

一种基于Repast Symphony平台的建模仿真方法

王纪才, 袁霄翔, 李建磊, 张梦瑶, 陈杨帅

周口师范学院人工智能学院, 河南 周口

收稿日期: 2026年3月11日; 录用日期: 2026年4月16日; 发布日期: 2026年4月28日

摘要

随着复杂系统研究的深入, 基于Agent建模与仿真技术得到了研究人员的广泛应用。Repast Symphony平台在基于Agent建模与仿真技术的催化之下应运而生, 其为研究者提供了极大的建模便利。为此, 本文对Repast Symphony平台上如何进行Agent建模和仿真的问题展开描述, 首先对Repast平台进行了具体的介绍、优势的分析以及类库的分析, 然后重点介绍了Repast Symphony平台建模仿真的一般流程, 最后通过一个具体的电子商务生态系统演化模型实例来对这一建模流程进行应用。本文的成果以期对研究者使用Repast Symphony进行Agent建模仿真研究提供一定的参考价值。

关键词

Repast Symphony, 复杂系统, 建模与仿真, Agent

A Modeling and Simulation Technology Based on Repast Symphony Platform

Jicai Wang, Xiaoxiang Yuan, Jianlei Li, Mengyao Zhang, Yangshuai Chen

School of Artificial Intelligence, Zhoukou Normal University, Zhoukou Henan

Received: March 11, 2026; accepted: April 16, 2026; published: April 28, 2026

Abstract

With the deepening of complex system research, agent-based modeling and simulation technology has been widely used by researchers. Repast Symphony platform came into being under the catalysis

文章引用: 王纪才, 袁霄翔, 李建磊, 张梦瑶, 陈杨帅. 一种基于 Repast Symphony 平台的建模仿真 方法[J]. 软件工程与应用, 2026, 15(2): 296-307. DOI: 10.12677/sea.2026.152028

of agent-based modeling and simulation technology, which provides great modeling convenience for researchers. Therefore, this paper describes how to carry out agent modeling and simulation on Repast Symphony platform. Firstly, it introduces the Repast platform, analyzes its advantages and class library, then focuses on the general process of modeling and simulation of Repast Symphony platform, and finally applies this modeling process through a specific example of e-commerce ecosystem evolution model. The results of this paper are expected to provide some reference value for researchers to use Repast Symphony for agent modeling and simulation research.

Keywords

Repast Symphony, Complex System, Modeling and Simulation, Agent

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来, 基于 Agent 建模与仿真技术成为了科研人员进行复杂系统建模研究的热点, 而技术的使用往往需要借助于 Agent 仿真程序, 这里的 Agent 指一个智能体, 具有自己的一些属性及行为[1]。由于复杂系统建模在编写程序时任务繁重且难度较大, 所以渐渐地出现了一些仿真平台来提供帮助[1]。目前使用的复杂系统建模仿真平台有 Repast、Repast Symphony、NetLogo 及 Swarm 等, 而这些平台中使用频率较多的为基于 Java 编程建模的 Repast Symphony 平台[2], 它是以 Repast 平台为基础发展而来的面向对象的多 Agent 建模与仿真平台, 并且可以实现仿真结果的可视化, 具有良好的扩展性。

当前, Repast Symphony 平台的发展速度很快, 应用领域较广(如, 经济学、社会学和军事等)。但通过查找相关文献发现, 目前关于 Repast Symphony 平台如何使用、如何建模的相关研究比较少。因此, 在 Repast Symphony 平台上对特定的模型进行研究, 建立一个具有普遍性的仿真模型是非常必要的。基于此, 本文以 Repast Symphony 平台为研究对象, 首先介绍了其技术特点及主要类库, 然后在相关文献[1]-[7]的基础上总结出 Repast Symphony 平台建模仿真的一般流程, 最后通过一个电子商务生态系统演化模型仿真实例进一步描述了利用 Repast Symphony 平台构建复杂系统的设计与实现方法, 同时对全文进行了总结。

2. Repast 平台分析

2.1. Repast 简介

Repast Symphony 是一款广泛应用于多 Agent 的建模仿真平台, 其最初由芝加哥大学社会科学计算研究中心开发, 旨在为社会仿真提供一个功能强大且易于扩展的工具环境。经过多年发展, 该平台已演变为支持多种编程语言的通用仿真框架, 适用于经济、生态、军事等多个研究领域。目前, 该平台拥有的类库具有生成、运行、显示和收集数据的功能, 并且能够对任意时刻的模型状态进行记录。Repast Symphony 仿真平台是在 Eclipse 开发平台上嵌入 Repast 仿真软件的集成开发环境[2]。该平台具有很强的交互能力, 易于学习和使用, 基于 Java 编程建模的软件系统, 尤其适合在 workstation 和小计算集群中使用[8]。Repast 平台最初仅支持 Java 语言, 自 3.0 版本起逐步扩展至 C# 与 Python, 进一步拓宽了用户群体。考虑到 Java 版本在功能与稳定性方面的优势, 本文后续讨论将围绕该版本展开。

2.2. Repast 优势

2.2.1. 跨平台

Repast Simphony 开发人员在设计之初便考虑到了跨平台的问题，保证了每个操作系统都有其相应的下载版本，现在可以适用于常见的 Linux、Windows、Mac 等操作系统[2]。它的跨平台性不仅表现在操作系统的版本下载上，更重要的是在仿真模型的使用上，在研究人员设计并发布一个新的模型之后，后来的研究者可以将其导入到和原来所使用的不同的平台上运行，这种优势让平台的安装和使用都变得很方便。

2.2.2. 建模类型丰富

开发人员考虑到 Repast Simphony 的使用人群编程能力可能参差不齐，因此，在设计建模方式时，不仅沿用了之前的通过编写具体代码来实现建模仿真的方法，而且还特地增加了用可视化的流程图完成仿真设计、用“point-and-click”完成建模的方法，这种方法适用于非计算机领域的研究者。

2.2.3. 强大的算法支持

鉴于人工智能技术所应用的领域越来越广泛，在算法的设计上，Repast Simphony 不仅支持以往其它平台所使用的传统的算法，还增加了对当前流行的人工智能算法的支持，这一算法的实现是通过引用内置的开源算法类库包，在构建具体模型时通过对所需要的算法在类库包里进行调用即可。

2.2.4. 数据的批处理和收集

开发人员在设计平台之初便考虑到了后续的研究者在实验时要通过手动输入大量的参数来获取不同的实验数据，收集数据时也要手动完成。这无疑是效率低下且任务艰巨的工作。为此，在平台上增设了批处理的功能和数据收集的功能，研究者可以把所有想要实现的参数一次性输入并配置好，然后让平台进行仿真实验。在收集数据时，通过平台上的数据收集功能把仿真结果进行记录并保存。

2.2.5. 支持构建 3D 的表现形式

通常来说，建模仿真平台提供的都是与 2D 相关的工具包，而 Repast Simphony 在设计时考虑到了一些复杂系统不能通过 2D 工具包得出直观的仿真结果，因此增加了 3D 表现相关的工具包，在研究复杂系统时通过构建 3D 模型，进一步对 Agent 的活动进行精确的研究[2]。

2.2.6. 建模元素的抽象性

由于 Repast Simphony 为基于 Java 的编程建模仿真软件，因此，Repast Simphony 具有一定的抽象性。在进行建模与仿真流程设计时，把仿真所要涉及的一些重要的元素抽象出来，将它们描述为具体的 Java 类[9]，这有利于描述元素之间的关系，以此形成仿真时所需要的流程图，让仿真看起来比较简洁和直观。

2.2.7. 高性能仿真

仿真性能优化问题只有经过平台得到一定规模的具体应用之后才能够很好地实现，在开发人员设计之初虽然没有着重关注性能优化问题，而是重点关注平台的运行速度问题，但是随着平台被大规模使用以及平台版本的更新，性能优化问题已经得到了很大的改进和重视[9]，随着研究的深入和广泛，性能优化将会得到进一步的提升。

2.3. 类库分析

开发人员在开发 Repast Simphony 平台时沿用了 Repast 的许多内容[6]，类库便是其中之一，分析类库对理解整个仿真过程有着至关重要的作用，因为仿真平台所提供功能是通过调用类库实现的。以

下介绍在一般的建模仿真过程中经常用到的类。

2.3.1. 引擎类(Engine)

在 Repast Symphony 中, 引擎类至关重要, 负责了整个仿真过程的调度与控制, 而且其调度机制融合了离散时间推进与事件触发两种类型。在仿真的过程中, 系统运行的每个时间周期称为“tick”, 推动仿真周期前进的事件则由 Agent 行为触发, 并通过 Schedule (时间表)对象设定执行时间与优先级。这种设计使得仿真过程具备良好的灵活性与可扩展性。

2.3.2. 空间类(Space)

Agent 的行为与交互通常依赖于其所处的环境, Repast Symphony 通过空间类(Space)来定义这一环境 [7]。为了满足不同用户的建模需求, 该平台提供了多种空间类型, 例如网格(Grid)用于离散空间建模、连续空间(ContinuousSpace)适用于连续坐标系统、网络(Network)与 GIS 模块则分别用于关系建模与地理信息模拟, 这几种空间类型可以被单独或者组合使用。

2.3.3. Agent 类

Agent 类属于为数不多的需要被研究者自己定义的类, 因为在 Repast 类库中没有提供具体的 Agent 类, 但这并不意味着 Agent 类不重要, 而是因为 Agent 类与 Java 这种面向对象的编程语言中的 Object 相类似, 在进行建模仿真的过程之中, 研究者可以根据自己所研究的具体模型来定义所需要的 Agent 类以及属性和行为。

2.3.4. 随机数生成类(Random)

随机数生成类在仿真的过程中发挥着重要的作用, 仿真时所产生的随机数会直接影响到仿真的结果 [2]。随机数的生成必须借助于类 RandomHelper, 借助 RandomHelper 可以迅速地产生随机数, 以便供研究者使用。

2.3.5. 输入输出类

Repast Symphony 支持两种仿真执行模式: 批处理与非批处理。前者通过读取预设参数文件, 实现多组实验的自动批量运行及结果输出, 主要适用于参数敏感性分析与大规模仿真实验; 后者则借助图形用户界面(GUI)允许用户在仿真过程中动态调整模型参数以产生实验数据, 适合模型调试与实时观察。

2.4. 建模与仿真流程

在 Repast Symphony 平台中, 模型构建与仿真执行分别对应“设计态”与“运行态”两个阶段, 并且相应阶段所展现的界面各异。在设计态下, 用户可通过编写代码或绘制流程图的方式构建 Agent 模型及其行为逻辑; 运行态则由平台自动生成可执行框架[10][11], 用户无需手动干预, 专注于参数设置与结果分析。通过研究 Repast Symphony 平台的特点, 总结出了以下建模步骤及其仿真运行流程图(如图 1 所示), 同时在第 2 节以一个具体的电子商务生态系统演化模型实例来对这一建模流程进行应用。具体内容如下:

(1) Repast Symphony 平台建模步骤

- ① 首先, 根据用户所研究的对象确定相应的模型名称;
- ② 然后, 将模型中涉及的交互对象抽象成相应的 Agent;
- ③ 接着, 定义 Agent 的环境、属性和行为;
- ④ 最后, 根据用户所研究对象在现实中的运行环境设置仿真参数并收集相应的实验数据。

(2) Repast Symphony 平台仿真运行流程图

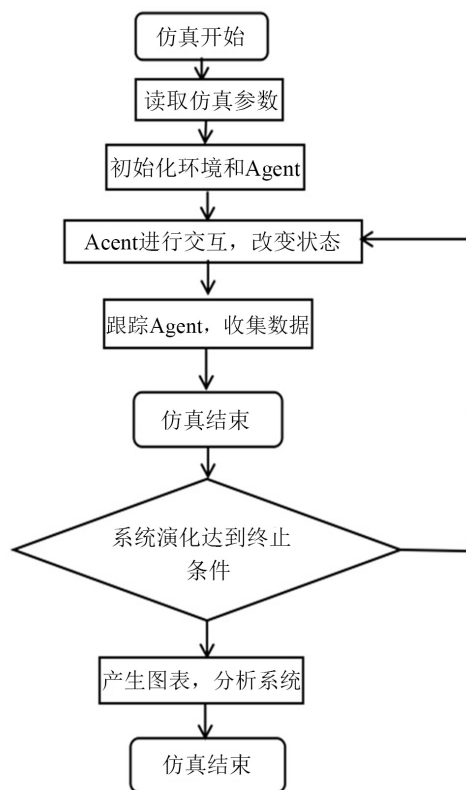


Figure 1. The flowchart of simulation operation
图 1. 仿真运行流程图

3. 基于 Repast Simphony 的电子商务系统演化模型构建

3.1. 问题背景

通过对国内外相关文献的搜索, 研究人员发现: 国外的文献对这方面的研究主要针对的是“商业生态系统”这一更加广泛的概念, 几乎很难看到有学者对“电子商务生态系统”这一名词进行研究。“电子商务生态系统”这一概念源于国内学者[12]对商业生态系统理论的延伸与应用。胡岗岚[13]等人以此为理论基础, 分析了中国电子商务发展中的集群化现象, 并提出了相应的系统演化路径, 为后续基于 Agent 的建模仿真研究提供了理论支持。在电子商务生态系统中, 电子商务的直接参与者分为了三种: 服务需求方、电子商务平台、服务提供商。利用多 Agent 建模技术在 Repast Simphony 平台上构建电子商务生态系统模型, 以便直接地观察电子商务系统的演变状况, 并且根据设置的参数研究系统内部各个主体之间的关系。下面以电子商务平台 Agent 为例来介绍系统模型中 Agent 属性与行为。

3.2. Agent 的环境和构建

3.2.1. Agent 的环境

在 Repast Simphony 建模中, 系统环境(Context)主要用于规定主体(Agent)的活动区域及其相互间存在的关系。该平台提供了多种空间模型, 包括网格(Grid)、连续空间(Continuous Space)和网络(Network)。以电子商务生态系统模型为例, 网格被选为实现 Agent 环境空间的核心类, 通常用于描述多个 Agent 在限定区域内的离散空间分布及其活动范围。另外, 该电子商务生态系统模型中抽象出了三类 Agent, 分别为服务需求方 Agent、电子商务平台 Agent 及服务提供商 Agent, 它们被布置于网格之上, 每个 Agent 均具

有唯一的二维坐标，其感知范围则由视野参数决定[12]。

3.2.2. Agent 的构建

模拟仿真系统中 Agent 的构建需要遵循以下规则：

(1) Agent 是有限理性的，角色和类型各异，对不同的行为的选择也不尽相同。获取的信息、处理信息的能力也是有限的，它的有限还表现在决策是基于有限的信息基础上，往往会根据所拥有的信息、认知和行为偏好而做出不同的反应。

(2) Agent 会不断受到周围其他的 Agent 的影响而产生一定的动作，这些 Agent 的行为会产生有限的和局部性的效果，但它们可以通过邻近 Agent 的相互传播来产生对较远的 Agent 的影响和作用。

(3) 外界因素在任何时刻都会对系统和 Agent 的行为产生大大小小的影响，虽然 Agent 自身无法对外部因素做出相应的改变，但是可以通过它的适应性特点来适应系统的环境。

3.3. Agent 的属性和行为

3.3.1. Agent 的属性

以电子商务平台 Agent 为例，其属性可划分为特征属性与状态属性两类。特征属性用于表述 Agent 的内在特征，直接影响其行为模式与决策机制，主要包括运营成本、风险偏好及 Agent 类型。状态属性则用于表述 Agent 在当前仿真周期中的实时状态，并随系统运行而动态更新，决定其在环境中的行为能力。本文所抽象的电子商务平台 Agent 的状态属性具体包括：能量值(对应现实场景中的企业规模)、感知与获取资源的范围(如寻找订单的最大距离)、外包发布的范围(即向外发布订单外包的最大范围)，以及每个周期内可处理的订单数量。

3.3.2. Agent 的行为

以电子商务平台 Agent 为例，其行为主要包括移动(用于寻找订单)、捕食(即完成订单)、繁殖(对应现实中的企业优胜劣汰)以及外包需求。每当 Agent 完成某一行为后，其状态属性将随之更新。以繁殖行为为例：在每个运行周期内，Agent 会根据更新后的状态属性判断是否触发繁殖机制。系统模型借鉴遗传算法中的交叉变异思想，通过平台类型的交叉重组生成新的变体(如 O2O 平台)，进而实现 Agent 类型的演化。在此过程中，Agent 的特征属性与状态属性也将相应调整，以维持种群多样性并模拟自然选择过程。

4. 基于 Repast Symphony 的电子商务系统演化模型实验

4.1. 设置仿真参数

为了真实地模拟现实社会中的电子商务系统，使实验结果更具有参考性和科学性，本文在设置仿真参数的时候参考了我国 2025 年国家工业生产者中制造业的相关数据[14]。基于这些相关的数据，对本文的电子商务生态系统模型所涉及的主要参数设置如表 1 所示。

Table 1. E-commerce service ecosystem model

表 1. 电子商务服务生态系统模型表

变量	值
系统环境大小	200 × 90
环境结构	网格 Grid 结构，循环边界
初始电子商务平台数量	30
初始服务提供商数量	100

续表

服务平台与服务提供商分布	随机
服务平台初始能力	各种属性、能力在规定范围内随机获取
服务提供商初始能力	各种属性、能力在规定范围内随机获取
需求分布区域个数	3
单个需求初始需求量	15~80 之间的随机数，越接近区域中心，需求量越大
单个需求的初始增量	20~45 之间的随机数
传统电子商务需求分布区域 1	位置(45, 35)，半径 20~25
传统电子商务需求分布区域 2	位置(150, 40)，半径 30~40
O2O 需求分布区域 1	位置(95, 55)，半径 30~35
需求量变化类型	稳步增长，波动变化
变异产生 O2O 平台的时间节点	50
初始电子商务平台类型与类型编码	EB_Platform, 000
初始服务提供商类型与类型编码	EB_ServiceProvider, 000
类型编码变异概率	0.2
类型编码交叉概率	0.2
平台运营成本	固定值 6

4.2. 实验结果与分析

图 2 展示了系统模型初始化后的空间布局。图中颜色深浅代表需求量大小，区域 1 和 3 代表传统的电子商务需求，区域 2 代表 O2O 类型的需求。服务平台与服务提供商分别以电脑图标与卡车图标标识，便于区分主体类型与功能角色。

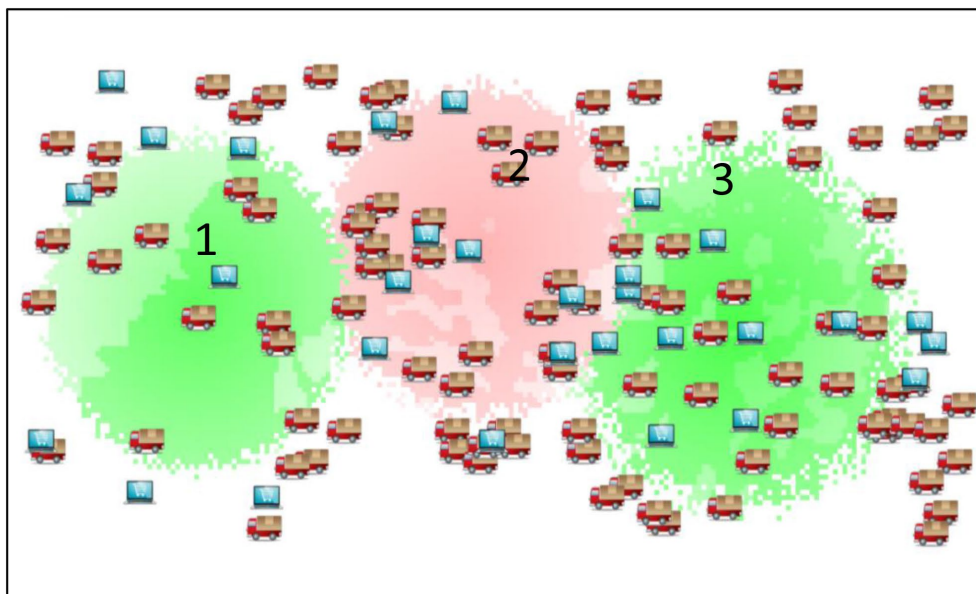


Figure 2. The overall situation after system model initialization
图 2. 系统模型初始化后的整体情况

在当前参数设定下，系统模型演化过程中服务平台数量和服务提供商数量出现的巨大的波动，最终在第 120 个周期时，服务平台个数趋近于 0，模型停止运行。图 3 和图 4 分别表现了服务平台和服务提供商在整个运行过程中的数量变化。可以看到，平台数量的波动和服务提供商数量的波动有相似的趋势，这是因为平台和服务提供商存在互利共生的关系，同处一个价值链。

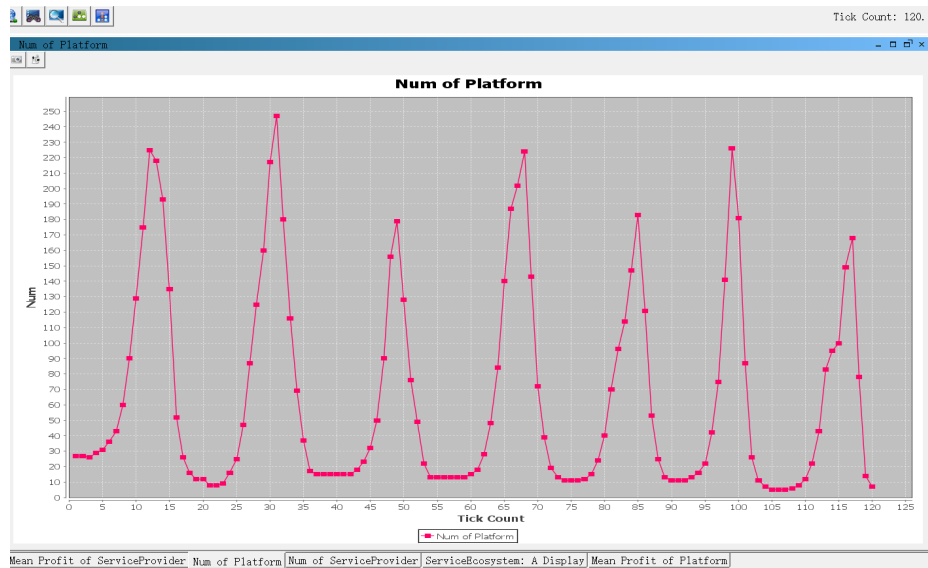


Figure 3. The changes in the number of service platforms

图 3. 服务平台数量变化

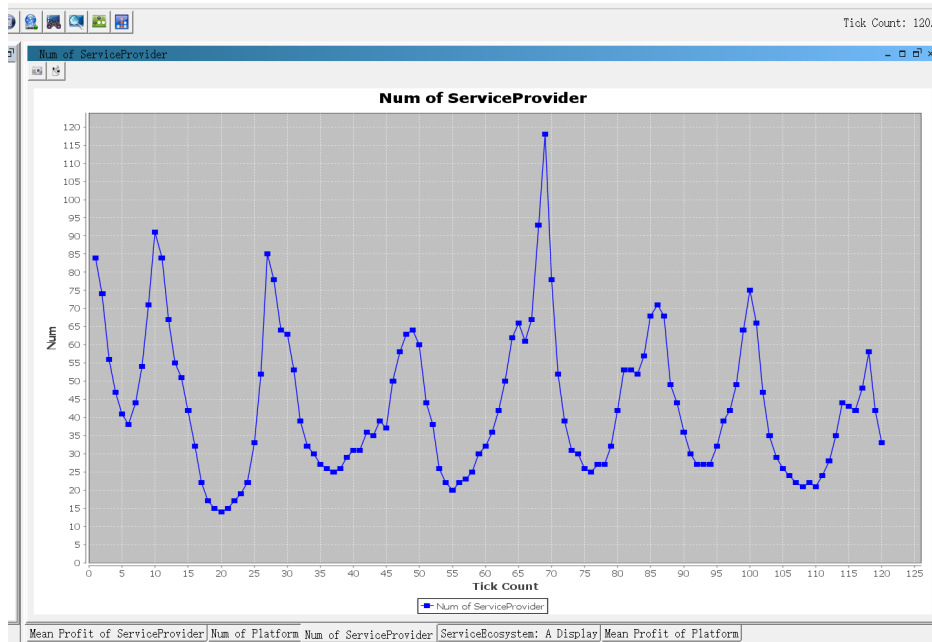


Figure 4. The changes in the number of service providers

图 4. 服务提供商数量变化

系统模型运行过程中，在第 8 个运行周期时，电子商务平台和服务提供商出现了聚集，如图 5 所示，其中深红色方块代表 Platform Agent 产生的需求，颜色越深需求量越大。

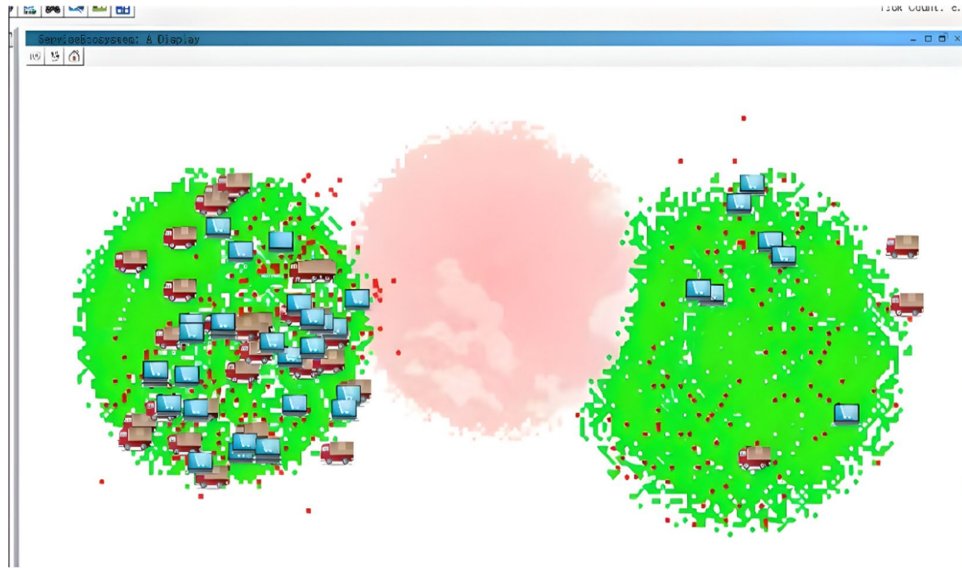


Figure 5. The system model at the 8th operational cycle
图 5. 系统模型在第 8 个运行周期时的情况

图 6 显示系统在第 14 个运行周期时电子商务服务生态系统出现爆发式发展，这也产生了激烈的竞争，平台和服务提供商利润快速下降。从图中可以明显地看到左侧区域 Platform Agent 产生的需求颜色相比之前变浅，因为在激烈的竞争环境下，每个平台能够获得的需求量减少，随之产生的下阶段需求在减少。

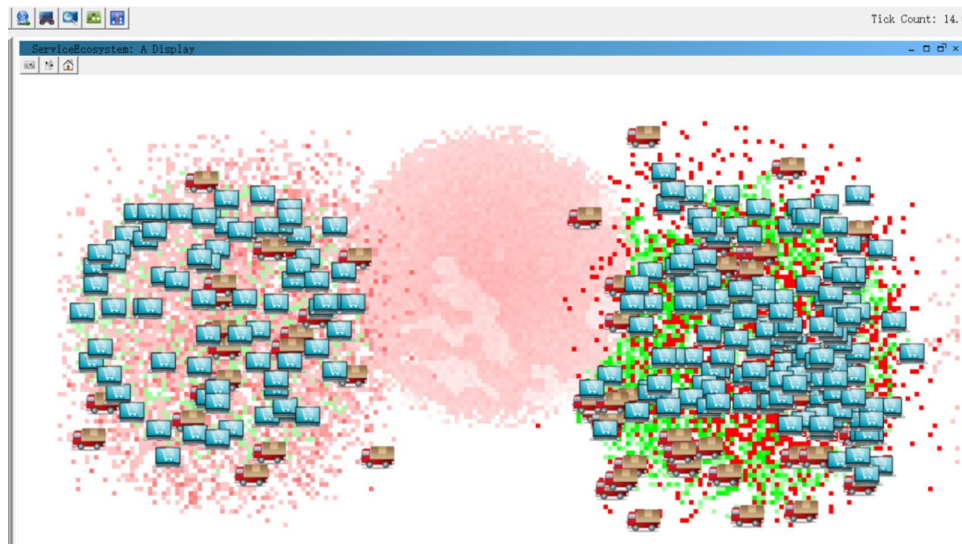


Figure 6. The system model at the 14th operational cycle
图 6. 系统模型在第 14 个运行周期时的情况

由于每个 Platform Agent 能够获得的需求量在不断减少，获得的利润不能支持它们的运行，它们的数量在此后也不断下降，同时 Service Provider Agent 能够获得的需求也随之减少，它们的数量也出现急剧下降。系统模型在第 21 个运行周期时达到低谷，如图 7 所示。此后系统一直处于剧烈变化状态。

实验参数设定在第 50 个周期时类型编码可以通过交叉变异产生 O2O 类型的电子商务平台, 如图 8 所示。当系统模型运行到第 60 个周期时, O2O 平台出现, 并在 O2O 需求区域出现聚集, 如图 9 所示。之后, 系统也同样出现剧烈变化, 最终在第 120 个周期时所有 Platform Agent 消亡且系统运行结束, 如图 10 所示。

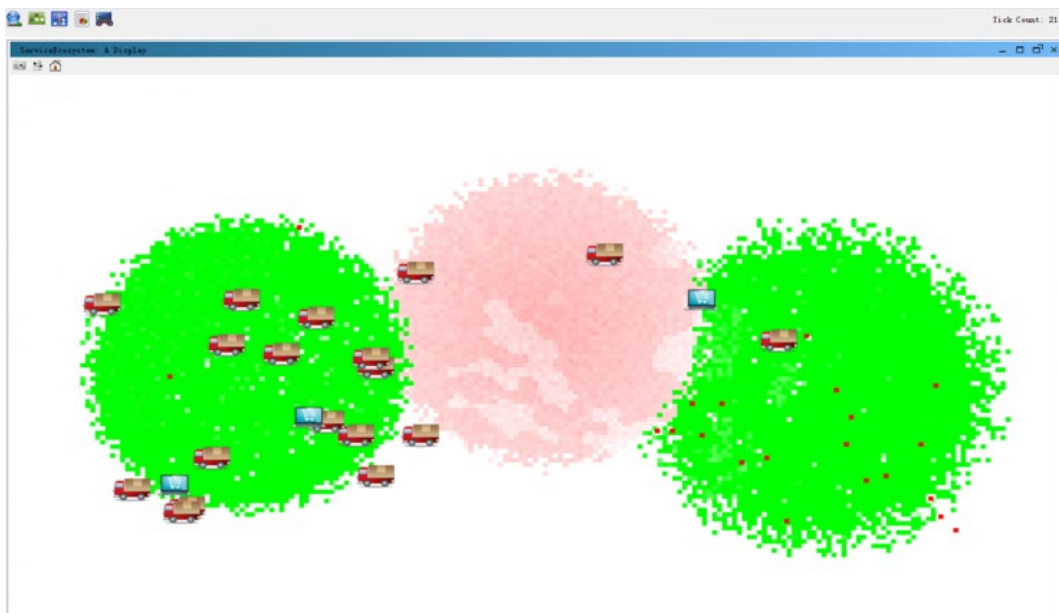


Figure 7. The system model in the 21st operation cycle
图 7. 系统模型在第 21 个运行周期时的情况

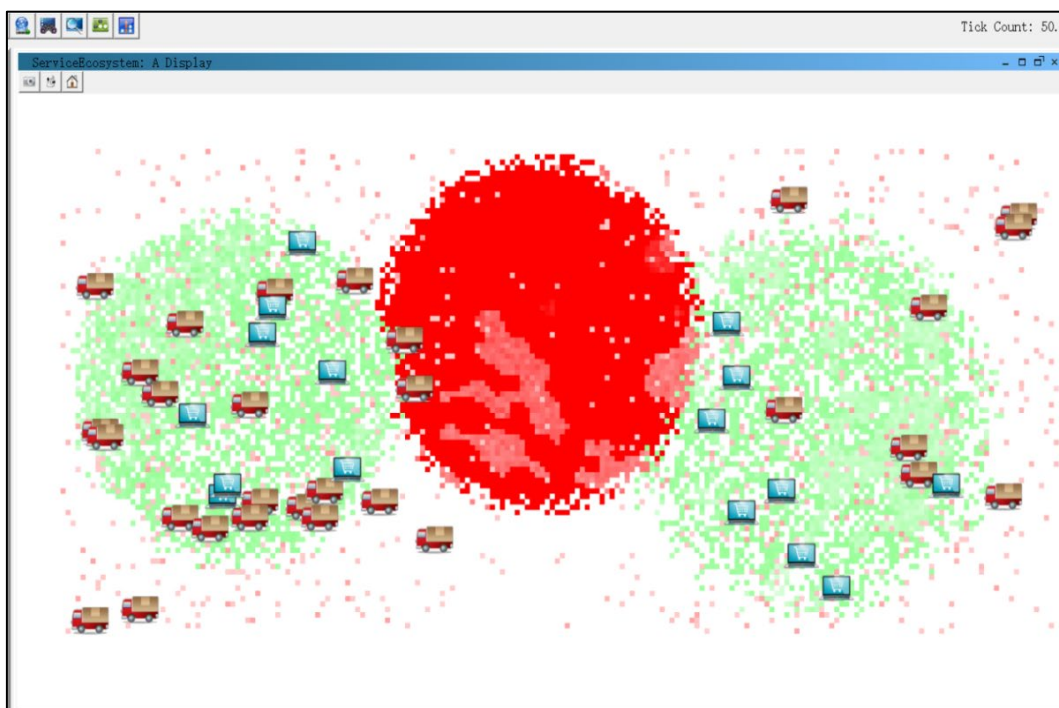


Figure 8. The system model at the 50th operation cycle
图 8. 系统模型在第 50 个运行周期时的情况

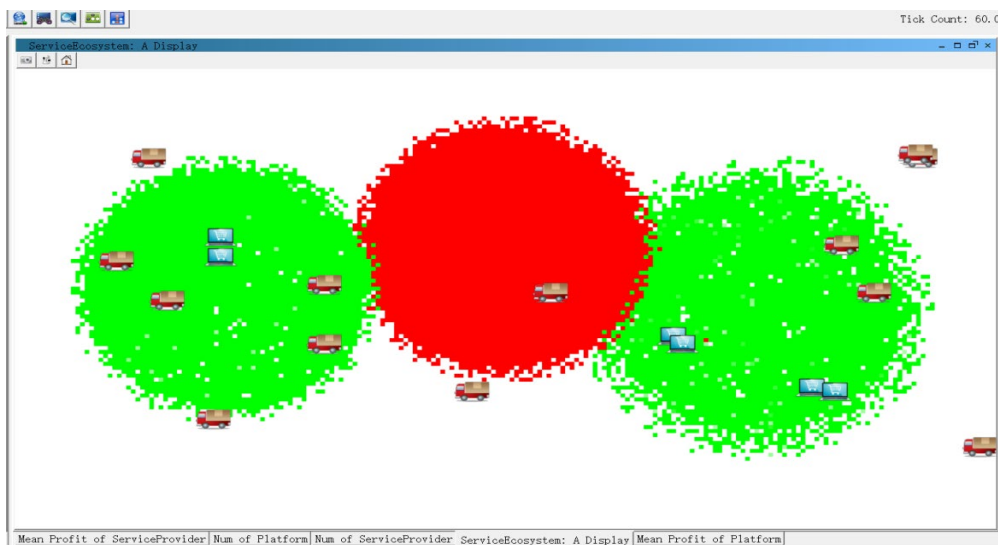


Figure 9. The system model in the 60th operation cycle
图 9. 系统模型在第 60 个运行周期时的情况

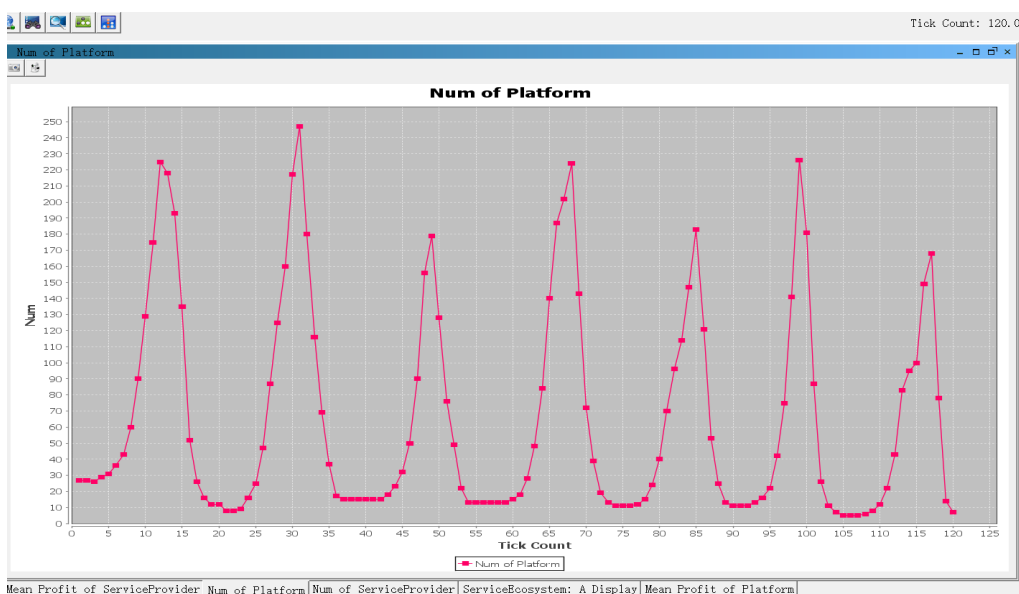


Figure 10. The system model in the 120th operation cycle
图 10. 系统模型在第 120 个运行周期时的情况

5. 总结

Repast Symphony 平台为复杂系统的建模与仿真分析提供了一种高效的计算实现手段。借助平台所提供的图表与可视化输出，研究人员能够直观地评估不同模型、策略及方法的实际表现。本文围绕 Repast Symphony 平台展开研究，首先对其技术架构与核心类库进行了梳理；随后，在归纳已有文献的基础上，总结了基于该平台的建模仿真流程，主要涵盖模型构建与仿真执行两个主要阶段；最后，以电子商务生态系统演化模型为例，展示了利用 Repast Symphony 构建复杂系统模型的具体实现过程，并对实验结果进行了分析。本文的研究可为基于 Repast Symphony 的 Agent 建模与仿真提供一定参考，后续研究可依据具体问题，参照本文提出的建模流程，分析 Agent 的环境、属性及行为特征，并借助平台内置的基础模型

进行扩展,从而实现复杂系统的高效建模。未来,本文将在更多领域开展基于 Repast Symphony 的建模仿真研究,以进一步丰富相关应用案例。

基金项目

2025 年度河南省高等学校重点科研项目“基于计算实验评估的社群化制造服务匹配策略研究”(25A520052)。

参考文献

- [1] 张砚劼, 丁晨. 基于 Repast Symphony 平台的经济系统仿真初探[J]. 中国商界(下半月), 2009(3): 161-162.
- [2] 谢欧. 基于多智能体的急性呼吸道病毒传播建模及仿真研究[D]: [硕士学位论文]. 岳阳: 湖南理工学院, 2023.
- [3] 刘卉. 基于强化学习的草地可持续利用多情景仿真模型研究[D]: [硕士学位论文]. 太原: 太原科技大学, 2024.
- [4] 刘文龙. 基于 Repastsymphony 平台人群运动行为仿真及优化[D]: [硕士学位论文]. 天津: 天津工业大学, 2018.
- [5] 刘文龙, 张晶, 周绥平, 等. 基于 Repast Symphony 平台的人群运动行为仿真及优化[J]. 计算机科学, 2018, 45(12): 187-191.
- [6] Xue, X., Chang, J. and Liu, Z. (2014) Context-Aware Intelligent Service System for Coal Mine Industry. *Computers in Industry*, **65**, 291-305. <https://doi.org/10.1016/j.compind.2013.11.010>
- [7] 王宇宾. 基于 Repast Symphony 平台的建模与仿真技术[J]. 计算机系统应用, 2015, 24(10): 17-22.
- [8] 靳晶. 基于 Agent 电子商务系统的建模与实现[D]: [硕士学位论文]. 石家庄: 河北工程大学, 2011.
- [9] 傅游, 杜宇. 基于 Agent 的 Repast 建模仿真平台[J]. 信息技术与信息化, 2009(2): 53-55.
- [10] 张俊瑞, 王秀华. 基于 Agent 的建模与仿真方法及仿真平台浅析[J]. 网络安全技术与应用, 2016(12): 53-54.
- [11] 宾厚, 徐子澄. 虚拟农业供应链模式的 Repast 仿真分析[J]. 洛阳师范学院学报, 2024, 43(8): 18-21.
- [12] 胡岗岚, 卢向华, 黄丽华. 电子商务生态系统及其演化路径[J]. 经济管理, 2009, 31(6): 110-116.
- [13] 胡岗岚, 卢向华, 黄丽华. 电子商务生态系统及其协调机制研究——以阿里巴巴集团为例[J]. 软科学, 2009, 23(9): 5-10.
- [14] 国家统计局. 2025 年月度工业生产者出厂价格分类指[EB/OL]. <https://data.stats.gov.cn/dg/website/page.html>, 2026-03-11.