural decomposition of the complex system into multiple subsystems. The subsystems interact with each other following local rules to achieve a desired global goal. It is these subsystems that are modeled as agents, and the operation of agents is supported and managed by distributed software platforms known as *MAS*.

Since its inception around the mid-1980s, MAS has become a key concept and method in DAI.[114] *DAI* is a subfield of AI dedicated to developing distributed solutions for complex problems regarded as requiring intelligence. DAI is closely related to MAS, and the use of the term *MAS* in those studies essentially describes an agent-based computational method of DAI.

The agent paradigm in AI is based on the notion of reactive, autonomous, internally motivated entities that inhabit dynamic, not necessarily fully predictable, environments. An agent is autonomous and decides for itself how to relate data to commands to achieve goals. According to the National Aeronautics and Space Administration (NASA),[115] "Autonomy is the ability to function as an independent unit or element over an extended period of time, performing a variety of actions necessary to achieve predesigned objectives while responding to stimuli produced by the system."

DAI solves problems by using multiple cooperative agents. In these systems, control and information are often distributed among a set of collectively interactive subsystems and components, represented by agents. This reduces the complexity of each subsystem and allows subsystems to work in parallel and to speed up problem-solving. Each agent also has resource and knowledge limitations, which could limit the ability of a single agent system to solve large, complex problems. In general, the learning and cooperation of multiple agents contribute to improving the performance of the agent group as a whole and increasing the domain knowledge of the group. Under this concept, the MAS can aid in the distribution of the problem over the various agents (subsystems) that comprise the MAS and facilitate coordination of the activities of the integrated system when required.

MAS can be characterized by the interaction among many agents that are trying to solve a variety of problems in a cooperative fashion. Along with some AI, an intelligent agent could have some additional attributes that enable it to solve problems by itself, to understand information, to set up goals and intentions, to draw distinctions between situations, to generalize and synthesize ideas, to model the world they operate in and plan, and to evaluate alternatives. The problem-solving component of an intelligent agent can be a simple rule-based system, a neural network, or some fuzzy rules.

Learning and cooperation among neighborhood agents is one of the important features of MAS. Dowell and Bonnell[116] classified the learning strategies into the following four categories:

- *Control learning*—Learning and adapting to work with other agents involves adjusting the control of each agent's problem-solving plan or agenda.
- *Organization learning*—Learning what type of information and knowledge each agent possesses allows for an increase in performance by specifying the long-term responsibilities of each agent.
- *Communication learning*—Learning what type of information, knowledge, reliability, and capability each agent possesses allows for an increase in performance by allowing improved communication.
- *Group observation and discovery learning*—Individual agents incorporate different information and knowledge.

Strength of MAS

Parunak[117] listed the following characteristics for an ideal application of agent technology:

- *Modular*—Each entity has a well-defined set of state variables that is distinct from those of its environment, and the interface with the environment can be clearly identified.
- *Decentralized*—The application can be decomposed into stand-alone software processes capable of performing useful tasks without continuous direction from some other software process.
- *Changeable*—The structure of the application may change quickly and frequently.
- *Ill-Structured*—All information about the application is not available when the system is designed.
- *Complex*—The system exhibits many different behaviors, which may interact in sophisticated ways.

Bonabeau[2] suggested that MAS is appropriate:

- When the interactions between the agents are complex, nonlinear, discontinuous, or discrete, and behavior of individuals cannot be clearly defined through aggregate transition functions.
- When the population is heterogeneous, each individual is different, and the behavior of agents is stochastic in nature.
- When the agents exhibit complex behavior, including learning adaptation.
- When activities are a more natural way of describing the system than processes. When the appropriate level of description or complexity is not known ahead of time.

Adler and Blue[118] concluded, in summary, that the multiagent technology can significantly enhance the design and analysis of problem domains under the following three conditions:

- The problem domain is geographically distributed.
- The subsystems exist in a dynamic environment.
- The subsystems need to interact with each other flexibly.

It is believed that the domain of traffic and transportation systems is well-suited for an agent-based approach because of its geographically distributed and dynamically changing nature.[119,120]

Weakness of MAS

Bernhardt[121] summarized the weakness of modeling MAS as follows:

- Requires significant quantities of data and can be computationally intensive.
- Simulation of individual agents requires significant computational power, particularly when individual agents have complex characteristics and decisions, as with human beings.
- Often requires behavioral data, which may be difficult and costly to obtain.
- Difficult to validate such a model, particularly if the goal is prediction of behavior in an untested system.

In their review of transportation applications of MAS, Kikuchi et al.[122] summarized the features of MAS as follows:

- Local state and local knowledge dictate the actions of an agent. This means that individual agents do not make globally optimal decisions. Reconciling decisionmaking based on local knowledge with the desire to achieve globally optimal performance is a problem when agent-based modeling is used for system optimization.
- Of special concern, MAS may not be appropriate for control problems in which global constraints and objectives have to be satisfied.
- To apply MAS, the analyst needs to feel comfortable with the idea of delegating tasks to the agents, rather than controlling the tasks. The agents that work on behalf of the analyst eventually exhibit a self-organizing character.
- Calibration of parameters is difficult because how individual behavior affects overall behavior is not completely known.

MAS Practiced in Transportation Problems

MAS has been widely applied by both researchers and practitioners in a spectrum of disciplines, from biology, business, and computer simulation to social science, political science, and economic science. Knowledge of ABMS and the recognition of applications continue to expand (for comprehensive overviews of a variety of applications of MAS for traffic and transportation problems, see references 2, 123, and 124). This section summarizes part of the applications developed so far in the transportation field that apply to MAS. Note that the description of review in this chapter is mainly based on scientific papers in the published literature and does not suggest any level of development, maturity of application, or readiness to use, unless otherwise noted. Because interest in MAS continues to expand, the number of applications continues to grow as well.

From the traffic and transportation management perspective, the most appealing features characterized by MAS are autonomy, collaboration, and reactivity. Transportation systems modeled by MAS allow distributed subsystems to collaborate with each other to perform traffic control and management based on real-time traffic conditions.

In recent years, more and more agent-based traffic and transportation applications have been reported, including modeling and simulation,[125–129] traffic control,[130–139] traffic management frameworks (see references 114, 119, and 140–144), dynamic routing (see references 118, 140, and 145–147), congestion management,[148,149] fleet management,[150,151] rail traffic,[152–155] and air traffic.[156–158]

Most existing MAS seen in transportation problems present a general structural framework, as follows:

- A complex system (e.g., traffic network, transportation management system, or control system) composed of a set of interactive subsystems is in need of management.
- The system seeks to perform a global, or network-wide, improvement through synthesizing and guiding their cooperation in the distributed subsystems.
- The distributed subsystems are modeled as agents. These agents have their own local knowledge and rules for responding to stimuli, and they have logic to cooperate with their neighborhood agents in formulating their local strategies. Agents have abilities to achieve cooperation and acquire learning. The learning and cooperation mechanisms usually are key components in MAS.
- The agents submit their local plans and strategies to the master system processor, which coordinates all subsystems' plans to ensure a certain (acceptable) level of the global performance of the system. This process usually is modeled through a knowledge base. Learning and cooperation between the master system processor and subsystems could also be realized by agents through protocols.
- The system monitors the performance of each subsystem and assesses all proposals put forward by subsystem agents. It could achieve the overall improved performance by coordinating all subsystems.

It should be noted that many MAS applied in the transportation domain and reviewed in this chapter have been developed outside of the United States. The level of complexity of MAS applications in the transportation domain also reviewed in this chapter appears far less frequently than the agent-based transportation platforms discussed in chapter 4.

MAS Applied in Traffic Management

Traffic management herein refers to a management framework, rather than a comprehensive system with its complete subcomponents. *Signal control* and *route choice* components could be ingredients of a management framework but are discussed separately because of their complexities. A selection of MAS applications in traffic management are highlighted in the following examples.

Cooperative Traffic Management and Route Guidance System (CTMRGS), USA[118,140]

CTMRGS is a cooperative, distributed MAS that assists in the improvement of dynamic routing and traffic management. Agents represent both individual drivers and the system operator. Allocation of network capacity and distribution of traffic advisories are performed by agents that act on behalf of information service providers (ISP). Drivers are ultimately responsible for making travel choices, but ISPs provide advice so that the two entities behave cooperatively, satisfying their own objectives simultaneously. Such a negotiation between an ISP and driver agents seeks a more efficient route allocation across time and space. Drivers' route choice behavior utilizes a knowledge base. An ISP uses a set of utility functions to evaluate the route proposals put forward by drivers.

Tomas and Garcia,[149] Spain

Tomas and Garcia[149] conducted research to study the incident management plan. When an incident is detected offline, a set of traffic management strategies is developed. The implementation of these strategies usually involves negotiations among several traffic administrations. A set of agents, who represent different traffic management operators, share information to produce a knowledge base and communicate with each other to produce a strategy for managing an offline incident scenario on a non-urban road.

TRYS/TRYSA2 (Tráfico, Razonamiento y Simulación/Tráfico, Razonamiento y Simulación Autonomous Agents), Spain[142]

TRYS/TRYSA2 is an agent-based architecture for intelligent traffic management systems. A set of traffic management operators is represented and modeled by agents; proposals of agents are modeled as knowledge-based, or rule-based. Management plan proposals are put forward by different agents who can negotiate and coordinate through heuristic-based artificial intelligent algorithms. The system is reported to support real-time traffic management in the urban motorway network in Barcelona.

CLAIRE, France[159]

CLAIRE is a traffic management system based on Automatic Control and AI. Congestions of the system could be ameliorated by traffic engineering methods, modeled as an operator agent, to propose congestion-mitigation strategies. Proposals of an operator are modeled by knowledge-based AI methods.

CARTESIUS (Coordinated Adaptive Real-Time Expert System for Incident Management in Urban Systems), Germany[148]

CARTESIUS is a multiagent architecture for the provision of real-time decision support to a traffic operations center for coordinated, interjurisdictional traffic congestion management on freeway and arterial networks. CARTESIUS is composed of two interacting knowledge-based systems that perform cooperative reasoning and resolve conflicts for the analysis of nonrecurring congestion and the online formulation of integrated control plans.

MAS Applied in Dynamic Route Guidance

The following examples outline applications of MAS in dynamic route guidance.

TRACK-R (TRaffic Agent City for Knowledge-based Recommendation), Spain[160]

TRACK-R is an agent aiming to generate and sort possible routes to determine the optimum route for a car driver going from one city to another. To generate this information, the TRACK-R agent infers a knowledge base, composed by a partial instantiation of a traffic ontology. Every TRACK-R agent is responsible for a geographical area. If the network involves different areas but with shared elements, the related TRACK-R agents will communicate with each other to achieve a joint recommendation.

Dia,[145] Australia

This study by Dia proposed an agent-based method to model individual driver behavior when subject to the influence of real-time traffic information. A set of survey data was used to calibrate the multinomial logit model for en route quantitative delay information. Several other multinomial logit models were also developed. The results were used to identify the relevant factors and their suitable value for implementation in the agent-based behavioral models. The driver–vehicle units were modeled as autonomous software components that can each be assigned a set of goals to achieve and a database of knowledge computing preferences concerning the driving task.

Bazzan et al.[161] and Wahle et al.,[146] Germany

Studies conducted by Bazzan et al.[161] and Wahle et al.[146] modeled the impact of real-time information on traffic patterns by using an agent-based model, with special attention to investigating different types of information and their specific effects on traffic patterns. Each driver is an agent, characterized by its goals, resources, and behavior. In the proposed architecture, drivers' behavior is described based on BDI. Traffic models are modeled by a standard cellular automata method.

MAS Applied in Signal Control

The following examples highlight applications of MAS in signal control.

Agent-Based Dynamic Activity Planning and Travel Scheduling (aDAPTS), USA[114,162]

aDAPTS uses a hierarchical architecture that was developed for intelligent control systems to divide an agent-based control system's structure into three levels: organization, coordination, and execution. A global traffic operation center develops and maintains various control agents for interactive traffic control, road incident detection, and other transportation activities. The agent organization level mainly performs reasoning and planning for task sequences and organizes control agents to achieve specified goals. The agent coordination level is the interface between the organization and execution levels. The agent execution level consists of hardware and software units for deploying, replacing, hosting, and running control agents. Generally this level consists of many field-programmable and configurable devices and is distributed among local area network-linked local systems connected by wide area networks.

HUTSIG, Finland[136]

Developed by the Helsinki University of Technology (HUT), the HUTSIG system is incorporated in a microsimulator called

HUTSIM. Each signal operates individually as an agent in HUTSIM, negotiating with its neighborhood signals about the control strategy. The decisionmaking of the agents is based on fuzzy inference that allows a combination of various aspects like fluency, economy, environment, and safety. Area signal control stands on top of individual signals with the goal of minimizing overall delay, which requires cooperation between individual controllers to achieve better performance in the area.

Choy et al.,[133] Singapore

Choy et al.[133] introduced a multiagent architecture for real-time coordinated signal control in an urban traffic network. The multiagent architecture consists of three hierarchical layers of controller agents: intersection, zone, and regional controllers. Each controller agent is implemented by applying AI concepts (e.g., fuzzy logic, neural network, and evolutionary algorithm). With the fuzzy rule as a base, each individual controller agent recommends an appropriate signal policy at the end of each signal phase. An online reinforcement learning module is used to update the knowledge base and inference rules of the agents.

Botelho,[131] Portugal

This study introduced an interaction control structure with respect to the agents in a traffic-monitoring MAS. The goals of the agent are acquired by three mechanisms: an agent's innate goals, reception of requests in interagent communication, and subgoaling. Agents do not have the same goals irrespective of their current contexts. The system applies conditional goals to build agents with context-dependent goals.

Summary

With respect to *MAS,* as applied to transportation problems, the review in the prior sections summarizes the following propositions:

- In most of the reviewed transportation applications, MAS is a system method in the paradigm of AI and DAI.
- MAS is a useful tool to model interaction, coordination, and learning of a set of interactive subsystems; it is a decentralized method that simplifies dynamics among subsystems.
- To achieve a global or network-wide objective of the transportation system, MAS could be applied in a hierarchical structure. The higher level agent (system) investigates the performance of a group (subsystems) and has the ability, through agent coordination, to adjust the goals or actions of agents in its group. The global knowledge that informs and activates the coordination mechanism could be modeled by a knowledge base.
- The intelligence of an agent (subsystem) usually is modeled by AI technologies (e.g., rule-based, knowledge-based, or fuzzy rules).
- Because most applied MAS for transportation problems use methods of DAI, they may have less consideration to human behavior (i.e., an agent is a decentralized subsystem rather than an individual traveler).

CHAPTER 6
Agent-Based Modeling for Route Choice Behaviors—An Illustrative Example

Motivation

Travel behavior is an important component, and perhaps the most complex factor, contributing to the high complexity of a transportation system. The author's motivation here is to show that it is viable to use a behavior-based ABMS approach to study transportation in a bottom-up manner, rather than using the traditional top-down methods, which lack sufficient understanding of underlying behavioral factors. The shortcoming of a top-down method is in always providing a scenario-specific indicator. A change in the studied scenario usually requires establishing a new top-down method, with appropriate assumption, from scratch. A bottom-up approach has the flexibility to apply in a variant scenario, is capable of predicting the system performance under presently non-existing scenarios, and can possibly observe emergent behavior as a stimulus to a new environment setup. This is because it captures the underlying interacting and evolutionary mechanisms in a complex system. In summary, the traditional top-down approach studies *what* is the performance of a complex transportation system, whereas the bottom-up ABMS approach tries to understand *why* travelers make those decisions and *how* does the transportation system perform in such a circumstance.

Understanding traveler behavior is one of the important studies with respect to the transportation system. Traveler behavior can be divided into two parts: *before a trip* (pre-planned) and *within a trip* (en route decisions). The before-trip behavior mainly refers to route choice, and this topic is well studied by the activity-based travel demand models tied with an ABMS (see agent-based transportation platforms in chapter 4). In general, the route choice behavior may change as time elapses, because of the interactions between travelers as well as sudden changes in the transportation network topology and performance (e.g., due to an incident). In addition, travelers' route choice behavior involves learning from previous experiences, heterogeneity of travelers, incomplete network information, and communications among travelers. Those behaviors, which are not viable to model through the conventional equilibrium method or discrete choice models, could be tackled by ABMS. In the next section, the authors use a simple example to demonstrate this route choice model framework, which is implemented in AnyLogic software and tested with two simulation experiments. Results from numeric examples are compared with the classical network equilibrium solutions. The goal is to exhibit how network topology changes can influence the traveler's decisionmaking in an ABMS framework, which could lead to results similar to those of the classical model as reported in the literature. This agent-based model provides an example to show the possibility of studying and understanding the travelers' complex decisionmaking under a wide variety of scenarios.

One traditional benchmark of a traveler's route choice criterion is the user equilibrium (UE) principle, in which a traveler chooses a route so as to minimize his or her travel time, and all used routes have equal and minimal travel time.[163] This behavior at the individual level creates equilibrium at the system (or network) level. Deterministic UE assumes that all travelers are homogeneous, that they have full perception of the network, and that they always choose routes with the lowest cost. Boundedly rational UE assumes travelers have full perception of the network but that they may choose a route with a higher travel time within a boundary.[164] In contrast, Stochastic UE assumes travelers have perception errors and that they make route choice decisions based on their perceived travel time.[165] Discrete choice models are often used to depict the heterogeneity. DTA considers time variations in a traffic network, which assumes that travel times on links vary over time. The UE condition therefore only applies to the same departure time interval between the same O-D pair.[166] This extension could analyze phenomena such as peak-hour congestion or time-varying tolls.[167] In recent years, owing to the continuously increasing computer power, researchers have been able to simulate an individual traveler's behavior in a large transportation network. Such applications include MIcroscopic Traffic SIMulator (MITSIM),[168] DYNASMART,[169] and DynusT,[91] in which either microscopic or mesoscopic simulators are embedded. Those studies focus more on *how* the travelers make their decisions rather than *why* the travelers make such decisions.

Although disaggregated travel demand models and microscopic traffic simulation models have been applied to modeling the route choice behavior in an integrated simulation environment,[167–168] it is difficult to model the information-sharing among travelers, the interactions among travelers, and the changes to the transportation network by using traditional non-agent based modeling schemes. Agent-based modeling was specifically developed to address this complexity and to support individual decisionmaking. ABMS has been widely implemented in many areas; however, as discussed in chapter 5, those studies with the subject termed as *MAS* come mainly under the umbrella of the computational method of AI and DAI. Differentiating from those studies, ABMS demonstrated in this chapter explicitly models individual-based traveler's route choice behaviors, with an emphasis on the capability of the effects of learning and interaction.

In summary, the strengths and benefits of integrating ABMS to study travelers' route choice behavior, rather than the traditional route choice models, include the ability to:

- Capture an individual traveler's rational and irrational behavior and preferences that are difficult to quantify or measure in the traditional route choice models.
- Recognize and consider that travelers have different socioeconomic properties, travel habits, preferences, manner of reaction to the en route information, etc., and thus exhibit heterogeneity.
- Capture the interaction effects and collective behavior that stem from travelers' heterogeneity.
- Recognize and consider that travelers may have limited knowledge about factors such as traffic conditions, incidents, and weather conditions; hence, ABMS captures the vagueness of driver behavior, unlike discrete choice models that assume drivers are always rational with perfect access to full information.

- Enable travelers' decision and knowledge (learning) to be updated on a real-time basis, rather than on a day-to-day basis.
- Formulate the mechanism of travelers' complex decisionmaking process.
- Predict travelers' reasonable response to real-time en route information or similar unexpected stimulus (sudden change of network) imposed in the environment.
- Possibly observe emergent behavior as a stimulus to a new environment setup.

An Example Applying ABMS Model to Route Choice Behavior Model

Travelers are modeled as agents, who choose a route based on their knowledge of the network prior to each trip (en route choice is not considered in this example). In the route choice model, a traveler agent first decides which route to travel when their trip starts. The traveler could decide to stay on the same route as the previous trip or could decide to change to an alternative route. At first, the traveler may have little or no information about which is the best route, but experience can help the traveler find his or her best route. In this example, *best* is based on travel time. Travelers might not have sufficient incentive to change routes if their experienced travel time is close enough to their perceived minimum travel time. If, however, they experience a travel time that is sufficiently different from their expectation, they will consider changing routes.

The following rules are developed to mimic the behavior of an agent who is considering a route change. Suppose TT_j^n is the experienced travel time of jth route on nth day (that means a traveler agent chooses route j on the nth day) and TT_{min}^n is the traveler's perceived minimum travel time on the nth day. It is reasonable to assume that a traveler agent may know the travel time only for the route he or she has experienced; thus, the perceived minimum travel time may not be the actual minimum travel time. An initial travel time, which reflects the agent's expectation of each route, is assigned to every traveler agent before the first trip. If the travel time of a route is not observed in a certain trip, the traveler agent uses previously experienced travel times of the route or the initial travel time if the route has never been chosen before, to determine TT_{min}^n. The traveler agent updates the travel time only for the selected routes, while leaving those of other routes unchanged. These rules are stated as follows:

RULE 1: If $(TT_j^n = TT_{min}^n)$, then the traveler agent does not change route or n + 1th day.

RULE 2: If $(TT_j^n - TT_{min}^n) \leq \varepsilon$, then the traveler agent does not change route on n + 1th day, where ε is a threshold related to the perception error.

RULE 3: If $(TT_j^n - TT_{min}^n) > \varepsilon$, then the traveler agent changes route with probability $(TT_j^n - TT_{min}^n)/ TT_j^n$ and the choice probability is based on the posterior probability given the route choice and previously experienced travel time.

RULE 1 represents the case when traveler agents are already travelling on the route that corresponds to their perceived minimum travel time; hence, they do not change routes. RULE 2 represents the case when the travel time of the current route is very close to the perceived minimum travel time; hence, the traveler agents will maintain their original choice. RULE 3 represents the situation when traveler agents might change their routes, and the route change probability is related to the difference between the experienced travel time and the perceived minimum travel time. The larger the difference, the higher the route changes probability.

When a traveler agent decides to change routes, a decision must be made on which route to choose. This primer considers the learning process of an agent and the heterogeneity of different travelers.

The learning process details how agents make route choice decisions based on their previous experiences. It can be characterized as *Bayesian learning*.[170] For each traveler agent, the prior probability represents the subjective probability (traveler's belief) that one route takes the minimum travel time. Data is based on the experience of the traveler and the perceived minimum travel time. The corresponding posterior subjective probability is updated based on the prior subjective probability and the data.

For each O-D pair, suppose p_{jn} denotes the subjective probability that the jth route takes the minimum travel time on the nth day and $\mathbf{p}_n$ denotes the vector of subjective probabilities. d_{jn} is a data variable that equals 1 if the traveler perceives that the jth route takes the minimum travel time (TT_{min}^n) on the nth day, and 0 otherwise. $\mathbf{d}_n$ is the vector of minimum travel time variables. Based on Bayes' theorem, the posterior distribution can be expressed as:

$$f\ \mathbf{p}_{n+1}\ = f(\mathbf{p}_n|\mathbf{d}_n) \propto g(\mathbf{d}_n|\mathbf{p}_n)f(\mathbf{p}_n) \qquad (1)$$

The route choice given the subjective probability $g(\mathbf{d}_n|\mathbf{p}_n)$ follows a multinomial distribution with trial number one. The probability mass function of $\mathbf{d}_n$ is:

$$g(\mathbf{d}_n|\mathbf{p}_n) = \prod_{j\in J} {p_{jn}}^{d_{jn}} \qquad (2)$$

where j is the total number of routes between this O-D pair. Because only one route is chosen by each traveler agent, $\sum_{j\in J} d_{jn} = 1$.

The authors assume that the prior distribution $f(\ \mathbf{p}_n\ )$ follows a Dirichlet distribution with parameter set $\boldsymbol{\alpha}_n = (\alpha_{1n}, \alpha_{2n}, \ldots, \alpha_{jn})$. Because the Dirichlet distribution is the conjugate prior of the parameters of the multinomial distribution, the posterior distribution will also be a Dirichlet distribution with parameter set $\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \mathbf{d}_n$. The probability density function of Dirichlet distribution is defined by:

$$f(\boldsymbol{p}_n) = \frac{1}{B(\boldsymbol{\alpha}_n)} \prod_{j\in J} {p_{jn}}^{\alpha_{jn}-1} \qquad (3)$$

In practice, the mean of each random variable $E\ p_{jn}\ = \alpha_{jn}/\alpha_{0n}$ is used to represent the subjective probability of the ith route on the nth day, where $\alpha_{0n} = \sum_{j\in J} \alpha_{jn}$. For example, suppose on the nth day, the i^{th} route takes the perceived minimum travel time, the posterior subjective probability on the n+1th day can be updated as:

$$p_{in+1} = (\alpha_{in} + 1)/(\alpha_{0n} + 1) \quad i \in J \qquad (4)$$

$$p_{jn+1} = \alpha_{jn}/(\alpha_{0n} + 1) \quad j \in J, j \neq i \qquad (5)$$

To provide the parameters for the first day, an initial parameter vector of Dirichlet distribution needs to be given. Then, the Bayesian learning can be repeated iteratively.

This model assumes that all agents use the same Bayesian-updating scheme, but each agent has his or her own perception error on the experienced travel time. Suppose the parameter of Dirichlet distribution has an error term γ_{jn} (jth route on nth day), which stands for the perception error. Then the parameter can be expressed as $\alpha_{jn} + \gamma_{jn}$, where α_{jn} is the deterministic part and γ_{jn} is the random part. Assume γ_{jn} follows a Gumbel distribution with parameter θ, then the choice probability is given by a multinomial logit model:[171]

$$p_{jn+1} = \frac{e^{\theta(\alpha_{jn}+d_{jn})}}{\sum_{j' \epsilon J} e^{\theta(\alpha_{j'n}+d_{j'n})}} \qquad (6)$$

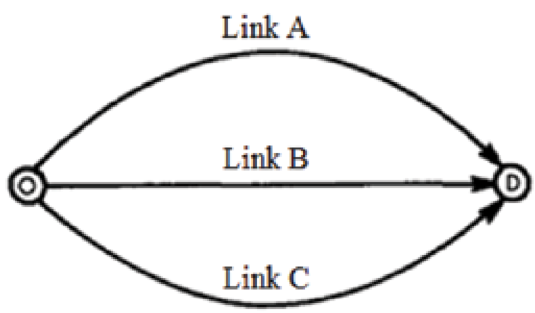
Besides his or her own experience, a traveler agent may also, from time to time, acquire the network information (travel time) from other traveler agents or from the environment. The *environment* here refers to media such as radio and Internet, from which the traveler agent could get travel time information about all routes in the network. In this model, it is assumed that 1 percent of the total agents (called *communicating agents*) who are randomly selected are given the actual travel time information with respect to all routes in the network. This modification speeds up the convergence rate because the communicating agents tend to make more rational decisions.

Experiment Design

Three experiments were designed and conducted to test and validate the aforementioned proposed model. The first experiment used a simple network to show that the agent-based route choice model is able to reach the same equilibrium solution as obtained from classical traffic assignment models. In the second experiment, the proposed model showed how changes in network topology influence the agents' decisions and how the traveler agents adapt to the new network and form a new traffic pattern. The goal of the third experiment was to test the influence of communicating agents.

A simple network is shown in figure 9. The network has only one O-D pair with three different routes (links). The capacity of each of the three differing routes is 200 vehicles, 400 vehicles, and 300 vehicles, and the free flow travel time of each route is 10 minutes, 20 minutes, and 25 minutes, respectively. The total flow between this O-D pair is 1,000 vehicles. Initial travel times for the first iteration of the experiment were calculated by the Bureau of Public Roads (BPR) function. The network configuration is the same as in the sample network used in Sheffi;[172] thus, the results can be compared with the results obtained by the classical UE models.

The model was implemented in AnyLogic simulation software. In this experiment, the number of traveler agents was 1,000, which is equal to the total O-D flow. Each traveler agent made a route choice every iteration and updated his or her choice probability based on the rules described in the previous section. The initial parameters of the Dirichlet distribution were set to be $\alpha_1 = (1,1,1)$. As a result, the choice probability was 1/3 for each route, which suggests that the traveler agents did not have any preference on the routes initially. Each iteration was equal to 30 simulation time units, which means that the travel time and flow were updated every 30 time units.



$$f_1 + f_2 + f_3 = 1000$$

Figure 9. Simple network with three links.

Experimental Results

Comparison of Agent-Based Model and Classical Route Choice Results

In figure 10, the time plots of flow (A) and travel time (B) on three routes are shown. In figure 10, it is revealed in (A) that the flows on the three routes fluctuated for the first several iterations and then very quickly became stable. It is revealed in (B) that the travel times of the three routes converged to a single value (with travel time of ~25.4 min), which is exactly the same UE point calculated by the Frank–Wolfe Algorithm using the convex combination method found in classical traffic assignment models.

Behavioral Evolution Exhibited in the Agent-Based Route Choice Model

In this experiment, the microscopic simulation was incorporated to obtain travel time instead of BPR function. This microscopic model is characterized by a car-following model and a lane-changing model mainly derived from the Next-Generation Simulation program models. The car-following model is based on Newell's[173] piecewise linear car-following model, with additional considerations such as maximum acceleration, maximum deceleration, travel distance under free flow speed, and safety constraints.[174]

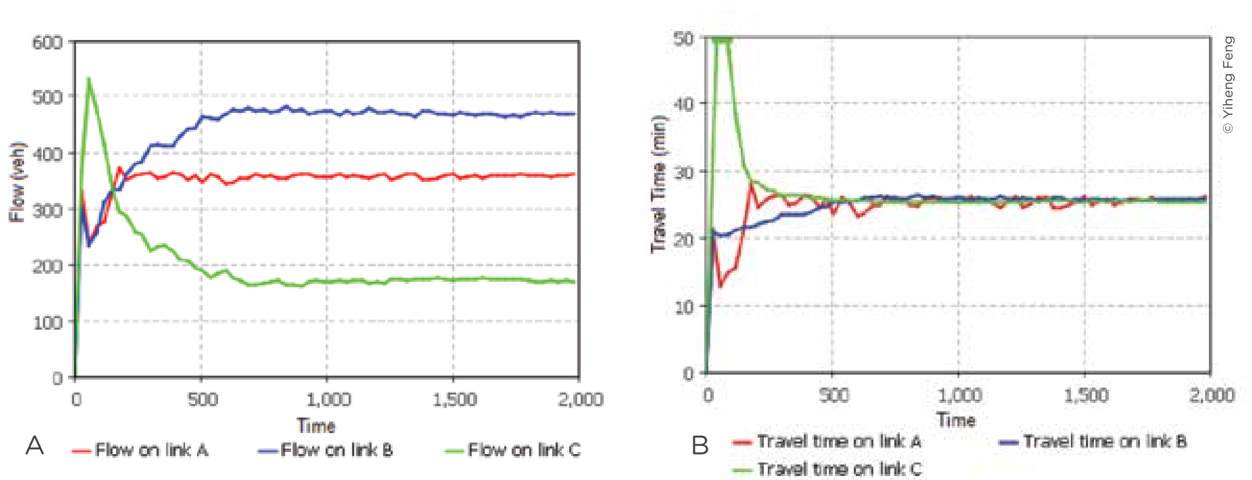


Figure 10. Time plots of flow and travel time (Experiment I).
NOTE: One iteration consists of 30 time units.

The lane-changing model consists of two levels of decision: lane-changing choice model[175] and gap-acceptance model.[174] The lane-changing choice model calculates the probability of whether to change lanes. The changing probability is dependent on the speed differences between the current vehicle and its lead vehicle. The gap-acceptance model calculates the necessary lead and lag gap in the target lane for lane changing. If both gaps are satisfied, the vehicle will perform lane changing. Each vehicle can only change to its immediate adjacent lane in one simulation step.

The same network was used in this experiment as in the first experiment, with one O-D pair and three routes. Different routes had different lengths and different free flow speeds. The total number of traveler agents was reduced to 500, but use of this lower number corresponded to a more realistic travel time, because vehicle interactions affect the travel time. Once the route choice was made, the traveler agents were loaded into the network from a virtual queue at the entrance of each route. Iterations ended when all traveler agents finished their trips. At the end of each iteration, the average travel time of all traveler agents and the flows on each route were recorded. Finally, all traveler agents updated their choice probability and made a route choice before the next iteration.

Experiment II was designed to study travelers' behavioral responses because of a network topology change. A scenario demonstrating the process is illustrated as follows: Routes B and C have three lanes, whereas Route A has two lanes at the beginning of simulation. It is assumed that at the 50th iteration, the government agency decides to expand the capacity of Route A by adding an additional lane. Adding one new lane requires construction work, which lasts a certain amount of time (assumed to be 50 iterations). During the construction period, the capacity of Route A is reduced to half. After completion of the construction, traveler agents can choose among the three lanes with the same probability.

The flow and average travel time on each link is shown in figure 11 and figure 12. The horizontal axis represents the number of iterations. The flows on each route gradually became stable after a certain amount of time, as shown in figure 11. At the 50th iteration, there was an abrupt drop in the flow on Route A, which indicates that the number of available lanes was changed to one, and the travel time of Route A in figure 12 was increased suddenly. Meanwhile, in figure 12 it is revealed that the travel time of Route A between the 50th iteration and 100th iteration varied more severely, because when the capacity was decreased, the travel time was more sensitive to the vehicle interactions captured by car-following and lane changing. After the 100th iteration, construction was complete, and the number of lanes of Route A was increased to three. Therefore, the travel time on Route A was decreased, and the flow of Route A starts to increase gradually. It took about 40 iterations before the flows

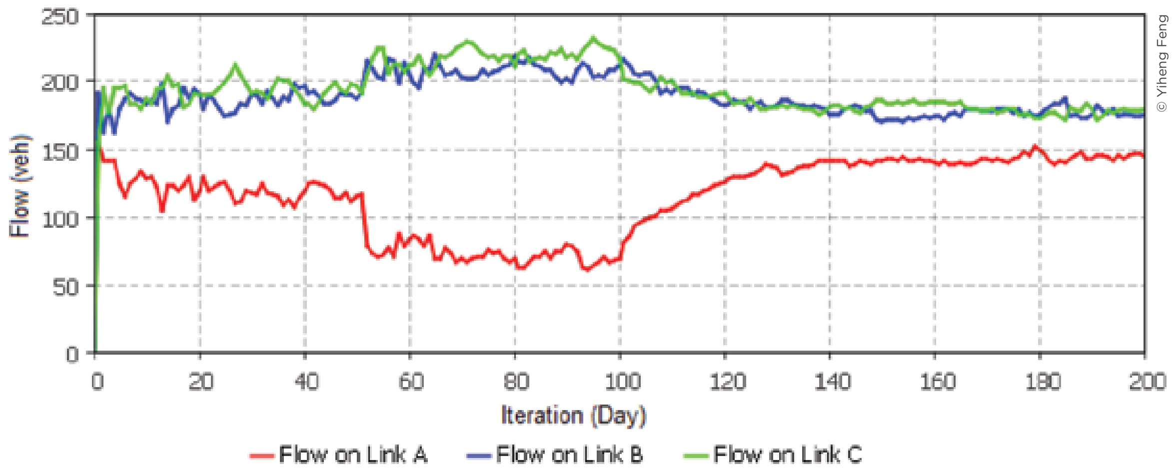


Figure 11. Flows on each route (Experiment II).

on each link became stable and the travel time converged to a single value. After about the 140th iteration, a new traffic pattern was formed.

As shown in figure 11, flows dropped fast (only a few iterations) when one lane was closed on Route A; however, it took a longer time for flows to recover to a steady value (~40 iterations) when the blocked lane re-opened and the new lane was made available. Travelers on Route A immediately recognized the sudden delay because of blockage, and because the extra delay was much higher than the risk tolerance (parameter in the model), this triggered the route choice mechanism in the agent-based model

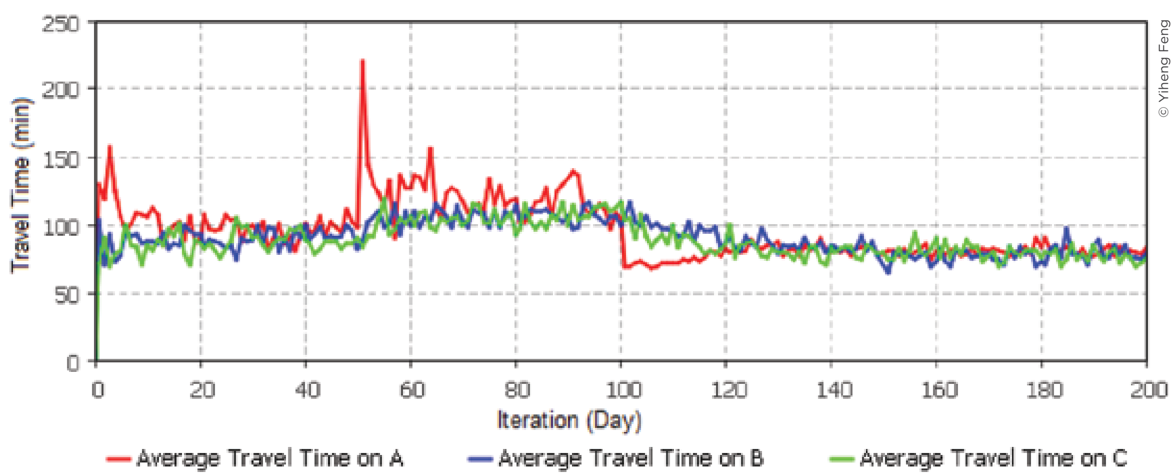


Figure 12. Travel times on each route (Experiment II).

with high probability. As a result, flows on Route A dropped quickly because of diversion to alternative Routes B and C. When the capacity of Route A was recovered, traveler agents on Routes B and C had difficulty detecting the recovery of Route A, because they had only partial network information (except for those communicating agents). Those travelers still believed that the travel time on Route A was high, until they happened to randomly experience Route A sometime later; however, the probability of changing routes for traveler agents in Routes B and C at an equilibrium status was rather low. For a different reason, traveler agents already in Route A did not change their routes either, because they were now experiencing a lower travel time. As a consequence, the recovery process was slow. The result is somewhat consistent with similar experiences in a real-world situation, that is, people are more likely to change decisions when experiencing a worse situation but are less likely to change decisions for a better solution—particularly if, because of partial knowledge of the network, the better situation is not obvious. In economics and decision theory, this finding is called *loss aversion*, which means losses and disadvantages have a greater impact on preferences than do gains and advantages.[176]

In Experiment III, 10 percent of agents were randomly chosen to be communicating agents (compared with 1 percent in the previous experiment). Communicating agents were aware of 50 percent of the travel times in other parts of the network. That is, if a communicating agent chose Route A, he or she only randomly knew the additional information of Route B or C with 0.5 probabilities, respectively. The same rule was applied to communicating agents who chose Route B or C. The simulation results are shown in figure 13 and figure 14.

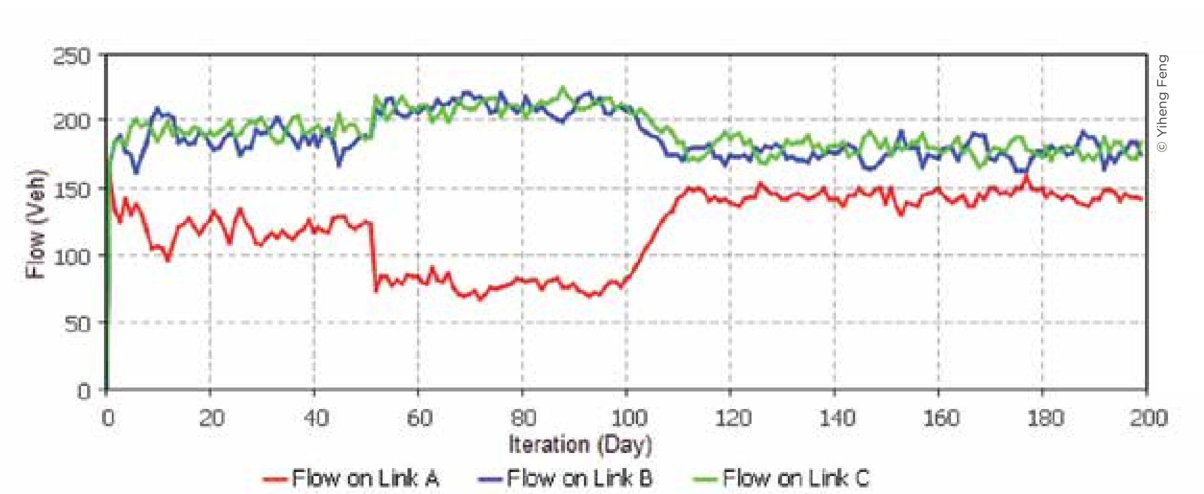


Figure 13. Flows on each route (Experiment III).

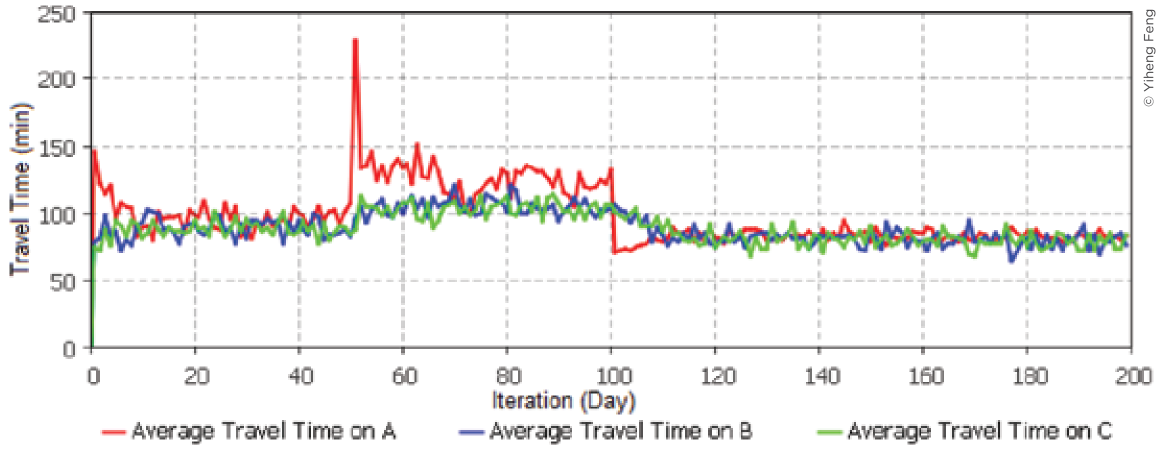


Figure 14. Travel times on each route (Experiment III).

Compared with Experiment II, the convergence speed after the construction was much faster in Experiment III. It only took about 13 iterations to converge to a new traffic pattern. Although communicating agents had only partial information about half of the other route travel times, the number of communicating agents was increased 10 fold. Overall, the real-time network information acquired in each iteration for all agents was increased.

Concluding Remarks

For this section, the authors presented an agent-based simulation model exhibiting travelers' route choice behavior. The route choice model considers a traveler's learning from previous experiences, heterogeneity of travelers, partial network information, and communication between travelers and the environment. The proposed model has been implemented in AnyLogic agent-based simulation software. Two experiments were conducted to examine the behavioral characteristics exposed by the model. In the first experiment, the proposed agent-based route choice model reached the same UE solution as reported in classical models in the literature. The second experiment successfully demonstrated how a network topology change influenced the traveler's behaviors and how traveler agents adapted to the new network to form a new traffic pattern.

The authors use this trial example to demonstrate the capability of an agent-based model in studying a transportation system if the agent-based model is armed with a well-defined travelers' behavior component. The example not only successfully replicates the overall performance that the traditional method can accomplish but also provides extra behavioral insights that demonstrate the day-to-day equilibrium process. The behavioral mechanisms of an agent-based route choice model could be flexibly applied in other scenarios to predict the network performance, which is typically not within the classical approach's reach. This agent-based modeling paradigm opens the possibility of studying and understanding the complexity of travelers' decisionmaking under a wide variety of scenarios. The flexibility and extensibility of agent-based modeling allows for the analysis of more complex human behaviors in future work. For example, travel time may not be the only criterion for route choice. Besides, more realistic human decisionmaking models, such as extended BDI, can be employed to mimic travelers' route-selecting process.[25] This paradigm is expected to be deployed to analyze a real-world transportation network with real traffic data.

Conclusions

Conclusions

ABMS has been widely applied in a spectrum of disciplines, including, but not limited to, ecology, biology, business, economic science, computer simulation, social sciences, political science, policy, and military studies. The cognition and knowledge of ABMS and the recognition of applications continue to expand in step with its rapid development.

This primer reviews and summarizes the ABMS approaches that have been studied in the transportation paradigm in the past few decades and thus presents and depicts the concept of ABMS as scoped in the literature. Those applications fall primarily into two methodological domains: individual-based models that study personal transportation-related activities and behavior, and system and computational methods, known as *MAS,* to study a collaborative and reactive transportation system by modeling autonomous decisionmaking by a collection of subsystem entities called *agents*. In a nontrivial review effort, the authors offer the summary that the former is closely related to models for activity-based travel demand and land use, whereas the latter is typically scoped as a powerful technique for simulating dynamic complex systems to observe emergent behavior.

Another goal of this primer is to promote the understanding that the traditional transportation behavioral models could be viable for use within ABMS. This effort is demonstrated in the travelers' route choice decisionmaking process in this primer. In chapter 6, a behavioral model is established in a bottom-up framework and tries to formulate the mechanism of a traveler's complex route choice behavioral process as a collaborative and reactive result of users' mindset and the network environment. The authors hope that this ABMS modeling paradigm demonstrates that one can flexibly predict travelers' behavioral actions in response to real-time information and sudden changes in the network environment and that it is plausible to observe emergent behavior as a stimulus to a new environment setup.

Acknowledgements

Acknowledgements

This research is supported by the Federal Highway Administration's Exploratory Advanced Research Program project titled "Agent-Based Modeling and Simulation for Transportation, VASTO: Evolutionary Agent System for Transportation Outlook." The authors are grateful for the guidance, comments, and support from the following peer exchange panel members:

Sanjay Jain, George Washington University
Jay Jayakrishnan, University of California, Irvine
Ram Pendyala, Arizona State University
Chang-Tien Lu, Virginia Tech
Sarath Joshua, Maricopa Association of Governments

References

References

1. Samuelson, D. A., & Macal, C. M. (2006). Agent-based simulation comes of age: Software opens up many new areas of application. *OR/MS Today*, 33(4).
2. Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *PANS*, 99, 7280–7287.
3. Gilbert, N. (2007). *Agent-based models*. Thousand Oaks, CA: SAGE Publications, Inc.
4. Macal, C. M., & North, M. J. (2006b). *Tutorial on agent-based modeling and simulation Part 2: How to model with agents.* Proceedings of the 2006 Winter Simulation Conference, Monterey, CA.
5. Davidsson, P. (2002). Agent-based social simulation: A computer science view. *Journal of Artificial Societies and Social Simulation*, 5(1).
6. Niazi, M., & Hussain, A. (2011). Agent-based computing from multi-agent systems to agent-based models: A visual survey. *Scientometrics*, 89(2), 479–499.
7. Gardner, M. (1970). The fantastic combination of John Conway's new solitaire game "Life." *Scientific American*, 223, 120–123.
8. Macal, C. M., & North, M. J. (2009). *Agent-based modeling and simulation*. Proceedings of the 2009 Winter Simulation Conference, Austin, TX.
9. Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: Social science from the bottom up*: MIT Press.
10. Heath, B. L. (2010). *The history, philosophy, and practice of agent-based modeling and the development of the conceptual model for simulation diagram*. Unpublished doctoral thesis, Wright State University, Dayton, OH.
11. Macal, C. M., & North, M. J. (2006a). *Tutorial on agent-based modeling and simulation Part 2: How to model with agents.* Paper presented at the Winter Simulation Conference, Monterey, CA.
12. Samuelson, D. A. (2005). Agent of change: How agent-based modeling may transform social science. *OR/MS Today*, 32(1).
13. Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence, 117*(2), 277–296.
14. Hughes, H. P. N., Clegg, C. W., Robinson, M. A., & Crowder, R. M. (2012). Agent-based modelling and simulation: The potential contribution to organizational psychology. *Journal of Occupational and Organizational Psychology*, 85(3), 487–502.
15. Crowder, R. M., Robinson, M. A., Hughes, H. P. N., & Sim, Y.-W. (2012). The development of an agent-based modeling framework for simulating engineering team work. *IEEE Transactions on systems, man, and cybernetics, Part A: Systems and humans,* 42(6), 1425–1439.
16. Macal, C. M. (2003). *Effects of global information availability in networks of supply chain agents.* Paper presented at the Agent 2003 Conference on Challenges in Social Simulation, Chicago, IL.
17. Said, L. B., Bouron, T., & Drogoul, A. (2002). *Agent-based interaction analysis of consumer behavior*. Paper presented at the Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, Bologna, Italy.

18. Singer, H. M., Singer, I., & Herrmann, H. J. (2009). Agent-based model for friendship in social networks. *Physical Review E80*, 026113.
19. Klugi, F., Bazzan, A., & Ossowski, S. (Eds.). (2005). *Applications of agent technology in traffic and transportation*. Basel, Switzerland: Birkhäuser Verlag.
20. Aschwanden, G. D. P. A., Wullschleger, T., Muller, H., & Schmitt, G. (2012). Agent-based evaluation of dynamic city models: A combination of human decision processes and an emission model for transportation based on acceleration and instantaneous speed. *Automation in Construction,* 22, 81–89.
21. Caplat, P., Anand, M., & Bauch, B. (2008). Symmetric competition causes population oscillations in an individual-based model of forest dynamics. *Ecological Modelling*, 211(3–4), 491–500.
22. Perez, L., & Dragicevic, S. (2009). An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics*, 8(50).
23. Heckbert, S. (2012). MayaSim: An agent-based model of the ancient Maya social–ecological system (Version 3). Retrived August 14, 2013, from http://hdl.handle.net/2286.0/oabm:3063
24. Bharathy, G. K., Yilmaz, L., & Tolk, A. (2012). Agent directed simulation for combat modeling and distributed simulation. In A. Tolk (Ed.), *Engineering principles of combat modeling and distributed simulation*, Hoboken, NJ: John Wiley & Sons, Inc.
25. Lee, S., Son, Y., & Jin, J. (2010). An integrated human decision making model for evacuation scenarios under a BDI framework. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 20*(4), Article 23.
26. Gibson, F. P., Fichman, M., & Plaut, D. C. (1997). Learning in dynamic decision tasks: Computational model and empirical evidence. *Organizational Behavior and Human Decision Processes, 71,* 1–35.
27. Mosteller, F., & Nogee, P. (1951). An experimental measurement of utility. *The Journal of Political Economy, 59*, 371–404.
28. Opaluch, J. J., & Segerson, K. (1989). Rational roots of irrational behavior: New theories of economic decision-making. *Northeastern Journal of Agricutural and Resource Economics, 18*(2), 81–95.
29. Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics, 69,* 99–118.
30. Busemeyer, J. R., & Diederich, A. (2002). Survey of decision field theory. *Mathematical Social Science, 43*(3), 345–370.
31. Edwards, W. (1954). The theory of decision-making. *Psychological Bulletin, 51*(4), 380–417.
32. Einhorn, H. J. (1970). The use of nonlinear, noncompensatory models in decision making. *Psychological Bulletin, 73*, 221–230.
33. Payne, J. W. (1982). Contingent decision behavior. *Psychological Bulletin, 92*, 382–402.
34. Konar, A., & Charkraborty, U. K. (2005). Reasoning and unsupervised learning in a fuzzy cognitive map. *Information Sciences, 170,* 419–441.
35. Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1–64.

36. Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
37. Rao, A. S., & Georgeff, M. P. (1998). Decision procedures for BDI logics. *Journal of Logic and Computation, 8*(3), 293–343.
38. Rothrock, L., & Yin, J. (2008). Integrating compensatory and noncompensatory decision making strategies in dynamic task environments. In T. Kugler, C. Smith, T. Connolly, & Y. Son (Eds.), *Decision modeling and behavior in uncertain and complex environments* (pp. 123–138). New York: Springer-Verlag.
39. Sirbiladez, G., & Gachechiladez, T. (2005). Restored fuzzy measures in expert decision-making. *Information Sciences, 169*, 71–95.
40. Zhao, X., & Son, Y. (2008). BDI-based human decision-making model in automated manufacturing systems. *International Journal of Modeling and Simulation, 28*(3), 347–356.
41. Norling, E. (2004). *Folk psychology for human modeling: Extending the BDI paradigm.* Paper presented at the Proceedings of International Conference on Autonomous Agents and Multi-Agent System, New York.
42. Kinny, D., Georgeff, M., & Rao, A. (1996). *A methodology and modeling technique for systems of BDI agents*. Paper presented at the Proceedings of the 7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands.
43. Lee, S., & Son, Y.-J. (2009). *Dynamic learning in human decision behavior under the extended belief–desire–intention framework*. Paper presented at the Proceedings of the 2009 Industrial Engineering Research Conference, Miami, FL.
44. Buntine, W. (1991). *Theory refinement on bayesian networks*. Paper presented at the 7th Conference on Uncertainty in Artificial Intelligence, Los Angeles, CA.
45 Heckerman, D., Geiger, D., & Chickering, D. (1994). *Learning bayesian networks: the combination of knowledge and statistical data*. Paper presented at the Proceedings of 10th Conference on Uncertainty in Artificial Intelligence, Seattle, WA.
46. Lam, W., & Bacchus, F. (1993). *Using causal information and local measures to learn bayesian networks*. Paper presented at the Proceedings of the 9th Conference on Uncertainty in Artifical Intelligence, Washington, DC.
47. Suzuki, J. (1993). *A construction of bayesian networks with discrete variables from data.* Paper presented at the Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence, Washington, DC.
48. Pearl, J., & Verma, T. S. (1991). *A theory of inferred causation.* Paper presented at the 2nd International Conference on Principles of Knowledge Representation and Reasoning. San Francisco, CA: Morgan Kaufmann.
49. Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search.* New York: Springer-Verlag.
50. Niculescu, R. S., Mitchell, T. M., & Rao, R. B. (2006). Bayesian network learning with parameter constraints. *Journal of Machine Learning Research, 7*, 1357–1383.
51. Djan-Sampson, P. O., & Sahin, F. (2004). *Structural learning of bayesian networks from complete data using the scatter search documents.* Paper presented at the IEEE International Conference on System, Man, and Cybernetics, The Hague, The Netherlands.

52. Fu, W., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General, 135*(2), 184–206.
53. Gray, W. D., Sims, C. R., Fu, W., & Schoelles, M. J. (2006). The soft contraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. *Psychological Review, 113*(3), 461–482.
54. Sun, R., Alusarz, P., & Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review, 112*(1), 159–192.
55. Sun, R., Merrill, E., & Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science, 25*(2), 203–244.
56. Holroyd, C. B., & Coles, M. G. H. (2002). The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review, 109*, 679–709.
57. Kim, S., Xi, H., Mungle, S., & Son, Y.-J. (2012). *Modeling human interactions with learning under the extended belief–desire–intention framework.* Paper presented at the Proceedings of the 2012 Industrial and Systems Engineering Research Conference, Orlando, FL.
58. Thomas, K. (1976). Conflict and conflict management. In M. D. Dunnette (Ed.), *Handbook of industrial and organizational psychology* (pp. 889–935). Chicago, IL: Rand McNally, Inc.
59. Elliott, E., & Kiel, L. D. (2002). Exploring cooperation and competition using agent-based modeling. *National Academy of Sciences of the United States of America, 99*(3), 7193–7194.
60. Danielson, P. (2002). Competition among cooperators: Altruism and reciprocity. *National Academy of Sciences of the United States of America, 99*(3), 7237–7242.
61. Macy, M. W., & Flache, A. (2002). Learning dynamics in social dilemmas. *National Academy of Sciences of the United States of America, 99*(3), 7229–7236.
62. Read, D. W. (2002). A multi-trajectory, competition model of emergent complexity in human social Organization. *National Academy of Sciences of the United States of America, 99*(10), 7251–7256.
63. Pan, X., Han, C. S., & Law, K. H. (2005). *A multi-agent based simulation framework for the study of human and social behavior in egress analysis.* Paper presented at the International Conference on Computing in Civil Engineering, Cancun, Mexico.
64. Zlotkin, G., & Rosenschein, J. S. (1991). Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(6), 1317–1324.
65. Allan, R. (2010). *Survey of agent based modelling and simulation tools.* Final report. Warrington, UK: Computational Science and Engineering Department, STFC Daresbury Laboratory.
66. Lytinen, S. L., & Railsback, S. F. (2009). *The evaluation of agent-based simulation platforms: A review of NetLogo 5.0 and ReLogo.* Paper presented at the European Meetings on Cybernetics and Systems Research.
67. Nikolai, C., & Madey, G. (2009). Tools of the trade: A survey of various agent based modeling platforms. *Journal of Aritificial Societies and Social Simulation, 12*(No. 22).

68. Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations. *Simulation, 82*(9), 609–623.
69. Serenko, A., & Detlor, B. (2002). Agent toolkits: A general overview of the market and an assessment of instructor satisfaction with utlizing toolkits in the classroom. *Research and Working Paper Series* (Vol. No. 455). Ontario, Canada: Michael G. DeGroote School of Business, McMaster University.
70. Macal, C. M., & North, M. (2008). *Agent-based modeling and simulation: ABMS examples*. Paper presented at the Winter Simulation Conference, Miami, FL.
71. Tisue, S., & Wilensky, U. (2004). *NetLogo: A simple environment for modeling complexity.* Paper presented at the International Conference on Complex Systems, Boston, MA.
72. Wilensky, U. (2011). *NetLogo 4.1.3 User Manual: Programming Guide.*
73. NetLogo Web Site. (2013). Retrieved August 15, 2013, from http://ccl.northwestern.edu/netlogo/models/
74. MASON Web site. (2012). Retrieved August 15, 2013, from http://cs.gmu.edu/~eclab/projects/mason/
75. Luke, S. (2011). *Multiagent simulation and the MASON library.* Final report. Retrieved on August 15, 2013, from http://cs.gmu.edu/~sean/temp/mason.pdf
76. Repast Web Site. (2012). Retrieved August 15, 2013, from http://repast.sourceforge.net/repast_3/index.html.
77. North, M. J., & Macal, C. M. (2011). *Product design patterns for agent-based modeling.* Paper presented at the Proceedings of the 2011 Winter Simulation Conference, Phoenix, AZ.
78. Ascape Web site. (2012). Retrived August 15, 2013, from http://ascape.sourceforge.net/index.html#Introduction.
79. AnyLogic Web site. (2013b). Overview. Retrieved August 15, 2013, from http://www.anylogic.com/overview.
80. Lee, S., Son, Y., & Jin, J. (2008). Decision field theory extensions for behavior modeling in dynamic environment using bayesian belief network. *Information Sciences, 178,* 2297–2314.
81. Hicklin, J., Moler, C., Webb, P., Boisvert, R. F., Miller, B., Pozo, R., et al. (2008). JAMA: A Java Matrix Package. Retrieved August 15, 2013, from http://math.nist.gov/javanumerics/jama/
82. AnyLogic Web site. (2013a). Downloads. Retrieved August 15, 2013, from http://www.anylogic.com/downloads.
83. Nagel, K., Beckman, R. L., & Barrett, C. L. (1999). *TRANSIMS for transportation planning.* Paper presented at the 6th International Conference on Computers in Urban Planning and Urban Management, Franco Angeli, Milano, Italy.
84. Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., & Nagel, K. (2009). MATSim-T: Architecture and simulation. In A. L. C. Bazzan & F. Klugl (Eds.), *Multi-agent systems for traffic and transportation engineering* (pp. 57–78). Hershey, PA: IGI Global.
85. Bradley, M., Bowman, J. L., & Griesenbeck, B. (2010). SACSIM: An applied activity-based model system with fine-level spatial and temporal resolution. *Journal of Choice Modelling, 3*(1), 5–31.

86. Goulias, K. G., Bhat, C. R., Pendyala, R. M., Chen, Y., Paleti, R., Londuri, K. C., et al. (2011). *Simulator of activities, greenhouse emissions, networks, and travel (SimAGENT) in southern California.* Paper presented at the Transportation Research Board 91st Annual Meeting, Washington, DC.
87. Pendyala, R. M., Kitamura, R., Kikuchi, A., Yomamoto, T., & Fujii, S. (2005). Florida activity mobility simulator. *Transportation Research Record, 1921*, 123–130.
88. Salvini, P. A., & Miller, E. (2005). ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems. *Networks and Spatial Economics, 5*(2), 217–234.
89. Jayakrishnan, R., Mahmassani, H. S., & Hu, T. Y. (1994). An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks. *Transportation research C*(2C(3)), 129–147.
90. Mahmassani, H. S. (2000). *DYNASMART-P programmer's and user's guides.* Final report. Austin, TX.: Center for Transportation Research, The University of Texas at Austin.
91. Chiu, Y.-C., Nava, E., Zheng, H., and Bustillos, B. (2011). DynusT user's manual. Retrieved August 15, 2013, from http://dynust.net/wikibin/doku.php
92. Noh, H., Khani, A., Hickman, M., Tian, Y., Chiu, Y.-C., & Villalobos, J. (2012). *Integrated vehicle and passenger simulation model considering a multimodal transportation network.* Paper presented at the 4th Transportation Researcb Board Conference on Innovations in Travel Modeling (ITM), Tampa, FL
93. Pendyala, R., Konduri, K., Chiu, Y.-C., Hickman, M., Noh, H., Waddell, P., et al. (2012). *Integrated land use–transport model system with dynamic time-dependent activity-travel microsimulation.* Paper presented at the Transportation Research Board 91st Annual Meeting, Washington, DC.
94. INRO Web site. (2010). Retrieved August 15, 2013, from http://www.inro.ca/en/products/dynameq/index.php.
95. Kettner, S. (2006, October 8–12). *Examining capacity increasing measures with the Dynameq model: equilibrium traffic assignment.* Paper presented at the 13th World Congress and Exhibition on Intelligent Transportation Systems, London, UK.
96. Boyce, D., Zhang, Y.-F., & Lupa, M. R. (1994). *Introducing feedback into the four-step travel forecasting procedure vs. the equilibrium solution of a combined model.* Paper presented at the Transportation Research Board 73rd Annual Meeting, Washington, DC.
97. Rickert, M., & Nagel, K. (2001). Traffic simulation: Dynamic traffic assignment on parallel computers in TRANSIMS. *Future Generation Computer Systems, 17*(5), 637–648.
98. Nagel, K., Stretz, P., Pieck, M., Donnelly, R., & Barrett, C. L. (1998). *TRANSIMS traffic flow characteristics*. Final report. Washington, DC.: U.S. Department of Transportation.
99. Balmer, M., Cetin, N., Nagel, K., & Raney, B. (2004). *Towards truly agent-based traffic and mobility simulations.* Paper presented at the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04), New York, NY.
100. Charypar, D., & Nagel, K. (2005). Generating complete all-day activity plans with genetic algorithms. *Transportation, 32*(4), 369–397.

101. Cetin, N., Burri, A., & Nagel, K. (2003). *A parallel queue model approach to traffic microsimulations.* Paper presented at the Transportation Research Board 82nd Annual Meeting, Washington, DC.
102. Meister, K., Balmer, M., Axhausen, K. W., & Nagel, K. (2006). *Planomat: A comprehensive scheduler for a large-scale multi-agent transportation simulation.* Paper presented at the 6th Swiss Transport Research Conference, Monte Verita, Ascona, Switzerland.
103. Kitamura, R., Chen, C., & Pendyala, R. M. (1997). Generation of synthetic daily activity-travel patterns. *Transportation Research Record, 1607*, 154–162.
104. Hagerstrand, T. (1970). What about people in regional science? *Papers of the Regional Science Association, 24,* 7–21.
105. Bradley, M. A., Bowman, J. L., & Griesenbeck, B. (2007). *Development and application of the SACSIM activity-based model system.* Paper presented at the 11th World Conference on Transportation Research, Berkeley, CA.
106. Miller, E. J. (2011). *Agent-Based Modeling for Integrated Transportation—Land Use Policy Analysis.* Paper presented at Northwestern University, Evanston, IL.
107. Mohammadian, A., & Miller, E. J. (2003a). *Dynamic Modeling of Household Automobile Transactions.* Paper presented at the Transportation Research Board 82th Annual Meeting, Washington, DC.
108. Mohammadian, A., & Miller, E. J. (2003b). *An empirical investigation of household vehicle type choice decision.* Paper presented at the Transportation Research Board 82th Annual Meeting, Washington, DC.
109. Salvini, P., & Miller, E. J. (2003). *ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems.* Paper presented at the 10th International Conference on Travel Behavior Research, Lucerne, Switzerland.
110. Miller, E. J., & Roorda, M. (2003). Prototype model of household activity-travel scheduling. *Transportation Research Record, 1831*, 114–121.
111. Miller, E. J., Farooq, B., Chingcuanco, F., & Wang, D. (2010). *Microsimulating urban spatial dynamics: historical validation tests using the ILUTE model system.* Paper presented at the Workshop on Urban Dynamics, Marbella, Chile.
112. Gao, W., Balmer, M., & Miller, E. J. (2010). Comparison between MATSim and EMME/2 on the grater Toronto and Hamilton area network. *Transportation Research Record: Journal of the Transportation Research Board, 2197,* 118–128.
113. Pendyala, R. M., Bhat, C. R., Goulias, K. G., Paleti, R., Konduri, K. C., Sidharthan, R., et al. (2012). *Application of a socio-economic model system for activity-based modeling: experience from southern California.* Paper presented at the Transportation Research Board 92nd Annual Meeting, Washington, DC.
114. Wang, F.-Y. (2005). Agent-based control for networked traffic management systems. *IEEE Intelligent Systems, 20*(5), 92–96.
115. NASA. (1985). *The Space Station Program.* NASA publication.
116. Dowell, M. L., & Bonnell, R. D. (1991). *Learning for distributed artificial intelligence systems.* Paper presented at the Twenty-Third Southeastern Symposium on System Theory, Columbia, SC.

117. Parunak, H. V. D. (1999). Industrial and practical applications of DAI. In G. Weiss (Ed.), *Multiagent Systems.* Cambridge, MA: MIT press.
118. Adler, J. L., & Blue, V. J. (2002). A cooperative multi-agent transportation management and route guidance system. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 433–454.
119. Chen, B., Cheng, H. H., & Palen, J. (2012). Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C: Emerging Technologies, 17*(1), 1–10.
120. Jennings, N. R., Sycara, K., & Wooldridge, M. J. (1998). A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems, 1*(1), 7–38.
121. Bernhardt, K. L. S. (2007). *Transportation Research Circular—Artificial intelligence in transportation: Information for application* (No. E-C113). Washington, DC: Transportation Research Board of the National Academies.
122. Kikuchi, S., Rhee, J., & Teodorovic, D. (2002). Applicability of an agent-based modeling concept to modeling of transportation phenomena. *Yugoslav Journal of Operations Research, 12*(2).
123. Chen, B., & Cheng, H. H. (2010). A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on Intelligent Transportation Systems, 11*(2), 485–497.
124. Parker, D. C., Manson, S. M., Janssen, M. A., Hoffimann, M. J., & Deadman, P. (2003). Multi-agent systems for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers, 93*(2), 314–337.
125. Burmeister, B., Doormann, J., & Matylis, G. (1997). Agent-oriented traffic simulation. *Transactions of the Society for Computer Simulation International—special issue: Multi-agent systems and simulation, 14*(2), 79–86.
126. Fischer, K., Chaib-draa, B., Muller, J. P., Pischel, M., & Gerber, C. (1999). A simulation approach based on negotiation and cooperation between agents: A case study. *IEEE Transactions on Systems, Man, and Cybernetics, 29*(4), 531–545.
127. Hidas, P. (2002). Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 351–371.
128. Hidas, P. (2005). Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research Part C: Emerging Technologies, 13*(1), 37–62.
129. Meignan, D., Simonin, O., & Koukam, A. (2007). Simulation and evaluation of urban bus-networks using a multiagent approach. *Simulation Modelling Practice and Theory, 15*(6), 659–671.
130. Bazzan, A. L. (2005). A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems, 10*(1), 131–164.
131. Botelho, L. M. (2000). *A control structure for agent interaction.* Paper presented at the Proceedings of the 2000 Intelligent Vehicles Conference, Dearborn, MI.
132. Choy, M. C., Cheu, R. L., Srinivasan, D., & Logi, F. (2003). *Real-time coordinated signal control using agents with online reinforcement learning.* Paper presented at the Transportation Research Board 82nd Annual Meeting, Washington, DC.

133. Choy, M. C., Srinivasan, D., & Cheu, R. L. (2003). Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 33*(5), 597–607.
134. Desjardins, C., Laumonier, J., & Chaib-draa, B. (2007). *Urban traffic control based on learning agents.* Paper presented at the Proceedings of IEEE Intelligent Transportation Systems Conference, Seattle, WA.
135. Katwijk, R. T. v., Koningsbruggen, P. v., Schutter, B. D., & Hellendoorn, J. (2005). Test bed for multiagent control systems in road traffic management. *Transportation Research Record, 1910*, 108–115.
136. Kosonen, L. (2003). Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C: Emerging Technologies, 11*(5), 389–403.
137. Roozemond, D. A. (1999). *Using intelligent agents for urban traffic control systems.* Paper presented at the Proceedings of the 6th World Congress on Intelligent Transport Systems, Toronto, Ontario, Canada.
138. Roozemond, D. A. (2001). Using intelligent agents for pro-active, real-time urban intersection control. *European Journal of Operational Research, 131*(2), 293–301.
139. Srinivasan, D., Choy, M. C., & Cheu, R. L. (2006). Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems, 7*(3), 261–272.
140. Adler, J. L., Satapathy, G., Manikonda, V., Bowles, B., & Blue, V. J. (2005). A multi-agent approach to cooperative traffic management and route guidance. *Transportation Research Part B: Methodological, 39*(4), 297–318.
141. Balbo, F., & Pinson, S. (2005). Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *Decision Support Systems, 41*(1), 131–146.
142. Hernandez, J. Z., Ossowski, S., & Garcia-Serrano, A. (2002). Multiagent architectures for intelligent traffic management systems. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 473–506.
143. Katwijk, R. v., & Koningsbruggen, P. v. (2002). Coordination of traffic management instruments using agent technology. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 455–471.
144. Ossowski, S., Hernandez, J. Z., Belmonte, M.-V., Fernandez, A., Garcia-Serrano, A., Perez-de-la-Cruz, J. L., et al. (2005). Decision support for traffic management based on organisational and communicative multiagent abstractions. *Transportation Research Part C: Emerging Technologies, 13*(4), 272–298.
145. Dia, H. (2002). An agent-based approach to modelling driver route choice behavior under the influence of real-time information. *Transportation Research Part C: Emerging Technologies, 10*(5), 331–349.
146. Wahle, J., Bazzan, A. L. C., Klugl, F., & Schreckenberg, M. (2002). The impact of real-time information in a two-route scenario using agent-based simulation. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 399–417.
147. Weyns, D., Holvoet, T., & Helleboogh, a. (2007). *Anticipatory vehicle routing using delegate multi-agent systems.* Paper presented at the Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Seattle, WA.

148. Logi, F., & Ritchie, S. G. (2002). A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transportation Research Part C: Emerging Technologies, 10*(5–6), 507–527.
149. Tomas, V. R., & Garcia, L. A. (2005). Agent-based management of non urban road meteorological incidents. *Multi-Agent Systems and Applications IV. Lecture Notes in Computer Science, 3690*, 213–222.
150. Belmonte, M.-V., Perez-de-la-Cruz, J. L., Triguero, F., & Fernandez, A. (2005). Agent coordination for bus fleet management. *2005 ACM Symposium on Applied Computing*, 462–466.
151. Belmonte, M. V., Perez-de-la-Cruz, J. L., & Triguero, F. (2008). Ontologies and agents for a bus fleet management system. *Expert Systems with Applications, 34*(2), 1351–1365.
152. Blum, J., & Eskandarian, A. (2002). Enhancing intelligent agent collaboration for flow optimization of railroad traffic. *Transportation Research Part A: Policy and Practice, 36*(10), 919–930.
153. Bocker, J., Lind, J., & Zirkler, B. (2001). Using a multi-agent approach to optimise the train coupling and sharing system. *European Journal of Operational Research, 131*(2), 242–252.
154. Lind, J., & Fischer, K. (1999). *Transportation scheduling and simulation in a railroad scenario: A multi-agent approach.* Paper presented at the Proceedings of the Fourth International Conference on the Practical Application of Intelligent Agents and Multi-Agents, London, UK.
155. Tsang, C. W., & Ho, T. K. (2008). Optimal track access rights allocation for agent negotiation in an open railway market. *IEEE Transactions on Intelligent Transportation Systems, 9*(1), 68–82.
156. Lordanova, B. N. (2003). Air traffic knowledge management policy. *European Journal of Operational Research, 146*(1), 83–100.
157. Nitschke, G. (2010). Cooperating air traffic control agents. *Applied Artificial Intelligence: An International Journal, 15*(2), 209–235.
158. Wangermann, J. P., & Stengel, R. F. (1998). Principled negotiation between intelligent agents:A model for air traffic management. *Artificial Intelligence in Engineering, 12*, 177–187.
159. Scemama, G. (1994). CLAIRE: A context-free AI-based supervisor for traffic control. In M. Bielli, G. Ambrosino, & M. Boero (Eds.), *Artificial intelligence applications to traffic engineering* (pp. 137–156). Utrecht, The Netherlands: VSP.
160. Garcia-Serrano, A. M., Vioque, D. T., Carbone, F., & Mendez, V. D. (2003). *FIPA-compliant MAS development for road traffic management with a knowledge-based approach: The TRACK-R agents.* Paper presented at the Challenges in Open Agent System Workshop, Melbourne, Australia.
161. Bazzan, A. L. C., Wahle, J., & Klugl, F. (1999). Agents in traffic modelling—from reactive to social behavior. In G. Weiss (Ed.), *Distributed artificial intelligence meets machine learning*: New York: Springer-Verlag.
162. Wang, F.-Y. (2008). Toward a revolution in transportations: AI for complex systems. *IEEE Intelligent Systems, 23(*6), 8–13.

163. Wardrop, J. G. (1952). *Some theoretic aspects of road traffic research*. Paper presented at the Conference of the Institute of Civil Engineering II.
164. Mahmassani, H. S. (1987). On boundedly rational user equilibrium in transportation systems. *Transportation Science, 21*(2), 89–99.
165. Daganzo, C. F. (1982). Unconstrained extremal formulation of some transportation equilibrium problems. *Transportation Science, 16*(3), 342–344.
166. Peeta, S., & Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Network and Spatial Economics, 1*, 233–265.
167. Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., et al. (2011). *Transportation Research Circular—Dynamic traffic assignment* (No. E-C153). Washington, DC: Transportation Research Board of the National Academies.
168. Yang, Q., & Koutsopoulos, H. N. (1996). A microsopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies, 4*(3), 113–129.
169. Mahmassani, H. S., & Hawas, Y. E. (1998). *A hierarchical distributed computational architecture for a centralized real-time dynamic traffic assignment.* Paper presented at the 3rd TRIannual Symposium on Transportation ANalysis (TRISTAN III), San Juan, Puerto Rico.
170. Nakayama, S. (2009). *Bayesian learning, day-to-day adjustment process, and stability of Wardrop equilibrium.* Paper presented at the 18th International Symposium on Transportation and Traffic Theory, Hong Kong, China.
171. Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and application to travel demand.* Cambridge, MA: MIT Press.
172. Sheffi, Y. (1984). *Urban transportation networks: Equilibrium analysis with mathematical programming methods.* Englewood Cliffs, NJ: Prentice-Hall, Inc.
173 Newell, G. F. (2002). A simplified car-following theory: A lower order model. *Transportation Research Part B: Methodological, 36*(3), 195–205.
174. Yeo, H., Skarbodonis, A, Halkias, J., Colyar, J., & Alexiadis, V. (2008). Oversaturated freeway flow algorithm for use in next generation simulation. *Transportation Research Record, 2088*, 68–79.
175. Laval, J. A., & Daganzo, C. F. (2006). Lane-changing in traffic streams. *Transportation Research Part B: Methodological, 40*(3), 251–264.
176. Tversky, A., & Kahneman, D. (1991). Loss aversion in riskless choice: A reference-dependent model. *The Quarterly Journal of Economics, 106*(4), 1039–1061.

Office of Safety Research and Development
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, VA 22101-2296

Publication No. FHWA-HRT-13-054
HRTM-30/11-13 (1M)E