

УДК 303.09

Москотіна, Р. (2021). Формалізація соціологічної теорії за допомогою агентного моделювання. *Соціологічні студії*, 1 (18), 41–48. <https://doi.org/10.29038/2306-3971-2021-01-41-48>

## Формалізація соціологічної теорії за допомогою агентного моделювання

**Руслана Москотіна** –  
*аспірантка спеціальності  
«Соціологія», факультет  
соціології, Київський  
національний університет  
ім. Тараса Шевченка,  
Україна*

**Ruslana Moskotina** –  
*PhD student in Sociology,  
Faculty of Sociology, Taras  
Shevchenko National University  
of Kyiv, Ukraine*  
Email: rmoskotina@ukr.net  
ORCID:  
<https://orcid.org/0000-0002-2195-3121>

У статті схарактеризовано процес розробки агентної моделі. Її розглянуто як формальну теорію, представлену у вигляді програмного коду, для створення якої потрібна вербальна теорія. Визначено, що теорія середнього рівня є такою соціологічною теорією, яку можна представити у вигляді агентної моделі. Вона дає змогу пояснити виникнення соціальних явищ або процесів макrorівня з дій і взаємодій агентів на мікрорівні. Теорія середнього рівня, не спрямована на пояснення всіх можливих соціальних явищ чи процесів, стосується лише обмеженого їх кола та уможливорює надавання часткових пояснень. Вона має таку структуру: агенти, середовище, правила. Також виокремлено етапи формалізації теорії середнього рівня. Вони стосуються випадків, коли ми реалізуємо агентну модель у середовищі R і розглядаємо її як функцію в R. Ці етапи такі: 1) специфікація моделі; 2) представлення характеристик агентів у вигляді програмного коду в R; 3) представлення середовища у вигляді програмного коду в R; 4) представлення правил у вигляді програмного коду в R; 5) створення візуалізації; 6) обчислення числових показників; 7) виявлення та виправлення помилок у програмному коді в R. У результаті буде отримано агентну модель, із якою можна проводити комп'ютерні експерименти.

DOI: 10.29038/2306-3971-2021-01-41-48

Received: April, 2021  
1<sup>st</sup> Revision: May, 2021  
Accepted: May, 2021

**Ключові слова:** агентна модель, вербальна теорія, формальна теорія, програмний код.

**Moskotina Ruslana. Sociological Theory Formalization with Agent-Based Modeling.** Sociologists usually use verbal theories to describe and explain the social phenomena. But formal theories also can be applied for this purpose. There are verbal theories written in formal language. In particular formal theories are important for studying complex systems. Complex systems are dynamic and their behavior changes over time. Verbal theories help us to describe the components of a complex system, their characteristics, actions and interactions but cannot catch changes in the behavior of complex system over time. Agent-based modeling is a method of theoretical analysis of complex systems. Agent-based model is a formal theory presented as a program code. It is applied to explain the behavior of complex system and require a verbal theory. We find out that the middle-range theory is a sociological theory can be presented as an agent-based model. It explains the emergence of macro-level phenomena from the actions and interactions of agents at the micro-level. The middle-range theory explains a limited range of macro-level phenomena and provides partial explanations. It consists of agents, environment and rules. We also propose the stages of middle-range theory formalization. They are relevant for cases when we implement an agent-based model in R and consider it as a function in R. The stages of middle-range theory formalization are: 1) model specification; 2) representation agents characteristics as a program code in R; 3) representation an environment as a program code in R; 4) representation rules as a program code in R; 5) creating a visualization; 6) calculation of numerical indicators; 7) detection and correction of errors in the R code. As a result we get an agent-based model and can conduct computer experiments with it.

**Key words:** agent-based model, verbal theory, formal theory, program code.

## ПОСТАНОВКА ПРОБЛЕМИ

Соціологи так чи інакше мають справу з теоріями, як вербальними (сформульованими природною мовою), так і формальними. Останні передбачають запис вербальних теорій за допомогою формальних мов, наприклад, у вигляді системи рівнянь, програмного коду. Потреба в представленні вербальної теорії у формалізованому вигляді може з'явитися тоді, коли її потрібно перевірити на емпіричних даних. Ще одна причина використання формальних теорій – пояснення поведінки складних систем. Вербальні теорії дають змогу описати компоненти складної системи: їхні характеристики, взаємодії між ними, правила дій. Однак поведінка складної системи емерджентна, її не можна звести до суми характеристик або дій окремих компонентів системи. До того ж, складні системи динамічні; це означає, що їхня поведінка з часом змінюється. Вербальні теорії не дають змоги спостерігати ці зміни на кожному часовому проміжку. Також П. Смальдіно (Smaldino, 2017) вважає, що вербальні теорії розпливчасті й допускають кілька інтерпретацій. Формальні теорії позбавлені цих недоліків. Методом теоретичного аналізу складних соціальних систем виступає агентне моделювання (Hedström, Bearman, 2017, с. 12). Воно призначене для розробки агентних моделей – формальних теорій, представлених у вигляді програмного коду. Відповідно, процес створення агентної моделі потребує процедури формалізації: спершу потрібно запропонувати вербальну теорію, а потім представити її у вигляді програмного коду. Нас цікавить формалізація соціологічних теорій, які розглядатимемо як вербальні теорії, спрямовані на опис і пояснення соціальних явищ та процесів. Однак які соціологічні теорії можуть бути представлені у вигляді агентних моделей і як має виглядати процедура їх «перекладу» тією чи іншою мовою програмування? Спробуємо дати відповіді на ці запитання.

### 1. АНАЛІЗ НАУКОВИХ ПУБЛІКАЦІЙ

Про формалізацію вербальної теорії говорив П. Смальдіно (Smaldino, 2017, 2020), надавши рекомендації з «перекладу» вербальної теорії формальною мовою. Проте він не приділяв уваги конкретно створенню агентної моделі й не торкався питання її реалізації в тому чи іншому програмному середовищі. Про розробку агентних моделей ідеться в працях Н. Гілберта у співавторстві з К. Троїтчем (Gilbert, Troitzsch, 2005) та П. Терна (Gilbert, Terna, 2000), а також М. Сальгадо (Salgado, Gilbert, 2013), Ч. Макала й М. Норта (Macal, North, 2014). Зокрема, Н. Гілберт і К. Троїтч у праці «Simulation for the social scientist» описували їх реалізацію в середовищі NetLogo. Однак дослідження цих науковців приділяють досить мало уваги ролі вербальної теорії у створенні агентної моделі. П. Хедстрем з П. Ілікоскі (Hedström, Ylikoski, 2010) та Л. Уден (Hedström, Udehn, 2017) указують на значення агентного моделювання в розвитку механістичних пояснень і теорій середнього рівня як приклади соціальних механізмів. Проте вони не приділяють уваги зв'язку між агентними моделями й теоріями середнього рівня.

### 2. МЕТА ТА ЗАВДАННЯ СТАТТІ

**Мета статті** – виокремити етапи формалізації соціологічної теорії за допомогою агентного моделювання. Відповідно до мети сформульовано такі **завдання**:

- 1) розглянути теорію середнього рівня як соціологічну теорію, що може бути представленою у вигляді агентної моделі, та окреслити її структуру;
- 2) виокремити етапи формалізації теорії середнього рівня.

### 3. ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ, ОБГРУНТУВАННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Агентне моделювання дає змогу створювати моделі, які є своєрідними «копіями» складних систем та поєднують мікро- й макрорівні аналізу. Е. Брух і Дж. Атвелл (Bruch, Atwell, 2015) зазначають, що ці моделі включають агентів, які діють та взаємодіють у середовищі відповідно до певних правил. На мікрорівні агентне моделювання розглядає окремі компоненти складної системи. Це агенти, їхні характеристики, дії й взаємодії. На макрорівні спостерігаємо поведінку складної системи – соціальні явища або процеси, які потрібно пояснити. Агентне моделювання не просто констатує факт наявності зв'язку між мікро- та макрорівнями аналізу, а й надає так звані

механістичні пояснення. Ф. Барбера (Barbera, 2006) вважає механізм мікропроцесом, що пояснює, як у тих або інших умовах може бути згенеровано явище чи процес макрорівня. Тобто, по суті, механізм дає змогу розповісти «історію» виникнення соціального явища або процесу. Як зазначають П. Хедстрем і П. Ілікоскі (Hedström, Ylikoski, 2010), механістичний підхід має багато спільного з ідеєю Р. Мертона щодо теорій середнього рівня. А П. Хедстрем і Л. Уден (Hedström, Udehn, 2017) вважають теорії середнього рівня пояснювальними. Вони складаються з експланандуму (явища чи процеси, які потрібно пояснити) та експланансу (сукупність факторів, що пояснюють соціальні явища або процеси). На думку П. Хедстрема та П. Ілікоскі (Hedström, Ylikoski, 2010), теорія середнього рівня дає змогу надавати так звані часткові пояснення. Це означає, що вона пояснює соціальні явища й процеси за допомогою найбільш суттєвих чинників. Також, як зазначають П. Хедстрем і Л. Уден (Hedström, Udehn, 2017), теорії середнього рівня стосуються не всіх можливих соціальних явищ і процесів, а лише обмеженого їх кола. Водночас вони є досить загальними, оскільки не «прив'язані» до окремих подій, що відбуваються в певному місці в певний час. Ф. Скавццоні (Squazzoni, 2012) виокремлює так звані моделі середнього рівня. Це агентні моделі, що стосуються певного класу соціальних явищ та процесів й уможливають надання часткових пояснень. Вони не мають стосунку до окремих подій, які відбуваються в певному місці в певний час. Отже, теорію середнього рівня може бути представлено у вигляді агентної моделі. Вона виступає соціологічною теорією, що дає змогу пояснити виникнення соціальних явищ або процесів макрорівня з дій і взаємодій агентів на мікрорівні. Агентна модель складається з трьох компонентів: агентів, середовища й правил<sup>1</sup>. Відповідно, теорія середнього рівня теж матиме таку структуру. Агентами можуть бути індивіди, домогосподарства, підприємства тощо. Причому в одній теорії середнього рівня можуть фігурувати кілька типів агентів, наприклад домогосподарства та окремі індивіди. Агентам притаманні ті або інші характеристики. Це можуть бути соціально-демографічні характеристики (наприклад стать, вік, рівень освіти); характеристики, котрі стосуються вподобань агентів, їхнього емоційного стану тощо. Середовище в теорії середнього рівня може бути як фізичним простором, що оточує агентів (наприклад місто, країна), так і виступати сукупністю зв'язків між ними. Ще один компонент теорії середнього рівня – правила. Вони можуть стосуватися взаємодій між агентами, їхніх дій, взаємодій між агентами й середовищем. Н. Гілберт і П. Терна (Gilbert, Terna, 2000) вважають, що правила складаються з умови, яка визначає, коли вони повинні виконуватися, та наслідку. Тобто вони мають вигляд твердження «якщо...то». Найчастіше правила прив'язані до певних характеристик агентів. І, звісно, у теорії середнього рівня згадується, які явища або процеси макрорівня будуть наслідком дій і взаємодій агентів на мікрорівні.

Перейдемо до формалізації теорії середнього рівня. Її можна визначити як процес створення агентної моделі з теорії середнього рівня. Перш ніж розпочати формалізацію теорії середнього рівня, потрібно сформулювати дослідницьке питання<sup>2</sup>. Очевидно, що воно повинно стосуватися соціальних явищ або процесів макрорівня, які потрібно пояснити. Дослідницьке питання в подальшому допоможе скерувати процес проведення комп'ютерних експериментів із моделлю<sup>3</sup>. Після цього варто спробувати знайти агентну модель, спроможну відповісти на дослідницьке питання, для якої виконується одна з двох умов:

1. Модель реалізовано в програмному середовищі, із яким досліднику зручно працювати.

2. Модель реалізовано в програмному середовищі, із котрим дослідник не знайомий. Однак є можливість працювати з нею в тому програмному середовищі, із яким дослідник знайомий. Наприклад, пакет «RNetLogo» уможливорює роботу з агентними моделями, реалізованими в NetLogo, у середовищі R.

Якщо агентна модель, здатна відповісти на дослідницьке питання, що відповідає одній із двох вищенаведених умов, існує, то потреба у формалізації теорії середнього рівня відпадає. Ми можемо перейти до проведення комп'ютерних експериментів з моделлю. В іншому разі здійснюємо формалізацію теорії середнього рівня. Цей процес розпочинається з того, що ми пропонуємо теорію

---

<sup>1</sup> Це впливає з визначення Е. Бруха і Дж. Атвелла, наведеного вище.

<sup>2</sup> На його важливості наголошують, зокрема, М. Сальгадо та Н. Гілберт (Salgado, Gilbert, 2013).

<sup>3</sup> Зі створеною агентною моделлю в подальшому проводяться комп'ютерні експерименти. Однак у межах цієї статті ми приділимо увагу лише процесу створення моделі.

середнього рівня. У деяких випадках можемо знайти таку теорію середнього рівня, яка дає підставу надати відповідь на дослідницьке питання й не потребує модифікацій. Ці модифікації можуть полягати, наприклад, у тому, що ми пропонуємо інші правила, замість наявних. Але якщо нам це не вдасться, потрібно сформулювати теорію середнього рівня. Це можна зробити:

1. Спираючись на наявні теорії середнього рівня. Так, ми можемо знайти одну теорію середнього рівня та модифікувати її (наприклад запропонувавши інші правила замість наявних). А можна знайти кілька теорій середнього рівня і на їх основі сформулювати власну.

2. Спираючись на описи агентних моделей, наведені в наукових публікаціях. Вони включають інформацію про агентів, середовище, правила, указують те, яких соціальних явищ чи процесів стосуються моделі. Також описи агентних моделей можуть містити математичні формули, значення характеристик агентів тощо. Можна знайти опис агентної моделі (один або декілька), що ляже в основу нашої теорії середнього рівня. Також при формулюванні власної теорії середнього рівня можна спиратись одночасно на наявні теорії середнього рівня та описи агентних моделей.

3. Самостійно, не спираючись ні на теорії середнього рівня, ні на описи агентних моделей.

Формулюючи теорію середнього рівня, не включаємо до неї якомога більше характеристик агентів чи правил. П. Смальдіно (Smaldino, 2020) вважає, що модель має бути простою й зосереджувати увагу на тих аспектах, які ми вважаємо найбільш важливими. Відповідно, теорія середнього рівня теж не повинна бути перевантажена зайвою інформацією.

Маючи теорію середнього рівня, можемо перейти до написання програмного коду. Передусім, треба обрати програмне середовище, де буде реалізовано модель. Ми зупинилися на програмному середовищі R. Найчастіше українські соціологи використовують його для статистичного аналізу даних. Однак ми розглянемо його як середовище для розробки агентних моделей. Агентна модель розглядатиметься як функція в R. Поняття функції має два значення. З одного боку, вона є програмним кодом, що включає параметри та набір команд, що дають змогу здійснювати операції з ними й складають тіло функції. Динамічною агентну модель можуть зробити цикли. Вони сприяють виконанню коду за декілька ітерацій, які є окремими часовими проміжками. Тож цикл виступатиме частиною тіла функції в R (ми звернемо увагу на цикл `repeat`)<sup>1</sup>. Надалі, говорячи про написання програмного коду, матимемо на увазі написання коду функції в R. З іншого боку, функція в R є об'єктом класу `function`. Указавши назву об'єкта класу `function`, підставивши ті чи інші значення параметрів, можемо дізнатися результати виконання функції. Отже, поговоримо про написання коду функції в R. Теорія середнього рівня та агентна модель, як уже згадувалося, складаються з трьох компонентів (агенти, середовище, правила), спрямовані на пояснення явищ або процесів макrorівня. Тож процес написання коду функції в R розглядатиметься для кожного з цих компонентів, а також охоплюватиме способи виведення результатів виконання функції. Розпочнемо з агентів (а саме з їхніх характеристик). Щоб представити їх у вигляді параметрів функції та/або набору команд у R, потрібно:

1. Виокремити всі характеристики, котрі містить теорія середнього рівня. Потрібно врахувати, що різні агенти можуть мати неоднаковий набір характеристик.

2. Визначити, яких значень набуватиме кожна з характеристик, і вказати шкалу вимірювання (номінальна, порядкова чи метрична).

3. З'ясувати, які характеристики визначатимуться на основі інших, представити формули їх розрахунку. Є характеристики, які залежать від розташування агентів у середовищі (наприклад кількість сусідів агента). Про них теж треба згадати.

4. Визначити, які характеристики набуватимуть однакових значень для всіх агентів, а які – ні.

5. Визначити, значення яких характеристик будуть сталими (тобто не змінюватимуться на кожному часовому проміжку), а яких – ні.

---

<sup>1</sup> Це умовний цикл, не прив'язаний до кількості ітерацій, як-от цикл `for`. Адже ми не завжди знаємо, як довго відбуватимуться зміни (наприклад, наскільки довго триватимуть переміщення агентів у середовищі). На відміну від циклу `while`, що теж є умовним, `repeat` може бути нескінченним. Урешті-решт, зміни (ті самі переміщення агентів у середовищі) можуть тривати нескінченно.

6. З'ясувати, які характеристики виступатимуть параметрами функції, які фігуруватимуть у тілі циклу, а які міститимуться в тілі функції, однак не в тілі циклу. Параметрами не можуть бути характеристики, значення яких розраховуються на основі інших. Характеристики, значення яких залежать від розташування агентів у середовищі, теж не можуть виступати параметрами моделі. У тіло циклу є сенс помістити характеристики, значення яких не будуть сталими. Для всіх характеристик агентів, що фігурують у тілі функції, треба окреслити набір команд, які дають змогу задати або розрахувати їхні значення. Для деяких характеристик, що містяться в тілі функції, можуть знадобитися параметри. Наприклад, є характеристика, яка позначає приналежність агентів однієї з двох груп. Для неї можна створити два параметри: кількість агентів у першій групі та кількість агентів у другій групі.

Перейдемо до наступного компонента – середовища. У цьому разі треба приділити увагу таким моментам:

1. Визначити спосіб представлення середовища. Як зазначають Ч. Макал і М. Норт (Macal, North, 2014), воно може мати вигляд  $n$ -вимірної решітки, графу (мережі), евклідового простору, ГІС. Ми більш детально розглянемо  $n$ -вимірну решітку (а саме – двовимірну решітку) і граф. У двовимірній решітці агентами виступають клітинки, у графі – вершини. Для побудови двовимірної решітки можна використати функцію *matrix*. У результаті буде отримано прямокутну матрицю з числами. Останні є значеннями однієї з характеристик агентів. Для створення мережі може стати в нагоді пакет «*igraph*». Отриманий граф є об'єктом класу *igraph*, що містить інформацію про вершини та зв'язки між ними.

2. Окреслити інформацію про початковий стан середовища. Потрібно визначити: 1) розмірність середовища. Тобто треба встановити розмірність двовимірної решітки або кількість вершин у графі; 2) розташування агентів у середовищі. Так, для графа варто визначити, між якими агентами існують зв'язки, а для двовимірної решітки потрібно встановити, у якій клітинці розміщуватиметься кожен з агентів. Якщо двовимірна решітка містить вільні клітинки (клітинки, де немає агентів), треба також визначити, де вони розташовуватимуться. Ця інформація може окреслюватися в параметрах моделі та/або задаватися в тілі функції, однак не в тілі циклу. Може бути так, що розмірність середовища й розташування агентів у ньому визначається в тілі функції. Однак існує кілька способів розміщення агентів, через що нам потрібен параметр, котрий дасть змогу «переключатися» між ними. Для інформації про початковий стан середовища, яка фігурує в тілі функції, треба задати набір команд в *R*.

3. Слід фіксувати інформацію про середовище на кожному наступному часовому проміжку. Це дасть змогу окреслити зміни, які відбуваються в ньому. Вони можуть стосуватися: 1) кількості агентів; 2) співвідношення агентів у групах (якщо агенти належать до кількох груп); 3) кількості вільних клітинок (якщо це двовимірна решітка, що містить вільні клітинки); 4) розташування агентів (це дасть змогу зобразити їхні переміщення). Інформація про середовище на кожному часовому проміжку зазначається в тілі циклу, для неї потрібно задати набір команд у *R*.

Перейдемо до правил. Для них окреслено таку послідовність дій:

1. Потрібно визначити всі правила, які є в теорії середнього рівня, а також установити, яких характеристик агентів стосуються правила.

2. Треба записати правила у вигляді математичних формул.

3. Потрібно визначити, чи є в моделі правила, що зупиняють зміни в часі. Вони виступатимуть умовою для завершення циклу *gereat*. Якщо такі правила відсутні, ми не вказуємо умову для циклу, тоді він стає нескінченним. Якщо ж нас така ситуація не влаштовує, можна вказати той чи інший часовий проміжок як умову для завершення циклу *gereat*.

4. Окреслити параметри для правил. Загалом, вони містяться або в тілі циклу, або є умовами для його завершення. Однак інколи ми маємо справу з кількома наборами правил, які не можуть виконуватись одночасно. У такому разі треба задати параметр – символічну змінну, значеннями якої будуть назви наборів правил, що дасть змогу «переключатися» між ними.

Ще один момент, який теж потрібно згадати – способи виведення результатів виконання функції. Саме це допомагає нам в подальшому робити висновки про поведінку складної системи. Ми можемо вивести результати у вигляді візуальних об'єктів та/або числових значень. Розпочнемо з першого способу. Створення візуалізації передбачає таку послідовність дій:

1. Треба з'ясувати, яким чином буде отримано візуалізацію. Відповідь на це запитання залежить від способу представлення середовища. Щоб зобразити двовимірну решітку, нам може стати в нагоді функція `image`. А для візуалізації графа можна використати функцію `plot` або ж функції з пакета «`gggraph`».

2. Потрібно визначити, для яких часових проміжків буде представлена візуалізація. Так, ми можемо зобразити лише початковий стан середовища і його стан на останньому часовому проміжку. Тоді набір команд у R для створення візуальних об'єктів фігуруватиме в тілі функції, однак не в тілі циклу. А можна візуалізувати початковий стан середовища і його стан на кожному наступному часовому проміжку. Тоді набір команд у R міститиметься і в тілі циклу, і в тілі функції (однак не в тілі циклу).

3. Потрібно звернути увагу на характеристики агентів, які відіграватимуть роль у створенні візуальних об'єктів. Ці характеристики не лише наводяться у вигляді числових значень, а й дають змогу маркувати агентів кольором та/або зображати їх у вигляді геометричних фігур.

4. Потрібно окреслити параметри, які стосуються представлення результатів виконання функції у вигляді візуальних об'єктів. Наприклад, ми хочемо самостійно встановлювати, коли виводити результати у вигляді візуальних об'єктів, а коли ні. Тоді треба створити параметр, який виступатиме логічною змінною. Значення `TRUE` дасть змогу вивести візуалізацію, а значення `FALSE` свідчатиме про те, що результати у вигляді візуальних об'єктів виводитися не будуть.

Перейдемо до другого способу представлення результатів виконання функції. Розрахунок числових показників здійснюється за такою схемою:

1. Потрібно виразити поняття, які позначають явище або процес макрорівня, у вигляді одного або кількох числових показників. Після цього вказати, як вони розраховуватимуться, яких значень набуватимуть, а також тип шкали.

2. Варто визначити, для яких часових проміжків розраховуватимуться показники. Це може бути, наприклад, лише останній часовий проміжок. Тоді набір команд в R для розрахунку числових показників наводиться в тілі функції, однак не в тілі циклу. Також числові показники можуть бути визначені на кожному часовому проміжку (у тому числі для часового проміжку, коли задається початковий стан середовища). Тоді набір команд у R наводитиметься в тілі функції, у тому числі й у тілі циклу.

3. Задати параметри, які стосуються виведення результатів, у вигляді числових показників. Наприклад, ми хочемо запропонувати два варіанти виведення результатів: 1) розрахунок числових показників на останньому часовому проміжку; 2) розрахунок числових показників на кожному часовому проміжку. Для цього потрібен параметр: символічна змінна, значення якої будуть назвами кожного з варіантів, що дасть змогу «переключатися» між ними.

Як бачимо, процес написання коду функції в R передбачає, у тому числі, запис тверджень теорії середнього рівня мовою математики. Так, ми визначаємо, яких значень набувають характеристики агентів і, за потреби, пропонуємо формули їх розрахунку; правила теж мають бути представлені у вигляді формул. Як уже згадувалося вище, сформульована нами теорія середнього рівня може спиратися на один або кілька описів агентних моделей. Якщо останні містять інформацію про значення характеристик агентів, формули, ми можемо використати їх (якщо це можливо та в цьому є потреба) для запису тверджень теорії середнього рівня мовою математики.

Однак процес формалізації теорії середнього рівня не завершується написанням програмного коду. Нам ще потрібно перевірити його на наявність помилок. Оскільки вище йшлося про написання коду функції в R, надалі зосередимося на виявленні й виправленні помилок у ньому. Умовно можна виокремити логічні та синтаксичні помилки<sup>1</sup>. Останні завжди супроводжуються повідомленням про помилку, через що результат виконання функції не виводиться. Інакше кажучи, наявність синтаксичних помилок свідчить про те, що ми написали «непрацюючу» функцію в R. Вони можуть спостерігатися під час:

---

<sup>1</sup> Загалом А. де Вріс та Дж. Мейс (De Vries, Meys, 2015) виокремлюють три типи помилок у R: синтаксичні, семантичні та логічні. Останні два стосуються випадків, коли ми написали «працюючий» код, де не виводиться повідомлення про помилку, однак отримані результати не відповідають очікуванням дослідника. Тож синтаксичні та логічні помилки було вирішено об'єднати під однією назвою – логічні помилки.

1. Виконання коду функції. У результаті має бути створено об'єкт класу `function`. Якщо під час цієї процедури R видає повідомлення про помилку, він не буде створений. У такому разі треба повернутися до коду функції, знайти помилки й виправити їх, після цього виконати його. Якщо R знову видає повідомлення про помилку, ми ще раз повертаємося до коду, виправляємо помилки та виконуємо його. Це повторюється доти, доки ми не отримаємо об'єкт класу `function`.

2. Спроби отримати результати виконання функції. Для цього потрібно вказати назву об'єкта класу `function`, підставити значення параметрів і виконати отриманий рядок коду. Якщо з'являється повідомлення про помилку, потрібно повернутися до коду функції, знайти помилки й виправити їх. У нагоді може стати функція `debug`, що дає змогу з'ясувати, де саме допущено помилки та в чому вони полягають. Виправивши їх, виконуємо код функції й слідуємо інструкціям, наведеним у пункті 1, після чого підставляємо до функції значення параметрів та виконуємо її. Якщо знову з'являється повідомлення про помилку, вносимо правки до коду функції. Процес повторюється доти, доки не буде виправлено всі синтаксичні помилки. У результаті отримаємо «працюючу» функцію в R, яка дасть змогу вивести результати у вигляді візуальних об'єктів і/або числових значень.

Однак «працююча» функція в R ще не є гарантією того, що її код не містить помилок. Наприклад, можемо зіткнутись із ситуацією, коли значення числових показників розраховуються неправильно. Це вказує на наявність логічних помилок, через які функція в R видає результати, що відрізняються від очікуваних. Для їх виправлення потрібно виконати такі дії:

1. Виконати код функції. У результаті буде створено об'єкт класу `function`.

2. Підставити до функції значення параметрів та виконати її. Бажано обирати ті значення параметрів, де результат виконання функції буде нескладно передбачити (Gilbert, Troitzsch, 2005, с. 22). Результати аналізуються, робиться висновок про те, чи містить код логічні помилки.

3. Якщо код функції містить логічні помилки, треба внести виправлення до нього. Потрібно враховувати, що такі правки можуть стати джерелом синтаксичних помилок. Якщо вони з'явилися після внесення виправлень до коду, потрібно їх усунути (про це йшлося вище).

Указану вище послідовність дій треба повторювати доти, доки не позбудемося всіх логічних помилок. Також R може виводити не лише повідомлення про помилку, а й попередження (`warning messages`). Вони потребують більш детального вивчення коду функції. У деяких випадках наявність попереджень ніяк не позначається на результатах. Однак існують ситуації, коли вони сигналізують про необхідність внесення правок до коду функції в R.

Отже, ураховуючи сказане вище, можемо перейти до виокремлення етапів формалізації теорії середнього рівня. Вони не універсальні й стосуються випадків, коли для реалізації агентної моделі обрано програмне середовище R, а агентну модель розглядаємо як функцію в R. Отже, процес формалізації теорії середнього рівня здійснюється за кілька етапів:

1. Специфікація моделі. Це етап, на якому ми пропонуємо теорію середнього рівня.

2. Представлення характеристик агентів у вигляді програмного коду в R. Спочатку ми визначаємо, яких значень набуватиме кожна з характеристик, тип шкали, пропонуємо формули розрахунку для них (за необхідності). А потім представляємо їх у вигляді параметрів та/або набору команд у тілі функції.

3. Представлення середовища у вигляді програмного коду в R. Спершу треба визначити спосіб представлення середовища (двовимірна решітка, граф тощо). Після цього потрібно задати його початковий стан і стан на кожному наступному часовому проміжку в параметрах і/або в тілі функції.

4. Представлення правил у вигляді програмного коду в R. Правила потрібно записати у вигляді формул, а потім у вигляді набору команд у тілі функції (якщо є необхідність, можна задати параметри для правил).

5. Створення візуалізації. Її вигляд залежить від способу представлення середовища. Для створення візуалізації слід задати набір команд у тілі функції й окреслити параметри.

6. Обчислення числових показників. Поняття, що позначає соціальне явище або процес макрорівня, потрібно представити у вигляді набору показників, визначити, яких значень вони набуватимуть, і тип шкали. Потім задати для них набір команд у тілі функції та окреслити параметри.

7. Виявлення й виправлення помилок у програмному коді в R. Ми маємо виправити синтаксичні та логічні помилки й проаналізувати попередження (warning messages), оскільки вони теж можуть сигналізувати про необхідність внесення правок до коду.

## ВИСНОВКИ

Ми розглянули агентне моделювання як метод теоретизування про складні системи. Воно дає змогу створювати агентні моделі, що виступають формальними теоріями, представленими у вигляді програмного коду. Для їх розробки потрібні вербальні теорії. Визначено, що теорії середнього рівня можуть бути представлені у вигляді агентних моделей. Вони є соціологічними теоріями, котрі дають змогу пояснити виникнення соціальних явищ або процесів макрорівня з дій і взаємодій агентів на мікрорівні. Теорія середнього рівня має таку структуру: агенти (треба зазначити, хто виступатиме в ролі агентів й окреслити їхні характеристики), середовище та правила. Розуміння того, які саме соціологічні теорії можуть бути представлені у вигляді агентних моделей, знання їхньої структури допоможе нам під час специфікації моделі (перший етап формалізації теорії середнього рівня). Виокремлені нами етапи формалізації теорії середнього рівня характеризують процес «перекладу» теорії середнього рівня мовою програмування R. У результаті отримаємо агентну модель, що матиме вигляд функції в R. Із нею в подальшому можна проводити комп'ютерні експерименти й робити висновки про поведінку складної системи. Однак треба враховувати, що виокремлені нами етапи не універсальні: вони можуть виявитися недоречними для інших мов програмування.

## ДЖЕРЕЛА ТА ЛІТЕРАТУРА

- Barbera, F. (2006). A Star is Born? The Authors, Principles and Objectives of Analytical Sociology. *Papers. Revista De Sociologia*, 80, 31–50. doi.org/10.5565/rev/papers/v80n0.1768
- Bruch, E., Atwell, J. (2015). Agent-Based Models in Empirical Social Research. *Sociological Methods & Research*, 44(2), 186–221. doi.org/10.1177/0049124113506405
- De Vries, A., Meys, J. (2015). *R for dummies*. Chichester: Wiley.
- Gilbert, N., Terna, P. (2000). How to build and use agent-based models in social science. *Mind & Society*, 1(1), 57–72. doi.org/10.1007/bf02512229
- Gilbert, N., Troitzsch, K. (2005). *Simulation for the Social Scientist*. Open University Press.
- Hedström, P., Bearman, P. (2017). What Is Analytical Sociology All About? An Introductory Essay, In: P. Hedström & P. Bearman (Eds.), *The Oxford Handbook of Analytical Sociology*. Oxford University Press, pp. 3–24. doi.org/10.1093/oxfordhb/9780199215362.013.1
- Hedström, P., Udehn, L. (2017). Analytical Sociology and Theories of the Middle-Range, In P. Hedström & P. Bearman (Eds.), *The Oxford Handbook of Analytical Sociology*. Oxford University Press, pp. 25–47. doi.org/10.1093/oxfordhb/9780199215362.013.2
- Hedström, P., Ylikoski, P. (2010). Causal mechanisms in the social sciences. *Annual Review of Sociology*, 36, 49–67. doi.org/10.1146/annurev.soc.012809.102632
- Macal, C., North, M. (2014). Introductory tutorial: Agent-based modeling and simulation. In: *Proceedings of the Winter Simulation Conference 2014*. IEEE, pp. 6–20. doi.org/10.1109/wsc.2014.7019874
- Salgado, M., Gilbert, N. (2013). Agent-Based Modeling. In: T. Teo (Ed.), *Handbook of Quantitative Methods for Educational Research*. Sense Publishers, pp. 247–265. doi.org/10.1007/978-94-6209-404-8\_12
- Smaldino, P. (2017). Models are stupid, and we need more of them. In: R. Vallacher, A. Nowak, & S. Read (Eds.), *Computational Social Psychology*, pp. 311–331. Routledge. doi.org/10.4324/9781315173726-14
- Smaldino, P. (2020). How to translate a verbal theory into a formal model. *Social Psychology*, 51(4), 207–218. dx.doi.org/10.1027/1864-9335/a000425
- Squazzoni, F. (2012). *Agent-Based Computational Sociology*. J. Wiley & Sons.