

ІІТМО

С.В. КОВАЛЬЧУК, И.И. БЕРЕГОВЕНКО

**МОДЕЛИРОВАНИЕ И ПРЕДСКАЗАНИЕ
ПОВЕДЕНИЯ АГЕНТОВ:
КОНСПЕКТ ЛЕКЦИЙ**



**Санкт-Петербург
2026**

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

УНИВЕРСИТЕТ ИТМО

С.В. Ковальчук, И.И. Береговенко

**Моделирование и предсказание поведения
агентов: конспект лекций**

учебное пособие

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлениям подготовки «Прикладная математика и информатика» и
«Программная инженерия» в качестве учебного пособия для реализации
основных профессиональных образовательных программ высшего образования
магистратуры

ИТМО

Санкт-Петербург

2026

Моделирование и предсказание поведения агентов: конспект лекций / С.В. Ковальчук, И.И. Береговенко – СПб: Университет ИТМО, 2026. – 211 с.

Рецензент: Костин Владимир Николаевич, доктор технических наук, доцент, доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем, Оренбургский государственный университет им. В.А. Бондаренко

Учебное пособие представляет собой конспект лекций о моделировании и предсказании поведения агентов, обзорно охватывающий широкий спектр подходов к описанию, анализу и построению агентных систем — от классических реактивных архитектур и многоагентного моделирования до обучения с подкреплением, когнитивных архитектур и методов работы с данными реального мира. Наряду с фундаментальными концепциями пособие знакомит читателя с прикладными инструментами и методами верификации и валидации моделей, формируя навыки их обоснованного выбора для конкретных задач. Пособие предназначено для студентов магистратуры, обучающихся по направлениям, связанным с искусственным интеллектом, вычислительными науками и прикладной математикой. Рекомендовано для применения в учебных курсах «Обучение с подкреплением», «Моделирование и предсказание поведения агентов финансовых систем», реализуемых в Университете ИТМО. Для успешного освоения материала достаточно базовой подготовки в области математики и программирования.



ИТМО (Санкт-Петербург) — национальный исследовательский университет, научно-образовательная корпорация. Альянс победителей международных соревнований по программированию. Приоритетные направления: ИТ и искусственный интеллект, фотоника, робототехника, квантовые коммуникации, трансляционная медицина, Life Sciences, Art&Science, Science Communication.

Лидер федеральной программы «Приоритет-2030», в рамках которой реализуется программа «Университет открытого кода». С 2022 ИТМО работает в рамках новой модели развития — научно-образовательной корпорации. В ее основе академическая свобода, поддержка начинаний студентов и сотрудников, распределенная система управления, приверженность открытому коду, бизнес-подходы к организации работы. Образование в университете основано на выборе индивидуальной траектории для каждого студента.

ИТМО пять лет подряд — в сотне лучших в области Automation & Control (кибернетика) Шанхайского рейтинга. По версии SuperJob занимает первое место в Петербурге и второе в России по уровню зарплат выпускников в сфере ИТ. Университет в топе международных рейтингов среди российских вузов. Входит в топ-5 российских университетов по качеству приема на бюджетные места. Рекордсмен по поступлению олимпиадников в Петербурге. С 2019 года ИТМО самостоятельно присуждает ученые степени кандидата и доктора наук.

© Университет ИТМО, 2026

© Ковальчук С.В., Береговенко И.И., 2026

Оглавление

Введение	6
Лекция 1. Введение в агентное моделирование: общие понятия ..	10
1.1. Что такое агентное моделирование?	10
1.2. Понятие агента	11
1.3. Свойства агента	12
1.4. Среда и её характеристики	13
1.5. Зачем нужно агентное моделирование? Цели и преимущества	17
1.6. Примеры: от простых правил к сложному поведению	18
1.7. Инструменты и технологии агентного моделирования	22
1.8. Ограничения и проблемы агентного моделирования	23
1.9. Заключение	24
Рекомендуемая литература и источники	25
Лекция 2. История и приложения агентного моделирования	27
2.1. Истоки: 1940–1960-е годы	27
2.2. Первые агентные модели: 1970-е годы	28
2.3. Институт Санта-Фе и сложные адаптивные системы: 1980-е годы	29
2.4. Эпоха вычислительных социальных наук: 1990-е годы	30
2.5. Зрелость и гибридизация: 2000-е годы — настоящее время	31
2.6. Области применения агентного моделирования	32
2.7. Тенденции и перспективы	40
2.8. Заключение	40
Рекомендуемая литература и источники	41
Лекция 3. Архитектура и характеристики агентов	43
3.1. Введение: агент в среде и понятие задачи	43
3.2. Понятие архитектуры агента	45
3.3. Реактивные агенты	46
3.4. Реактивные агенты с внутренним состоянием	50
3.5. Рассуждающие агенты	53
3.6. Гибридные архитектуры	60
3.7. Заключение	63
Рекомендуемая литература и источники	64
Лекция 4. Перцепция, действия и решения агентов	66
4.1. Перцепция агентов	66
4.2. Психологические модели перцепции	69
4.3. Действия агентов	71
4.4. Связанные свойства среды	72
4.5. Принятие решений агентом	74
4.6. Рациональность агента	75
4.7. Заключение	79
Рекомендуемая литература и источники	80
Лекция 5. Многоагентные системы	81
5.1. Понятие и ключевые свойства многоагентных систем	81
5.2. Коммуникация агентов	82

5.3. Структурирование агентов: группы, роли, организация, среда	84
5.4. Механизмы координации	86
5.5. Имитационное моделирование многоагентных систем	87
5.6. Кросс-дисциплинарность многоагентного моделирования	91
5.7. Инструментальные средства многоагентного моделирования	91
5.8. Классические примеры многоагентных моделей	92
5.9. Заключение	96
Рекомендуемая литература и источники	97
Лекция 6. Сложность, эмерджентность и коллективное поведение	98
6.1. Классы сложности: от Уивера к современным подходам	98
6.2. Свойства сложных систем	99
6.3. Квадранты риска по Нассиму Талебу	101
6.4. Сложные адаптивные системы	102
6.5. Эмерджентность: определение и механизмы	103
6.6. Примеры эмерджентного поведения в многоагентных моделях	104
6.7. Коллективное поведение: толпы, кооперация и конкуренция, управление	107
6.8. Заключение	109
Рекомендуемая литература и источники	110
Лекция 7. Введение в обучение с подкреплением	112
7.1. Обучение с подкреплением и агентное моделирование	112
7.2. Формальная постановка задачи	113
7.3. Марковские процессы принятия решений	114
7.4. Ценность, вознаграждение и стратегия	115
7.5. Уравнение Беллмана и оптимальные стратегии	117
7.6. Модель среды	118
7.7. Методы обучения с подкреплением	118
7.8. Глубокое обучение с подкреплением	122
7.9. Многоагентное обучение с подкреплением	124
7.10. Обучение с подкреплением на основе обратной связи от человека	126
7.11. Границы применимости и открытые вопросы	127
7.12. Заключение	128
Рекомендуемая литература и источники	129
Лекция 8. Модели принятия решений	130
8.1. Принятие решений как центральная задача агентного моделирования	130
8.2. Рациональный агент и максимизация ожидаемой полезности	130
8.3. Ограниченная рациональность: «ножницы поведения»	131
8.4. Многокритериальные оценки	133
8.5. Деревья принятия решений	136
8.6. Ранжирование, голосование и парадокс Кондорсе	137
8.7. Теоретико-игровые подходы	139
8.8. Дизайн механизмов	142
8.9. Модели общественного мнения	144
8.10. Заключение	146
Рекомендуемая литература и источники	147
Лекция 9. Моделирование поведения агентов реального мира ...	149
9.1. Философия и принципы моделирования	149

9.2. Иерархия моделей и компоненты	151
9.3. Типология агентных моделей: от абстрактных к эмпирическим	152
9.4. Имитационное моделирование: парадигмы и уровни абстракции	154
9.5. Работа с данными в контексте агентного моделирования	155
9.6. Усвоение данных и концепция DDDAS	156
9.7. Верификация и валидация агентных моделей	159
9.8. Прозрачность и интерпретируемость моделей	161
9.9. Заключение	163
Рекомендуемая литература и источники	164
Лекция 10. Когнитивные состояния: модели и архитектуры	165
10.1. Искусственный интеллект общего назначения и проблема оценки интеллекта	165
10.2. Уровни когнитивных способностей агентов	168
10.3. Когнитивные архитектуры	168
10.4. Взаимодействие человека и агента ИИ	171
10.5. Клонирование поведения и обратное обучение с подкреплением	172
10.6. Модель BDI и Theory of Mind	174
10.7. Моделирование иррационального поведения	176
10.8. Заключение	178
Рекомендуемая литература и источники	179
Лекция 11. Имитационное моделирование: подходы, инструменты, данные	180
11.1. Терминологические основы: модели и симуляции	180
11.2. Фундаментальный цикл моделирования и симуляции	180
11.3. Дилемма точности и стоимости: «все модели неверны»	181
11.4. Приложения имитационного моделирования	182
11.5. Роли имитационного моделирования в аналитическом процессе	183
11.6. PARTE-фреймворк: архитектура агентной модели	185
11.7. Лучшие практики разработки имитационных моделей	187
11.8. Инструменты для агентного моделирования	188
11.9. Заключение	190
Рекомендуемая литература и источники	190
Лекция 12. Открытые проблемы и перспективы агентного моделирования	192
12.1. Технические ограничения	192
12.2. Концептуальные ограничения	194
12.3. Критичные направления	195
12.4. Перспективы развития	198
12.5. Заключение	200
Рекомендуемая литература и источники	201
Заключение	202
Приложение 1. Вопросы для самопроверки и тестирования	204
Приложение 2. Вопросы для проведения экзамена	211

Введение

Понятие агента занимает одно из центральных мест в современной науке о сложных системах. Будь то человек в городской среде, клетка в биологической ткани, программный бот в цифровой сети или автономный робот на производстве — в каждом случае речь идёт о некоторой сущности, которая воспринимает своё окружение и целенаправленно действует в нём. Эта единая интуиция лежит в основе нескольких крупных научных направлений: разработки интеллектуальных агентов и систем на знаниях, вычислительного моделирования многоагентных и сложных систем, машинного обучения (в особенности обучения с подкреплением), а также робототехники. При всём единстве исходной идеи каждое из этих направлений расставляет свои акценты: одно сосредоточено на внутренней когнитивной архитектуре агента, другое — на коллективном поведении и эмерджентных свойствах системы, третье — на оптимизации стратегии через взаимодействие со средой, четвёртое — на физическом воплощении и управлении реальным устройством.

Классическое определение агента, восходящее к фундаментальным работам в области искусственного интеллекта, описывает его как автономную сущность, двунаправленно взаимодействующую со средой: воспринимающую её через сенсоры и воздействующую на неё через эффекторы. Настоящий курс опирается на расширение этого определения, акцентирующее два дополнительных измерения агентного поведения. Первое — управление ресурсами: агент действует в условиях ограниченных индивидуальных и коллективных ресурсов (вычислительных, энергетических, временных, информационных), и именно эти ограничения определяют его стратегию, горизонт планирования и выбор между рациональным и приближённым решением. Второе — коммуникация: агент существует не в изоляции, а в пространстве взаимодействий с другими агентами, которые могут реализовываться как попарный обмен сообщениями или широковещательная рассылка, требуя определённого протокола, языка и семантики. Оба измерения пронизывают все темы курса — от реактивных архитектур и многоагентных систем до теоретико-игровых моделей и обучения с подкреплением.

Особую роль в рамках агентного подхода играет человек — специфический агент, занимающий одновременно несколько позиций в прикладных задачах. Он может быть объектом анализа, чьё поведение

агентная модель стремится воспроизвести или предсказать; источником экспертного знания, формирующим функцию вознаграждения или оценочные критерии; пользователем или партнёром искусственного агента, совместно с которым решается задача. В последнем случае — и это одно из наиболее перспективных направлений современных исследований — речь идёт уже не просто об имитации или поддержке человека, а о коэволюции человека и агента ИИ, при которой эффективность системы превышает возможности каждого из её компонентов по отдельности. Понимание роли человека в агентных системах является сквозной темой курса: от моделей ограниченной рациональности и когнитивных искажений до обратного обучения с подкреплением и Theory of Mind.

Расширенная задача настоящего учебного пособия состоит в том, чтобы выстроить содержательный мост между фундаментальными концептуальными представлениями об агентах и конкретными методами, технологиями и инструментами, используемыми в различных прикладных областях. Один и тот же агентный подход находит воплощение в имитационном моделировании эпидемий и транспортных потоков, в алгоритмах глубокого обучения с подкреплением, в когнитивных архитектурах BDI и SOAR, в механизмах дизайна рынков и аукционов, в системах поддержки принятия решений с учётом человеческого фактора. Курс стремится показать, что за видимым разнообразием методов стоит единая понятийная структура: агент, среда, ресурсы, коммуникация, цель.

Центральной методологической идеей курса является подход «снизу вверх» (bottom-up): сложные глобальные закономерности — пространственные паттерны, коллективные феномены, устойчивые структуры — возникают из простых локальных правил поведения отдельных агентов. Именно эта способность порождать эмерджентное поведение отличает агентный подход от классических математических моделей, описывающих систему «сверху вниз» через агрегированные уравнения. Курс стремится показать, в каких ситуациях агентный взгляд необходим и незаменим, а в каких традиционные методы справляются лучше.

Пособие организовано в двенадцать лекций, последовательно раскрывающих тему с нарастающей глубиной. Первые две лекции формируют общий понятийный фундамент: даётся определение агента и среды, обсуждаются свойства агентов и история возникновения агентного моделирования. Лекции 3 и 4 посвящены внутреннему устройству агентов — архитектурам поведения (реактивным,

рассуждающим и гибридным), а также механизмам перцепции, действий и принятия решений. В лекции 5 рассматриваются многоагентные системы — принципы коммуникации, координации и организации коллективного поведения. Лекции 6 и 7 вводят теоретические инструменты анализа: теорию сложных систем и эмерджентности, а также формализм обучения с подкреплением как механизма адаптации агентов. Лекция 8 охватывает модели принятия решений — от рационального выбора и ограниченной рациональности до теоретико-игровых подходов и механизмов голосования. Лекции 9–11 посвящены практической стороне моделирования: работе с данными реального мира, верификации и валидации моделей, а также инструментальному обзору платформ имитационного моделирования. Лекция 10 особо рассматривает когнитивные архитектуры, модели BDI и Theory of Mind, клонирование поведения и иррациональность — темы, на пересечении которых находится роль человека как специфического агента. Завершает курс лекция 12, в которой анализируются открытые проблемы и перспективы развития области.

Пособие адресовано студентам, обладающим базовой математической подготовкой и общими представлениями об алгоритмах и программировании. Знакомство с теорией вероятностей, основами линейной алгебры и элементами математического анализа является желательным, однако большинство концепций излагается на содержательном уровне, доступном без глубокой формальной базы. Там, где это возможно, математические определения сопровождаются содержательными интерпретациями, конкретными примерами и иллюстрациями из реальных приложений.

Каждая лекция конспекта содержит список рекомендуемой литературы и источников, позволяющих углубиться в отдельные темы. В приложениях приведены тестовые вопросы для самопроверки и экзаменационные вопросы, охватывающие ключевые понятия курса. Пособие может использоваться как для последовательного изучения курса, так и для выборочного обращения к отдельным темам — каждая лекция в значительной мере самодостаточна, хотя и опирается на понятийный аппарат, введённый в предшествующих разделах.

При построении курса использовались материалы научных и учебных изданий, публикаций по теме курса. Материалы, используемые напрямую (в частности, иллюстрации) даются со ссылкой на источник (свободное использование в учебных целях в соответствии со статьей 1274 ГК РФ).

Авторы подтверждают, что при подготовке рукописи инструменты искусственного интеллекта использовались исключительно для литературного редактирования, проверки грамматики и коррекции стиля. Весь содержательный контент, логика изложения и научные выводы созданы автором самостоятельно.

Лекция 1. Введение в агентное моделирование: общие понятия

1.1. Что такое агентное моделирование?

Агентное моделирование (**agent-based modeling, ABM**) — это вычислительный метод исследования сложных систем, основанный на создании и анализе поведения множества автономных взаимодействующих сущностей — агентов — в определённой среде. В отличие от классических подходов, описывающих систему как единое целое через дифференциальные уравнения или статистические агрегаты, агентное моделирование реализует методологию «снизу вверх» (*bottom-up*): глобальные закономерности, паттерны и структуры проявляются из простых локальных правил, которым следуют отдельные агенты.

Вместе с тем агентное моделирование — это не просто вычислительный инструмент, но принципиально новый способ мышления о сложных системах. Его философской основой служит синтез двух, казалось бы, противоположных идей: холизма — идеи о том, что целое больше суммы своих частей — и редукционизма, стремящегося объяснить сложные феномены через взаимодействие элементарных компонентов. Этот парадоксальный синтез и находит воплощение в агентном подходе: мы описываем простые правила поведения отдельных агентов и наблюдаем, как из них возникают сложные коллективные явления.

Исторически агентное моделирование возникло как ответ на ограничения классических подходов для систем, обладающих свойствами сложности: нелинейностью, гетерогенностью элементов, наличием обратных связей и способностью к адаптации. Именно такие системы — от экосистем и социальных структур до городского транспорта и рынков — становятся всё более актуальными объектами научного исследования. При этом агентное моделирование находит применение в трёх широких дисциплинарных направлениях.

Первое направление — численное (вычислительное) моделирование (*computational science*), ориентированное на описание и понимание существующих систем — природных, социальных или технических. Здесь особое место занимает многоагентное моделирование (*multi-agent simulation, MAS*), в котором изучается коллективное поведение большого числа агентов. Второе —

искусственный интеллект (AI), где акцент делается на внутреннем устройстве агента: его когнитивных способностях, логике рассуждений, механизмах принятия решений. Здесь широко применяется понятие интеллектуального агента, в том числе в рамках обучения с подкреплением (reinforcement learning). Третье направление — робототехника, где агентные модели интегрируются с физическими системами и учитывают реальные механические ограничения.

Несмотря на различие точек зрения, все эти направления сходятся в базовом интуитивном понимании того, что такое агент. Будь то человек в городе, птица в стае, программный бот в интернете или автономный робот — в каждом случае речь идёт о некой сущности, которая действует в определённом окружении, воспринимает его и реагирует на него.

1.2. Понятие агента

Центральным понятием агентного моделирования является агент (agent). Несмотря на то, что различные дисциплины трактуют его по-разному, можно сформулировать общее определение: **агент** — это автономная сущность, действующая в рамках среды, воспринимающая своё окружение через сенсоры, принимающая решения на основе внутренней логики и совершающая целенаправленные действия посредством эффекторов.

Это определение включает несколько ключевых элементов. Во-первых, агент и среда — две неразрывно связанные сущности. Агент не может существовать «сам по себе»: он всегда помещён в некое окружение, от которого получает сигналы и на которое оказывает воздействие. Во-вторых, агент обладает перцепцией — способностью воспринимать среду. Инструменты восприятия называются сенсорами. Наконец, в-третьих, агент воздействует на среду посредством действий, реализуемых через эффекторы (или актуаторы). Между восприятием и действием находится внутренняя логика агента — именно она определяет, как агент будет реагировать на те или иные сигналы.

Онтологический статус агентов чрезвычайно разнообразен. В роли агентов могут выступать биологические сущности (люди, животные, насекомые, клетки, микроорганизмы), технические системы (роботы, беспилотники, программные боты), а также социальные конструкторы (семья, группа, компания). Это разнообразие отражает универсальность агентной парадигмы.

Рисунок 1.1 схематично иллюстрирует взаимодействие агента и среды в терминах сенсоров, эффекторов и внутренней логики.

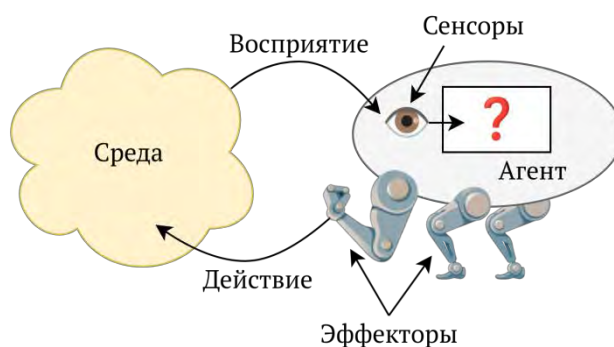


Рисунок 1.1 — Схема взаимодействия агента и среды: сенсоры воспринимают среду, внутренняя логика формирует решения, эффекторы осуществляют действия

1.3 Свойства агента

Агентов принято характеризовать через совокупность свойств. Ниже приведены ключевые свойства, встречающиеся во многих подходах к агентному моделированию (см. также систематизацию в таблице 1.1).

Особого внимания заслуживает свойство ограниченности агента. В отличие от «всеведущего» наблюдателя, агент воспринимает лишь некоторую окрестность среды — определяемую его сенсорными возможностями и структурой самой среды. Это ограничение области видимости и качество перцепции вносят принципиальную неопределённость в поведение агента. Именно так устроены и реальные агенты: ни человек, ни животное не видит бесконечно далеко; программный бот работает в рамках ограниченного информационного контекста. Такая ограниченность — не недостаток модели, а её реалистичное свойство.

Эмерджентное поведение — пожалуй, самое важное и специфическое свойство многоагентных систем. Под эмерджентностью понимается возникновение на уровне всей системы новых качеств и паттернов, которые не присущи отдельным агентам и не предусмотрены явно в правилах их поведения. Это явление подробнее рассматривается в лекции 6.

Таблица 1.1 — Свойства агента

Свойство	Описание	Применимость
Автономность	Самостоятельные решения на основе внутренних состояний	Все агентные системы
Адаптивность	Изменение поведения под влиянием среды	Все агентные системы
Перцепция	Восприятие окружающей среды через сенсоры	Все агентные системы
Принятие решений	Выбор действий на основе внутренней логики и сигналов	Все агентные системы
Интерактивность	Двунаправленное взаимодействие со средой и агентами	Все агентные системы
Эмерджентное поведение	Возникновение системных свойств из локальных взаимодействий	Многоагентные системы
Обучение	Корректировка внутренних правил на основе опыта	Адаптивные системы
Целеполагание	Ориентированность на достижение цели или максимизацию полезности	Целеустремленные агенты
Коммуникация	Обмен информацией между агентами напрямую или через среду	Многоагентные системы
Гетерогенность	Различие характеристик и правил поведения разных агентов	Многоагентные системы
Самоорганизация	Координация без централизованного управления	Многоагентные системы
Динамика	Изменение состояния во времени (дискретном или непрерывном)	Все агентные системы
Ограниченность	Локальность восприятия, неопределённость относительно среды	Все агентные системы

1.4. Среда и её характеристики

Второй фундаментальный элемент агентной системы — **среда** (*environment*), то есть окружение агента, в котором он существует и действует. Понимание структуры и свойств среды не менее важно, чем понимание самого агента: именно среда определяет контекст, в котором разворачивается поведение.

Прежде всего среды подразделяются на два принципиальных класса. Физические среды обладают явной пространственной

структурой и геометрией — это характерно для моделей городской среды, транспортных систем, робототехнических приложений и экологических моделей. Информационные среды лишены традиционной пространственной размерности; они организованы в виде сетей или других абстрактных структур — как в моделях социальных сетей, компьютерных систем или рынков.

Ключевые характеристики среды можно перечислить следующим образом.

Пространственная структура и топология — описание того, как устроена среда: какова позиция агента в ней, что является его окружением, как это окружение меняется при перемещении агента. В простейшем случае это регулярная сетка (как в клеточных автоматах); более сложные среды включают иерархические структуры или масштабно-инвариантные сети. Связанность среды — степень соединённости её различных частей — существенно влияет на динамику распространения информации и ресурсов.

Распределение ресурсов — характеристика того, как распределены в среде объекты и ценности, существенные для агентов. Ресурсы могут быть равномерными или локально сконцентрированными, возобновляемыми или истощаемыми. В экологических моделях трава, которую поедают животные, постепенно восстанавливается; в экономических — ресурсы могут быть строго ограничены.

Доступность для агентов — возможности перемещения, навигации и взаимодействия. В физических средах перемещение может быть ограничено препятствиями или особенностями ландшафта; в информационных — скоростью передачи данных или правами доступа.

Динамика и изменчивость во времени — среда может меняться как под воздействием агентов, так и в результате собственных внутренних процессов. Время может быть представлено дискретно (пошаговые итерации) или непрерывно.

Границы среды — вопрос о том, что произойдёт с агентом, достигшим края среды: он может остановиться, отразиться или «перейти» на противоположный край (топология тора).

Механизм воздействия и «видимость» — среда влияет на агента, изменяя его сенсорный ввод, а «видимость» среды (то, насколько далеко и полно агент её воспринимает) зависит как от устройства агента, так и от самой среды.

Пространственная неоднородность — неравномерное распределение физических и информационных характеристик.

Например, в городской среде разные районы имеют разную пропускную способность для транспортных потоков.

Ёмкость среды — предельное количество агентов, которое она может поддерживать. Перенаселение в биологических и социальных моделях может приводить к кризисным явлениям.

Примеры разнообразных топологий сред представлены на рисунке 1.2. В зависимости от топологии и размерности выделяют: (а) «безразмерную» (aspatial) среду — абстрактное хранилище агентов, где пространственные отношения не имеют значения; (b) клеточные автоматы (cellular automata) — регулярную сетку с дискретными состояниями ячеек; (c) евклидово двумерное пространство — наиболее распространённый вариант для теоретических моделей; (d) геоинформационную систему (GIS) — среду, моделирующую реальное географическое пространство; (e) сетевую топологию — граф взаимодействий, типичный для информационных агентов.

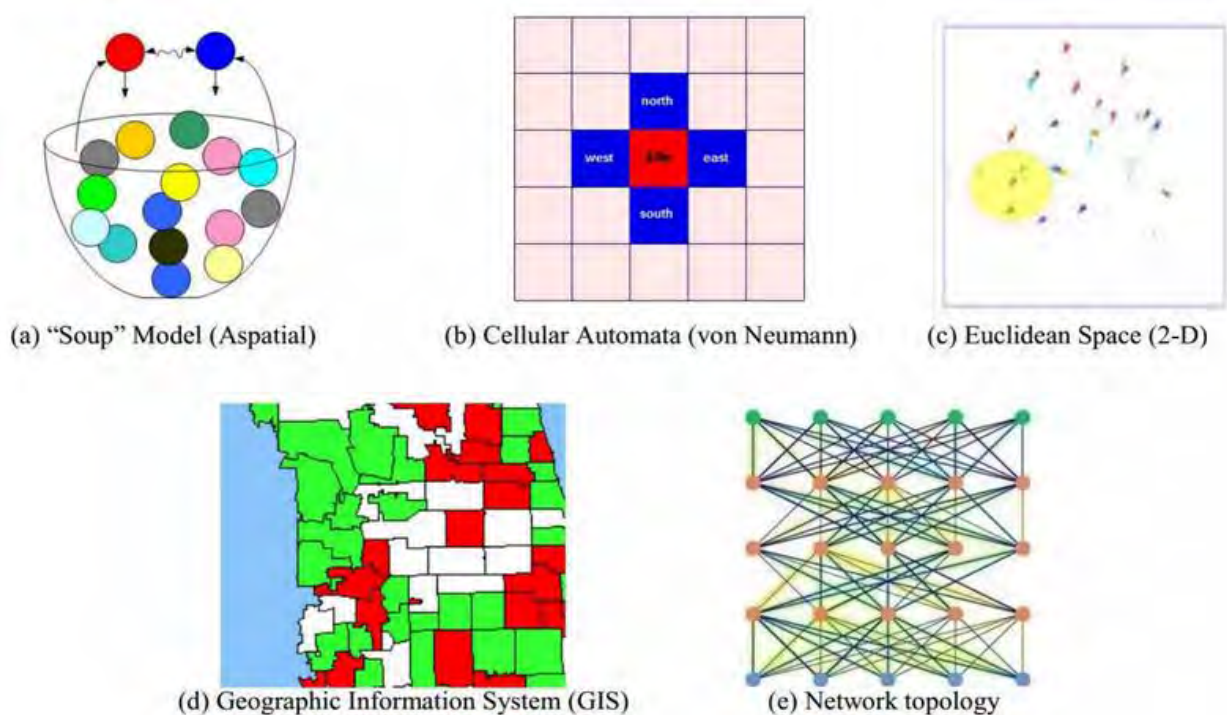


Рисунок 1.2 — Варианты топологии сред агентного моделирования:

- (а) безразмерная среда, (b) клеточные автоматы (фон Нейман),
- (c) евклидово 2D-пространство, (d) геоинформационная система,
- (e) сетевая топология (Источник: [9])

Взаимодействие между агентами и средой может принимать различные формы. Можно выделить четыре базовых сценария (рисунок 1.3), каждый из которых находит применение в определённых классах задач.

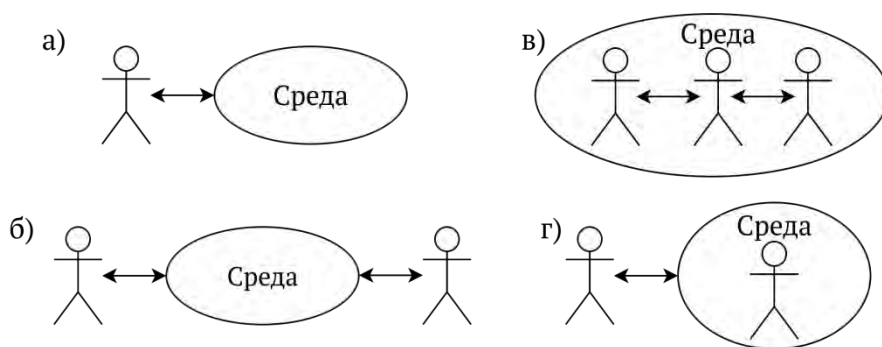


Рисунок 1.3 - Сценарии взаимодействия агентов и среды

Сценарий 1. Изолированный агент. Один агент действует автономно в среде, не взаимодействуя с другими агентами. Всё его внимание сосредоточено на восприятии и изменении среды. Типичные примеры — автономный робот, исследующий незнакомую местность, или программный агент, собирающий информацию в распределённой сети.

Сценарий 2. Среда как посредник. Несколько агентов не взаимодействуют напрямую, но влияют друг на друга через изменения в общей среде. Классический пример — ресурсные модели: травоядные животные конкурируют за растительность, при этом действия одного агента изменяют среду и косвенно влияют на поведение других.

Сценарий 3. Среда = совокупность агентов. Крайний случай, когда пассивной «фоновой» среды нет — все сущности являются активными агентами. Примеры: клеточные автоматы, где каждая клетка — простейший агент; сети ботов, где взаимодействуют только программные агенты. Все свойства системы целиком определяются правилами взаимодействия.

Сценарий 4. Среда населена агентами-оппонентами. Один основной агент взаимодействует со средой, которая сама содержит других агентов. В отличие от сценария 2, здесь подчёркивается асимметрия: агенты среды могут быть оппонентами или нейтральными элементами. Характерные примеры — системы поддержки принятия решений (СППР) и чат-боты, взаимодействующие с пользователями.

В реальных приложениях часто встречаются гибридные сценарии, сочетающие элементы нескольких базовых конфигураций. Например, в модели городского транспорта могут присутствовать как агенты-водители, непосредственно взаимодействующие друг с другом (сценарий 3), так и элементы инфраструктуры в роли среды (сценарий 2). Гибкость агентного подхода позволяет комбинировать различные типы взаимодействий для наиболее адекватного моделирования реальных систем.

1.5. Зачем нужно агентное моделирование? Цели и преимущества

Прежде чем разбирать технические детали, полезно ответить на вопрос: зачем вообще использовать агентное моделирование, если существуют традиционные математические подходы? Ответ связан с принципиальными различиями между моделированием «на основе уравнений» (equation-based modeling, EBM) и агентным подходом.

Классически выделяют три типа моделирования по функциональному назначению: описательное (descriptive) — понимание текущего состояния системы и механизмов её работы; предсказательное (predictive) — прогнозирование будущих состояний и поведения; рекомендательное (prescriptive) — выработка управляющих воздействий и оптимизация. Агентное моделирование применимо во всех трёх режимах, причём характер его применения существенно зависит от типа изучаемой системы — существующей (естественной или социальной) или создаваемой (технической) - см. Таблицу 1.2.

Таблица 1.2 — Применение агентного моделирования
в различных задачах

Тип моделирования	Описательное	Предсказательное	Рекомендательное
Существующие системы	Понимание естественных процессов; исследование сложных систем; эмерджентное поведение	Прогнозирование состояния; сценарный анализ «что, если»	Оптимизация регуляции (policy making)
Создаваемые системы (роботы, программные агенты)	Аналитика поведения агентов	Оценка эффективности решений; прогнозирование поведения агентов	Оптимизация системы агентов; обучение агентов

Для существующих систем агентный подход позволяет описывать сложные феномены, изучать эмерджентное поведение, проводить контрфактуальный анализ («что было бы, если бы...»), а также разрабатывать меры регулирования через изменение среды или введение нормативных ограничений — не «залезая в голову» каждому агенту, а воздействуя на условия их взаимодействия.

Для создаваемых систем (роботов, программных агентов, автономного транспорта) агентное моделирование даёт возможность проектировать и тестировать поведение ещё до создания физического

прототипа, оптимизировать алгоритмы управления и прогнозировать взаимодействие агентов в реальных сценариях. Таблица 1.3 представляет сравнение агентного моделирования с уравненно-ориентированным подходом по ключевым критериям.

Таблица 1.3 — Сравнение уравненного (EBM) и агентного (ABM) моделирования (Источник: адаптировано по [2])

Критерий	Уравнения (EBM)	Агентное моделирование (ABM)
Теоретическое обоснование	Высокое, теория-ориентированное	Низкое, данные-ориентированное
Направление подхода	Сверху вниз (top-down), агрегированное	Снизу вверх (bottom-up), дезагрегированное
Выражение	Математические уравнения	Язык программирования
Методы	Статистика	Нейросети, анализ больших данных
Вычислительные требования	Минимальные	Высокие
Сетевая структура	Нет	Есть
Предположения об агентах	Однородные	Гетерогенные
Взаимодействие	Нет / неявное	Есть / явное
Представление времени	Непрерывное	Дискретное
Подходящая область	Простые, физически-ориентированные	Сложные, локальные решения

Ключевое преимущество агентного подхода — способность описывать гетерогенные системы с нелинейными взаимодействиями, которые сложно или невозможно адекватно описать уравнениями. Благодаря гетерогенности и динамичности агентов можно смоделировать ситуации, уравнения для которых либо чрезвычайно сложно решить, либо вообще нельзя составить аналитически.

1.6. Примеры: от простых правил к сложному поведению

Одной из наиболее впечатляющих особенностей агентного моделирования является способность простых локальных правил порождать сложные и неожиданные глобальные паттерны — то, что

называется эмерджентностью. Рассмотрим несколько классических примеров.

Модель сегрегации Шеллинга. Одна из первых агентных моделей — модель сегрегации Томаса Шеллинга (1971), реализованная на клеточном автомате. В ней агенты двух типов (условно «красные» и «зелёные») распределены по клеткам решётки. Правило поведения предельно простое: если число соседей того же цвета составляет менее 30%, агент перемещается в случайную свободную клетку. Стартовое состояние — полностью случайное перемешивание. Интересен результат, показанный на рисунке 1.4: даже при такой слабой «предпочтительности» (всего 30%) со временем формируются чётко выраженные однородные кластеры — сегрегированные районы. Это эмерджентная структура: она не была задана явно, но возникла из элементарного правила.

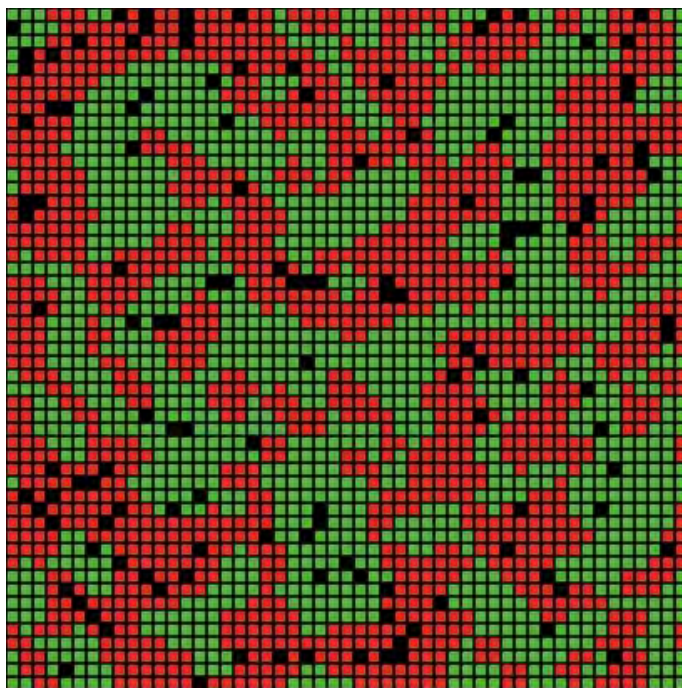


Рисунок 1.4 — Результат работы модели сегрегации Шеллинга: из случайного распределения двух типов агентов возникают устойчивые кластеры (Источник: [10], онлайн доступ:

<https://dewdis.github.io/schelling/>)

Практическая ценность модели оказалась весьма значительной: исследователи успешно применяли её для объяснения расслоения жителей в крупных американских городах (Чикаго, Нью-Йорк, Лос-Анджелес). Наиболее выраженная сегрегация была обнаружена по уровню достатка: люди стремятся жить в окружении похожих на себя, и это простое психологическое правило порождает наблюдаемые пространственные паттерны.

Модель Boids: стая как самоорганизующаяся система. Другой классический пример — модель Boids, предложенная Крейгом Рейнольдсом в 1986 году. Она воспроизводит коллективное движение стаи птиц или косяка рыб — явление, известное как мурмурация. Каждый агент (boid - «bird-oid», «птицеподобный») следует трём простым правилам: Сепарация (separation) — держаться на минимальном расстоянии от ближайших соседей, чтобы не столкнуться; Выравнивание (alignment) — двигаться в том же направлении, что и соседи; Сближение (cohesion) — держаться вместе, стремиться к центру группы.

Из этих трёх правил в двумерной или трёхмерной среде возникают плавные, текучие паттерны движения, визуально напоминающие реальные стаи птиц. Описать подобное поведение математическими уравнениями крайне затруднительно, тогда как агентная модель делает это элементарно. Примечательно, что схожие паттерны наблюдаются в реальной природе — это свидетельство того, что эмерджентность характерна не только для искусственных моделей, но и для живых систем.

Клеточные автоматы Вольфрама и эмерджентность. Стивен Вольфрам в книге «A New Kind of Science» [6] систематически исследовал поведение одномерных клеточных автоматов. В простейшем случае рассматривается среда, состоящая из линии ячеек, каждая из которых может находиться в состоянии 0 или 1. Следующее состояние каждой ячейки определяется её текущим состоянием и состоянием двух соседей. Поскольку комбинация трёх бит даёт 8 вариантов, а каждому можно сопоставить 0 или 1, существует ровно $2^8 = 256$ различных «правил». Вольфрам пронумеровал их и обнаружил четыре класса поведения. Класс 1 — постоянное поведение: система быстро приходит к однородному состоянию (все нули или все единицы). Класс 2 — повторяющееся поведение: возникают простые периодические структуры. Класс 3 — случайное (хаотическое) поведение: паттерны выглядят псевдослучайными. Класс 4 — сложное поведение: возникают устойчивые локализованные структуры, взаимодействующие сложным образом. Правило 30 (рисунок 1.5) — один из наиболее известных примеров класса 4. Запущенное из одной закрашенной клетки, оно порождает треугольный паттерн с характерными самоподобными структурами. Что особенно примечательно — практически идентичные узоры обнаруживаются на раковинах некоторых моллюсков, формирующихся в результате биологических процессов. Это ещё одно

подтверждение того, что эмерджентные паттерны, открытые в компьютерных моделях, находят аналоги в живой природе.

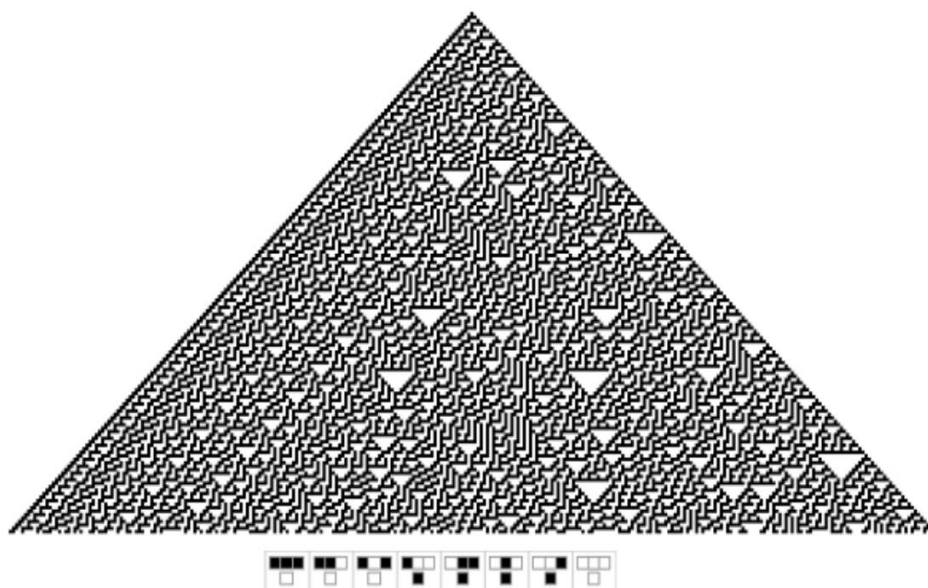


Рисунок 1.5 — Правило 30 Вольфрама: из одной закрашенной клетки возникает сложный треугольный паттерн с самоподобными структурами (Источник: [6])

Эмерджентные пробки в транспортных потоках. Ещё один наглядный пример — модель транспортного потока на кольцевой дороге (рисунок 1.6). Автомобили движутся с примерно одинаковой, но случайно варьирующейся скоростью. Никаких аварий или сужений дороги нет. Тем не менее при определённых условиях возникают волнообразные «пробки», распространяющиеся в направлении, противоположном движению. Этот феномен — классический пример эмерджентного поведения в транспортных системах: пробки возникают «сами собой» из коллективной динамики потока, а не из внешних причин.



Рисунок 1.6 — Симуляция транспортного потока на кольцевой дороге: эмерджентные «пробки» возникают без аварий и сужений дороги
(Источник: [11], онлайн доступ: <https://www.traffic-simulation.de/>)

1.7. Инструменты и технологии агентного моделирования

Реализация агентных моделей требует специализированных программных инструментов. Их можно разделить на три категории по уровню гибкости и масштабируемости.

Библиотеки программирования предоставляют максимальную гибкость: исследователь описывает агентов и среду на языке общего назначения. Среди них особо выделяются:

- **Mesa** (<https://mesa.readthedocs.io/>) — популярная открытая библиотека, предлагающая широкие возможности для создания агентных моделей с визуализацией. Хорошо интегрируется с экосистемой Python для научных вычислений (NumPy, pandas, matplotlib). Подходит для исследовательских задач.
- **Menge** (<http://gamma.cs.unc.com/Menge/>) — специализированное решение для моделирования толп и пешеходной динамики. Содержит готовые модели избегания столкновений — нетривиальной задачи, от которой во многом зависит реалистичность поведения толпы.

Среды визуального моделирования предоставляют высокоуровневый интерфейс с встроенной визуализацией:

- **NetLogo** (<https://www.netlogo.org/>) — наиболее широко используемая образовательная платформа агентного моделирования. Обладает интуитивным языком программирования и богатой библиотекой готовых моделей. Идеальна для начального знакомства с агентным подходом и быстрого прототипирования.
- **Repast** (<https://repast.github.io/>) — более мощный профессиональный инструмент, поддерживающий различные парадигмы моделирования и расширенные возможности анализа результатов. Широко применяется в академических исследованиях.

Промышленные решения ориентированы на прикладное применение в бизнесе и государственном управлении:

- **AnyLogic** (<https://www.anylogic.ru/>) — мощная коммерческая платформа, поддерживающая агентное, системно-динамическое и событийно-ориентированное моделирование. Предлагает богатые библиотеки для транспортных систем, логистики и городской среды.
- **Arena** (<https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html>) — специализированный

инструмент для имитационного моделирования бизнес-процессов, преимущественно дискретно-событийных систем. Более тяжёлый и дорогой по сравнению с AnyLogic.

Ключевые функциональные требования к инструментарию агентного моделирования включают: описание агентов (внешнее поведение и внутренняя когнитивная логика); моделирование среды (её структура, ресурсы, динамика); проведение имитационного эксперимента (развёртывание системы во времени); методы оптимизации (параметров агентов, правил среды, управляющих воздействий); методы идентификации и усвоения данных (калибровка модели); методы визуализации (пространственные, временные, сетевые). Важным вопросом является масштабируемость: при увеличении числа агентов вычислительная сложность растёт, и параллелизация одного сценария затруднена из-за зависимостей между агентами. Частичным решением служит иерархическое моделирование, когда группы агентов объединяются в «супер-агенты», или разбиение на независимые сценарии для параллельного запуска.

1.8. Ограничения и проблемы агентного моделирования

При всей привлекательности агентного подхода, он сопряжён с рядом существенных ограничений — как технических, так и концептуальных. Технические ограничения:

- **Вычислительная сложность** — по мере увеличения числа агентов и усложнения правил их поведения требования к ресурсам растут, что может делать моделирование практически неосуществимым для крупномасштабных систем. Распараллеливание на уровне одного сценария затруднено.
- **Ограниченность данных** — для калибровки и валидации часто необходимы подробные данные о поведении отдельных компонентов, которые могут быть недоступны. Внутренние механизмы принятия решений у людей и животных в принципе ненаблюдаемы, что вынуждает вводить эвристические предположения.
- **Идентификация модели** — построение корректной модели требует глубокого понимания системы. Существует риск создания модели, которая «воспроизводит» наблюдаемое поведение за счёт подгонки параметров, а не благодаря верному пониманию механизмов — аналог «переобучения» в машинном обучении.
- **Чувствительность к параметрам** — небольшие изменения параметров могут приводить к качественным изменениям

поведения системы. Это связано с нелинейностью агентных систем и их склонностью к эмерджентному поведению.

- **Недетерминированность** — стохастичность пронизывает все уровни агентной модели. Это требует проведения множественных прогонов и статистического анализа результатов.
- **Валидация** — проверка адекватности агентной модели особенно сложна: традиционные методы валидации часто неприменимы, а наблюдение поведения отдельных агентов в реальной системе может быть невозможным.

Концептуальные ограничения:

- **Редукционизм** — любая модель упрощает реальность. В агентных моделях важно понимать: именно опущенные аспекты поведения могут оказаться критически важными для воспроизведения ключевых свойств системы — особенно с учётом нелинейности и эмерджентности.
- **Неожиданное поведение** — способность порождать непредвиденные паттерны — одновременно сильная сторона и источник проблем. Неучтённые особенности взаимодействия могут приводить к нежелательным результатам, которые трудно предвидеть на этапе проектирования.
- **Переносимость моделей** — модель, разработанная для конкретных условий, может оказаться неприменимой в другой ситуации. В социальных и экономических системах культурные и институциональные различия существенно влияют на поведение агентов.
- **Этические аспекты** — особенно актуальны при моделировании человеческого поведения или использовании моделей для принятия решений. Вопросы приватности данных, потенциального манипулирования и ответственности за последствия требуют внимательного осмысления.

Осознание этих ограничений не умаляет ценности агентного подхода, но позволяет более взвешенно применять его на практике. Критическая оценка допущений и тщательная интерпретация результатов — необходимые условия эффективного использования агентного моделирования.

1.9. Заключение

Первая лекция познакомила нас с основными понятиями агентного моделирования. Центральными объектами этой парадигмы являются агент — автономная сущность, воспринимающая среду через

сенсоры и воздействующая на неё через эффекторы — и среда — окружение, в котором агенты существуют и взаимодействуют.

Агентное моделирование реализует методологию «снизу вверх»: простые локальные правила поведения агентов порождают сложные глобальные паттерны — эмерджентное поведение. Классические примеры (модель Шеллинга, Voids, клеточные автоматы Вольфрама, транспортные потоки) наглядно демонстрируют, как из единственного правила возникают структуры, удивительно похожие на природные явления.

По сравнению с традиционным математическим моделированием, агентный подход выигрывает в описании систем с гетерогенными компонентами, нелинейными взаимодействиями и адаптивным поведением. Платой за эту выразительность являются высокие вычислительные требования, сложность калибровки и валидации, а также ряд концептуальных ограничений, о которых необходимо помнить.

Классификация интеллектуальных агентов — от простых рефлексивных до обучающихся — отражает спектр возможных архитектур внутренней логики и указывает на связь агентного моделирования с теорией искусственного интеллекта. Эта связь будет подробнее раскрыта в последующих лекциях курса.

Последующие лекции посвящены углублению отдельных аспектов: истории и приложениям агентного моделирования (2), архитектуре и характеристикам агентов (3), механизмам перцепции, действий и принятия решений (4), принципам взаимодействия в многоагентных системах (5) и другим темам вплоть до открытых проблем и перспектив (12).

Рекомендуемая литература и источники

1. Russell S.J. Artificial Intelligence: A Modern Approach / S.J. Russell, P. Norvig. — 4th ed. — Hoboken : Pearson, 2020. — 1132 p. — ISBN 978-0-13-461099-3. — URL: <https://aima.cs.berkeley.edu/>
2. Macal C.M. Tutorial on Agent-Based Modelling and Simulation / C.M. Macal, M.J. North // Journal of Simulation. — 2010. — Vol. 4, No. 3. — P. 151–162. — DOI: 10.1057/jos.2010.3. — URL: <https://doi.org/10.1057/jos.2010.3>
3. Bonabeau E. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems / E. Bonabeau // Proceedings of the National Academy of Sciences. — 2002. — Vol. 99, Suppl. 3. — P. 7280–7287. — DOI: 10.1073/pnas.082080899. — URL: <https://www.pnas.org/doi/10.1073/pnas.082080899>
4. Schelling T.C. Dynamic Models of Segregation / T.C. Schelling // Journal of Mathematical Sociology. — 1971. — Vol. 1, No. 2. — P. 143–186. — DOI:

- 10.1080/0022250X.1971.9989794. — URL:
<https://www.tandfonline.com/doi/abs/10.1080/0022250X.1971.9989794>
5. Reynolds C.W. Flocks, Herds, and Schools: A Distributed Behavioral Model / C.W. Reynolds // Proceedings of ACM SIGGRAPH'87 (Computer Graphics). — New York : ACM, 1987. — Vol. 21, No. 4. — P. 25–34. — DOI: 10.1145/37402.37406. — URL:
<https://dl.acm.org/doi/10.1145/37402.37406>
6. Wolfram S. A New Kind of Science / S. Wolfram. — Champaign : Wolfram Media, 2002. — 1197 p. — ISBN 978-1-57955-008-0. — URL: <https://www.wolframscience.com/nks/>
7. Wilensky U. An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo / U. Wilensky, W. Rand. — Cambridge : MIT Press, 2015. — 504 p. — ISBN 978-0-262-73189-8.
8. Gilbert N. Simulation for the Social Scientist / N. Gilbert, K. Troitzsch. — 2nd ed. — Maidenhead : Open University Press, 2005. — 295 p. — ISBN 978-0-335-21600-0.
9. Macal C.M. Tutorial on Agent-Based Modelling and Simulation / C.M. Macal, M.J. North // Journal of Simulation. — 2010. — Vol. 4, No. 3. — P. 151–162. — DOI: 10.1057/jos.2010.3. — URL: <https://doi.org/10.1057/jos.2010.3>
10. Hunter E. Using a Socioeconomic Segregation Burn-in Model to Initialise an Agent-Based Model for Infectious Diseases / E. Hunter, B. Mac Namee, J.D. Kelleher // Journal of Artificial Societies and Social Simulation. — 2018. — Vol. 21, No. 4. — Art. 9. — DOI: 10.18564/jasss.3870. — URL: <http://jasss.soc.surrey.ac.uk/21/4/9.html>
11. Treiber M. Congested Traffic States in Empirical Observations and Microscopic Simulations / M. Treiber, A. Hennecke, D. Helbing // Physical Review E. — 2000. — Vol. 62, No. 2. — P. 1805–1824. — DOI: 10.1103/PhysRevE.62.1805. — URL: <https://doi.org/10.1103/PhysRevE.62.1805>

Лекция 2. История и приложения агентного моделирования

Агентное моделирование — одна из тех редких методологий, чья история позволяет чётко проследить путь от абстрактной математической идеи до зрелого инженерного инструмента, используемого в задачах государственного управления, медицины, урбанистики и проектирования интеллектуальных систем. Понимание этого пути необходимо не только для общей эрудиции; оно объясняет, почему сегодняшнее агентное моделирование выглядит именно так, какие задачи оно умеет решать лучше других методов и в каком направлении движется.

2.1. Истоки: 1940–1960-е годы

История агентного моделирования, как и многих современных вычислительных подходов, уходит корнями в 1940–1960-е годы — время, когда закладывались основы почти всех дисциплин в области информатики и смежных областей. Ключевую роль сыграли два имени: Джон фон Нейман и Станислав Улам.

Джон фон Нейман (John von Neumann) и **Станислав Улам** (Stanislaw Ulam) в конце 1940-х годов разработали концепцию *клеточных автоматов* (cellular automata) — дискретных вычислительных систем, в которых решётка ячеек меняет своё состояние согласно локальным правилам взаимодействия с соседями. Изначально фон Нейман интересовался задачей *самовоспроизводящихся автоматов* — абстрактных машин, способных создавать собственные копии. Улам предложил реализовать эту идею именно в клеточном формате, что и стало отправной точкой.

Значение этих работ для агентного моделирования трудно переоценить. Именно тогда была сформулирована ключевая идея: глобальное сложное поведение системы может возникать из простых локальных правил взаимодействия её элементарных компонентов. Каждая ячейка в клеточном автомате — прообраз агента: она имеет состояние, воспринимает состояния соседей и изменяется согласно заложенной логике.

Параллельно разворачивался другой важный интеллектуальный процесс. Алан Тьюринг в своих работах о вычислимости и статье «Вычислительные машины и разум» (1950) поставил вопрос о возможности имитации интеллекта машинами. Хотя сам Тьюринг не занимался агентными моделями, его рассуждения о границах между

имитацией и реальным мышлением питали дискуссии, которые впоследствии оформились в теорию интеллектуальных агентов.

2.2. Первые агентные модели: 1970-е годы

Семидесятые годы стали временем, когда идеи из теоретической информатики и математики начали встречаться с конкретными социальными и научными задачами. Одним из первых, кто реализовал это на практике в контексте, близком к современному агентному моделированию, был экономист Томас Шеллинг.

Томас Шеллинг (Thomas C. Schelling) в 1971 году опубликовал в *Journal of Mathematical Sociology* статью «Dynamic Models of Segregation», ставшую хрестоматийной для всей области. В ней он использовал клеточный автомат для изучения расселения людей в городе. Правило поведения агента было предельно простым: если менее 30% соседей принадлежат к той же группе, агент перемещается на случайную свободную клетку.

Journal of Mathematical Sociology
1971, Vol. 1, pp 143–186

© Gordon and Breach Science Publishers
Printed in Birkenhead, England

DYNAMIC MODELS OF SEGREGATION†

THOMAS C. SCHELLING
Harvard University

Some segregation results from the practices of organizations, some from specialized communication systems, some from correlation with a variable that is non-random; and some results from the interplay of individual choices. This is an abstract study of the interactive dynamics of discriminatory individual choices. One model is a simulation in which individual members of two recognizable groups distribute themselves in neighborhoods defined by reference to their own locations. A second model is analytic and deals with compartmented space. A final section applies the analytics to 'neighborhood tipping.' The systemic effects are found to be overwhelming: there is no simple correspondence of individual incentive to collective results. Exaggerated separation and patterning result from the dynamics of movement. Inferences about individual motives can usually not be drawn from aggregate patterns. Some unexpected phenomena, like density and vacancy, are generated. A general theory of 'tipping' begins to emerge.

Рисунок 2.1 — Титульная страница статьи Т. Шеллинга «Dynamic Models of Segregation» (1971), ставшей одной из первых агентных моделей социальных систем (Источник: [1])

Результат оказался неожиданно сильным: даже при таком мягком предпочтении (всего 30%!) на уровне всей системы возникала выраженная сегрегация — однородные кластеры, которые не были заложены ни в каком индивидуальном правиле. Это был первый убедительный демонстрационный пример того, что макросоциальные явления могут не соответствовать — и даже противоречить — микропредпочтениям отдельных людей. Важнейший вывод модели

Шеллинга состоит в том, что нельзя делать прямые выводы об индивидуальных мотивах, наблюдая лишь агрегатные паттерны: человек с умеренными предпочтениями может жить в полностью сегрегированном районе не потому, что сам этого хочет, а в силу системной динамики.

Практические применения модели оказались весьма широкими. Исследования в крупных американских городах (Чикаго, Нью-Йорке, Лос-Анджелесе, Сан-Франциско) обнаружили, что измеренные коэффициенты сегрегации вполне соответствуют предсказаниям модели Шеллинга.

2.3. Институт Санта-Фе и сложные адаптивные системы: 1980-е годы

Восьмидесятые годы стали периодом концептуального оформления агентного подхода. Центральным местом этого процесса стал Институт Санта-Фе (Santa Fe Institute, SFI) — частный исследовательский центр в Нью-Мексико, основанный в 1984 году и посвящённый изучению сложных адаптивных систем (complex adaptive systems, CAS). Именно здесь сформировалось то, что можно назвать научной парадигмой агентного моделирования.

Джон Холланд (John H. Holland) в своей книге «Adaptation in Natural and Artificial Systems» (1975, переиздание 1984) заложил фундамент понимания того, как системы агентов могут *обучаться и адаптироваться* через механизмы, аналогичные биологической эволюции. Холланд разработал теорию генетических алгоритмов и классификаторных систем, где популяции простых правил конкурируют и эволюционируют в среде. Это был принципиальный шаг: агенты перестали быть статичными — они могли учиться.

Роберт Аксельрод (Robert Axelrod) в книге «The Evolution of Cooperation» (1984) подошёл к агентному моделированию с другой стороны. Он организовал компьютерные турниры, в которых агенты, реализующие разные стратегии поведения в игре «Дилемма заключённого», соревновались друг с другом. Главный вопрос: как в популяции эгоистичных агентов может спонтанно возникнуть и поддерживаться *кооперация*? Победила стратегия «Tit-for-Tat» («зуб за зуб»): начинать с кооперации, а затем копировать последнее действие партнёра. Этот результат произвёл сильное впечатление на экономистов, политологов и социологов, поскольку давал объяснение механизмам возникновения сотрудничества без центрального координатора.

Работы Института Санта-Фе сформировали несколько ключевых понятий, ставших фундаментом агентного моделирования: сложные адаптивные системы как системы, состоящие из множества взаимодействующих агентов, способных адаптироваться; нелинейность как принципиальное свойство, делающее редуктивный анализ недостаточным; эмерджентность как возникновение системных свойств, не выводимых напрямую из свойств компонентов; а также самоорганизация — спонтанное упорядочение без внешнего управления.

2.4. Эпоха вычислительных социальных наук: 1990-е годы

Девяностые годы стали переломными для агентного моделирования: из теоретической абстракции оно превратилось в практический инструмент исследования. Катализатором послужил стремительный рост доступных вычислительных мощностей. Кластерные и высокопроизводительные вычислительные ресурсы стали доступны исследователям из самых разных областей, и агентные модели перестали быть игрушкой теоретиков.

Поворотным событием стала публикация книги **Джошуа Эпштейна** (Joshua M. Epstein) и **Роберта Акстелла** (Robert L. Axtell) «Growing Artificial Societies: Social Science from the Bottom Up» (1996). В ней авторы демонстрировали, как из взаимодействия простых агентов «вырастают» целые общества — со своей экономикой, культурой, социальной стратификацией. Центральной была знаменитая модель *Sugarscape*: агенты перемещались по решётке в поисках «сахара» (ресурса), торговали, размножались, умирали. Из этих простых правил возникали неравенство в распределении богатства, торговые сети и миграционные паттерны — явления, узнаваемо похожие на реальные социальные закономерности.

Именно в 1990-е годы агентное моделирование начало активно применяться в социальных науках, экологии, экономике и биологии. Так возникло направление, получившее название вычислительные социальные науки (computational social science, CSS) — интеграция вычислительных методов с задачами понимания общественных процессов. Параллельно шло развитие инструментальной базы: появились первые специализированные платформы для агентного моделирования, ставшие стандартами отрасли.

2.5. Зрелость и гибридизация: 2000-е годы — настоящее время

С начала двухтысячных годов и до наших дней развитие агентного моделирования характеризуется двумя взаимосвязанными тенденциями: экспансией в новые предметные области и активной гибридизацией с другими технологиями.

Экспансия охватила поистине широкий спектр дисциплин. Помимо классических социальных наук, агентные модели стали применяться в медицине (моделирование эпидемий, внутрибольничных инфекций, клеточной динамики), биологии (популяционная динамика, поведение стай), физике (квантовые системы, молекулярная динамика), экологии, экономике, политологии, урбанистике, логистике, кибербезопасности и даже в компьютерных играх.

Гибридизация проявляется в нескольких формах. Во-первых, это совмещение агентных моделей с методами искусственного интеллекта и машинного обучения для описания внутренней логики агентов: нейронные сети, обучение с подкреплением, генеративные модели. Во-вторых, интеграция с методами анализа больших данных для калибровки и валидации моделей на реальных данных. В-третьих, гибридизация с численными математическими моделями, когда, например, для описания физических процессов используются дифференциальные уравнения, а для поведения агентов — агентный подход. Наконец, в последние годы появился качественно новый класс агентов — агенты на основе больших языковых моделей (Large Language Models, LLM), открывающие принципиально новые возможности для моделирования сложного поведения.

Наряду с расширением возможностей, современный этап характеризуется усиленным вниманием к вопросам объяснимости, этики и сценариев применения. Агентные модели, будучи мощным инструментом предсказания и оптимизации, несут ответственность: их результаты влияют на принятие решений в области здравоохранения, городского планирования, управления рисками. Вопросы о том, насколько модель прозрачна, воспроизводима и нейтральна, стали неотъемлемой частью профессиональной дискуссии.

Таблица 2.1 — Ключевые этапы истории агентного моделирования

Период	Ключевые события и персоналии	Значение для АМ
1940–1960-е	Дж. фон Нейман и С. Улам — теория клеточных автоматов и самовоспроизводящихся автоматов; А. Тьюринг — работы по вычислимости и имитации интеллекта	Концептуальные основы: идея возникновения сложного поведения из простых локальных правил
1970-е	Т. Шеллинг — «Динамические модели сегрегации» (1971); первые осмысленные агентные модели социальных систем	Первые агентные модели с практическим применением; демонстрация эмерджентности
1980-е	Институт Санта-Фе; Дж. Холланд — «Адаптация в естественных и искусственных системах» (1975/1984); Р. Аксельрод — «Эволюция кооперации» (1984)	Формирование теории сложных адаптивных систем (CAS); модели кооперации и конкуренции
1990-е	Рост вычислительных мощностей; Дж. Эпштейн и Р. Акстелл — «Growing Artificial Societies» (1996); появление NetLogo и Repast	Взрыв практических применений; возникновение вычислительных социальных наук (CSS)
2000-е–н. в.	Экспансия в медицину, биологию, физику, урбанистику; гибридизация с ИИ, МО, большими данными; LLM-агенты (MetaGPT, ChatDev)	Зрелая методология; интеграция с другими парадигмами; новые горизонты применения

2.6. Области применения агентного моделирования

Обратимся теперь к конкретным приложениям — тем областям, где агентное моделирование продемонстрировало свою практическую ценность. Рассмотрим наиболее показательные примеры, иллюстрирующие разнообразие задач и подходов.

Эпидемиология и моделирование пандемий. Моделирование эпидемий — один из наиболее «естественных» классов задач для агентного подхода. Традиционные математические модели эпидемического распространения (семейство SIR — Susceptible-Infected-Recovered и его расширения) описывают динамику через системы дифференциальных уравнений для усреднённых популяций. Агентные модели позволяют принципиально выйти за это ограничение.

Агентная SEIR-модель (*Susceptible — Exposed — Infectious — Removed*) функционирует следующим образом (рисунки 2.2, 2.3): каждый человек представлен агентом с индивидуальными характеристиками (возраст, состояние здоровья, социальные связи, поведенческие паттерны), перемещающимся в физической среде. Заражение происходит при контакте агентов на определённой дистанции — это позволяет естественным образом моделировать влияние социального дистанцирования, ношения масок, вентиляции помещений. Каждый агент проходит через цепочку состояний: восприимчивый → экспонированный (инкубационный период) → заразный → переболевший/иммунный. При этом индивидуальные вариации в длительности каждой стадии задаются стохастически.

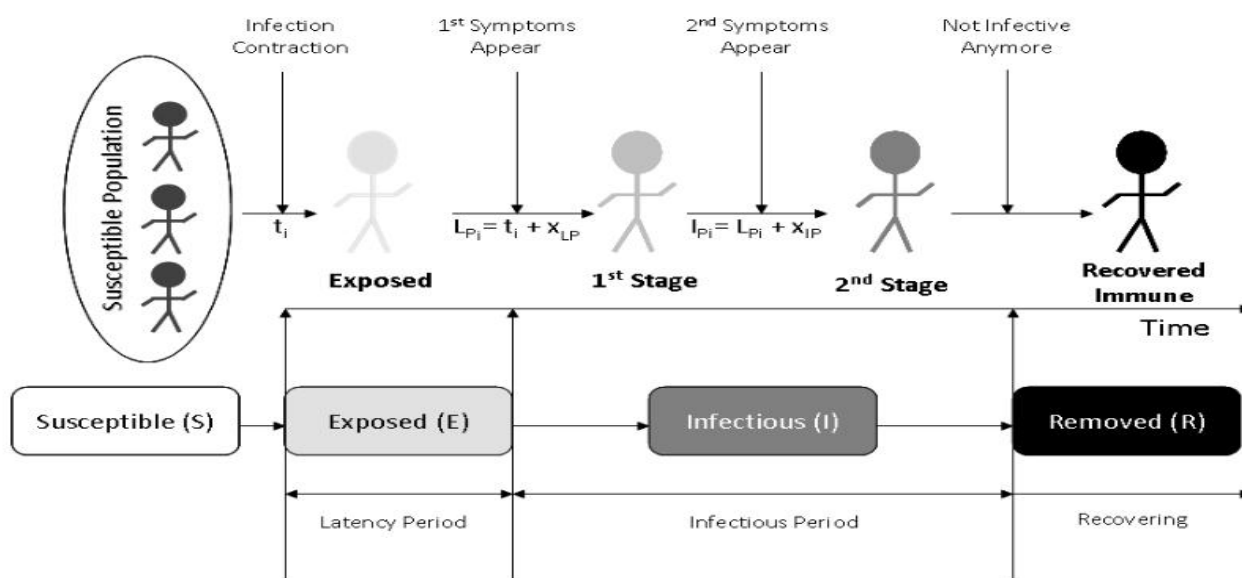


Рисунок 2.2 — Схема переходов между эпидемиологическими состояниями в модели SEIR: Susceptible (восприимчивые) → Exposed (экспонированные) → Infectious (заразные) → Removed (переболевшие/иммунные) (Источник: [6])

Особое значение агентные эпидемиологические модели приобрели в период пандемии COVID-19. Глобальные эпидемиологические модели имеют слишком большую неопределённость: слишком много неизвестных параметров, слишком разнородные популяции. Однако модели на уровне района, квартала или конкретного учреждения (метро, больница, торговый центр) позволяют получать практически значимые результаты. Такие модели были использованы для сравнительного анализа сценариев: что происходит при введении масочного режима, при частичном карантине, при изменении плотности заполнения транспортных узлов.

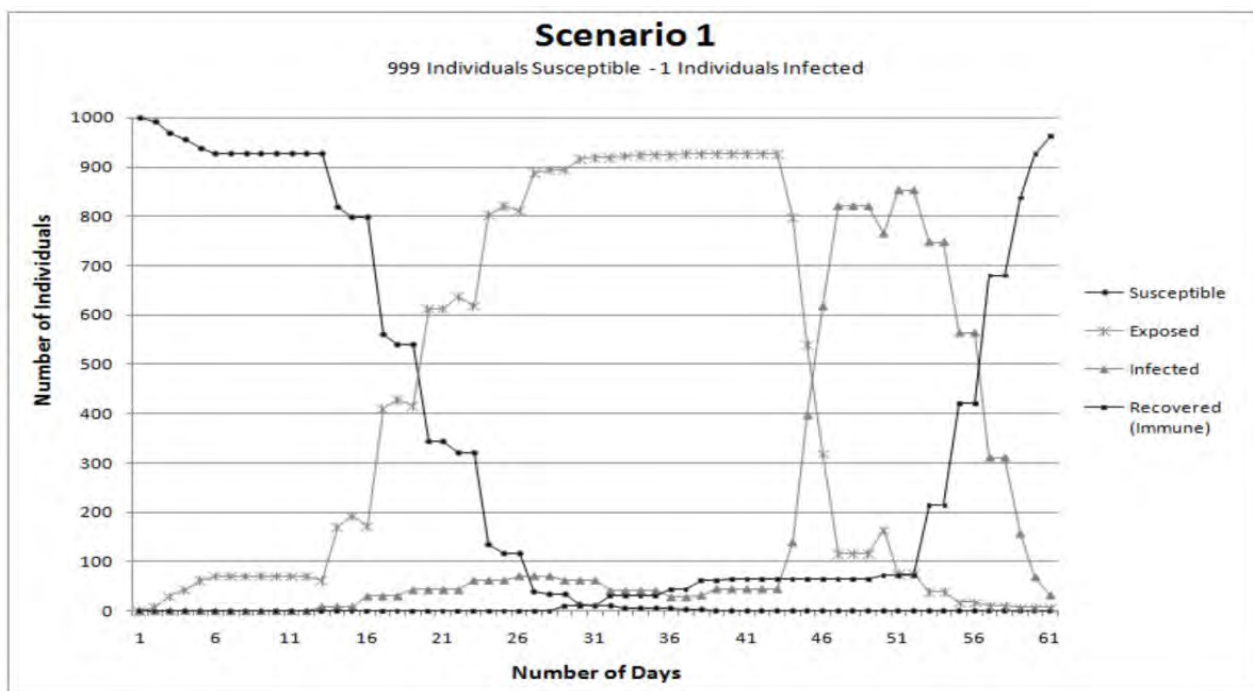


Рисунок 2.3 — Результаты имитационного моделирования SEIR: динамика числа агентов в каждом состоянии для сценария «999 восприимчивых и 1 заразный». Наблюдается характерный эпидемический пик заразных и запаздывающий рост переболевших (Источник: [6])

С точки зрения методологии, агентные эпидемиологические модели хорошо иллюстрируют главное преимущество подхода: способность воспроизводить пространственно неоднородную динамику и гетерогенность агентов. Два человека с одинаковыми биологическими параметрами могут иметь принципиально разные траектории заражения в зависимости от своих маршрутов, социальных контактов и поведенческих решений.

Сельское хозяйство. Другой яркий пример применения агентных моделей — задача управления распространением инвазивных видов в сельском хозяйстве. В исследовании [12] изучалось, как фермеры непреднамеренно переносят семена сорняков при сельскохозяйственных работах на своих фермах.

Модель включала ГИС-среду (геоинформационную систему), воспроизводящую реальный ландшафт с полями, дорогами и фермерскими хозяйствами. Агентами были фермеры, перемещающиеся по этой среде, с реалистичными маршрутами и поведенческими паттернами. Семена переносились в зависимости от используемой техники и пространственных контактов. Модель позволила выявить «критические узлы» распространения — конкретные маршруты и точки взаимодействия, через которые происходит наибольший перенос семян.

Особенно примечательно практическое применение этой модели: она была использована непосредственно в процессе обучения фермеров. Участники обучающей программы «ходили» по модели — наблюдали за последствиями своих решений в симулированной среде и вырабатывали более экологически осознанные стратегии управления. Это редкий пример, когда агентная модель используется не только для анализа, но и как интерактивный образовательный инструмент. По данным исследователей, после прохождения обучения с использованием модели распространение сорняков в реальных хозяйствах участников заметно снизилось — эффект, который «классическое» обучение не давало.

Динамика толпы и моделирование пешеходов. Моделирование поведения толпы — одна из наиболее развитых и практически востребованных областей применения агентного подхода. Задача отнюдь не тривиальна: даже такое простое действие, как обход угла в коридоре, реальный человек выполняет не по прямой (теоретически оптимальной) траектории, а следуя сложной логике избегания столкновений, социальных норм и восприятия пространства.

Среди классических моделей избегания столкновений особое место занимают две. Модель социальных сил (Social Force Model), предложенная Хельбингом и Молнаром в 1995 году, описывает пешеходов как частицы, подверженные «социальным силам» — привлечению к цели и отталкиванию от других людей и препятствий, аналогично физическому взаимодействию заряженных частиц. Скорость агента в каждый момент определяется равнодействующей этих сил. Модель RVO (Reciprocal Velocity Obstacles), выросшая из задач навигации кораблей в узких проливах, строит для каждого агента «конусы запрещённых скоростей» — направлений движения, при которых неизбежно столкновение с другим агентом — и выбирает оптимальную скорость вне этих конусов.

Исследование [7] показало, что ни одна из классических моделей не является универсальной: в разных пространственных зонах и при разной плотности толпы оптимальной оказывается разная модель. Гибридный ансамблевый подход (рисунок 2.4), в котором для разных частей среды и разных временных фаз активируются разные субмодели, даёт наилучшие результаты.

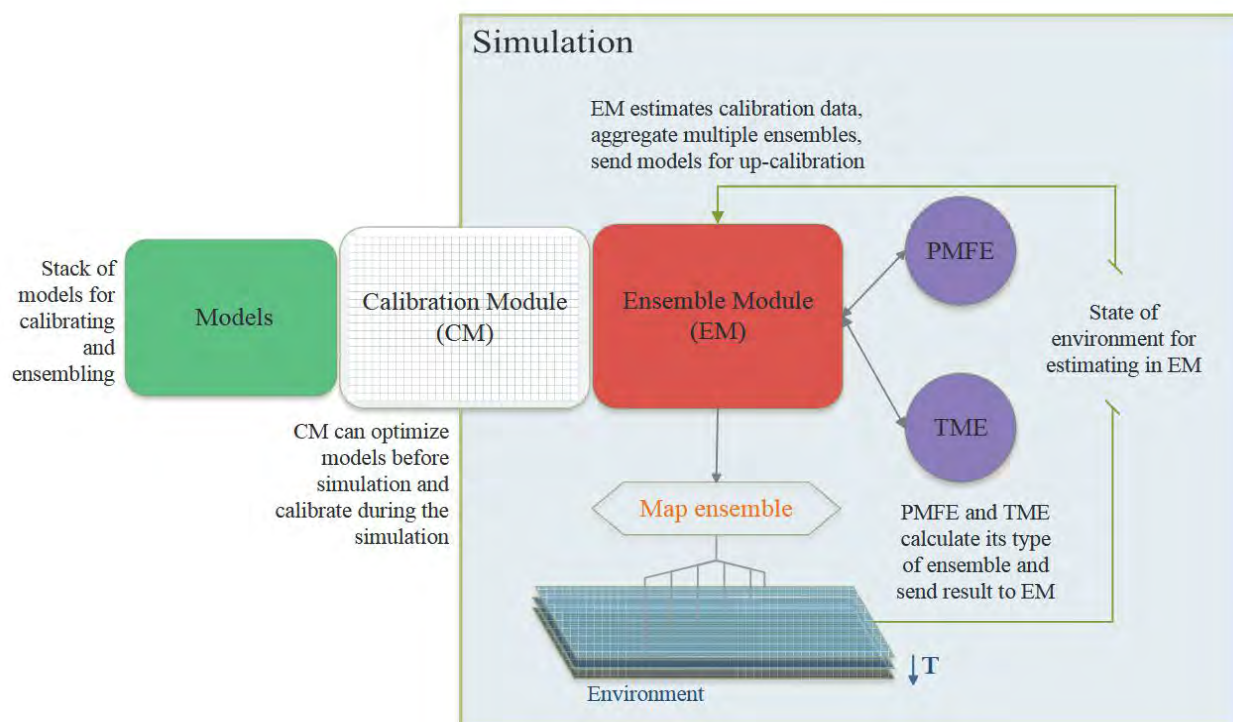


Рисунок 2.4 — Архитектура гибридной системы моделирования толпы: модуль калибровки (CM), ансамблевый модуль (EM) и два субмодуля предсказания (PMFE, TME) образуют контур адаптивного управления моделями в среде (Источник: [7])

Биомедицина: клеточные агенты. Одним из наиболее впечатляющих расширений агентного подхода является его применение на клеточном уровне в биомедицинских исследованиях. Здесь агентами выступают клетки биологических тканей, а средой — внеклеточный матрикс с его биохимическими и механическими свойствами (рисунок 2.5).

Показателен пример модели кровеносной системы человека, объединяющей множество масштабов моделирования [12]. Задача состояла в моделировании процесса рестеноза — повторного сужения коронарной артерии после установки стента. Модель включала три масштабных уровня: одномерная модель всей кровеносной системы (каждый сосуд — линейная динамика давления и скорости); трёхмерная гидродинамическая модель потока крови через стентированный участок артерии; агентная модель роста клеток — прорастания ткани через сетку стента, зависящего от биохимических и механических условий.

Такая иерархическая многомасштабная модель позволяла исследовать эффективность различных способов снижения рестеноза — например, нанесения на стент ингибитора роста клеток — без риска для

пациента. Это наглядный пример того, как агентное моделирование на клеточном уровне органично встраивается в общую задачу персонализированного прогнозирования исходов лечения.

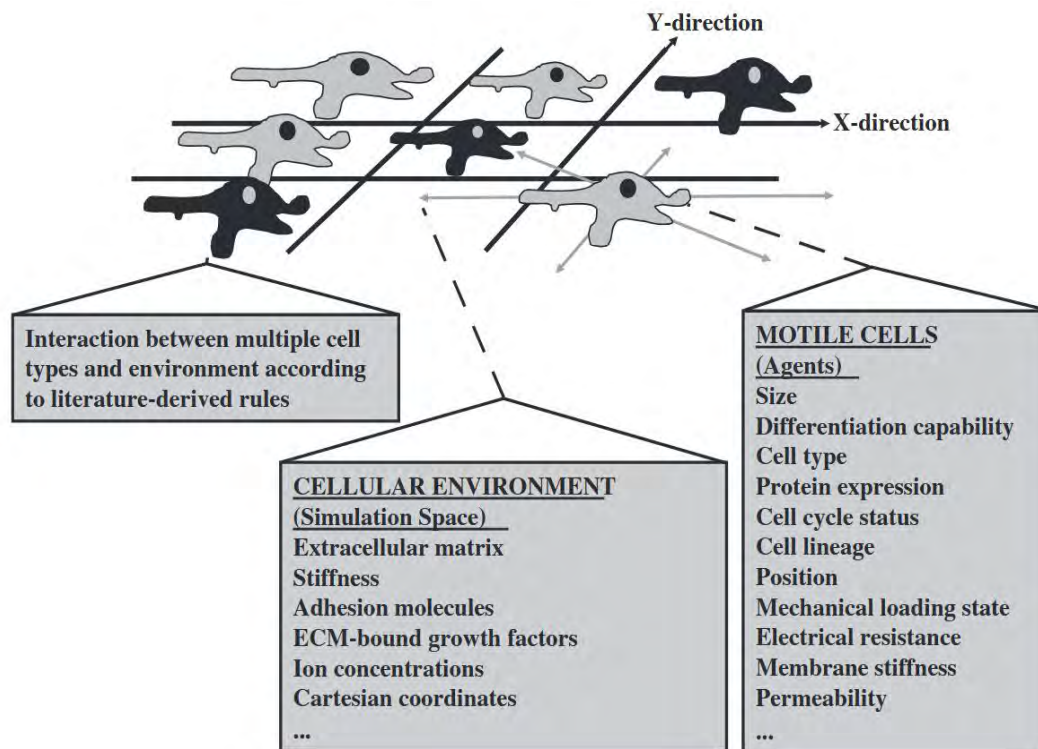


Рисунок 2.5 — Схема агентной модели клеточной динамики: подвижные клетки (агенты) взаимодействуют с клеточной средой (simulation space) согласно правилам, выведенным из литературы. Агенты характеризуются типом клетки, фазой клеточного цикла, белковой экспрессией и механическими свойствами (Источник: [11])

Программные агенты и большие языковые модели. Наиболее актуальным и быстро развивающимся направлением является применение агентного подхода в системах на основе больших языковых моделей (Large Language Models, LLM). Это принципиально новый класс агентов: их «внутренняя логика» реализована не набором детерминированных правил, а мощным генеративным языковым механизмом.

Один из показательных примеров — система MetaGPT и конкурирующая с ней ChatDev, в которых реализована команда программных агентов для автоматизации разработки программного обеспечения (рисунок 2.6). Каждый агент воплощает определённую роль: клиент, бизнес-аналитик, Team Lead, разработчики, тестировщики, технический писатель. За каждым агентом стоит языковая модель (ChatGPT или аналог) с системным промптом, описывающим его роль и функции.

агентным методом. Применение же обученных моделей в роли агентов в многоагентных системах замыкает этот круг: агентное моделирование становится основой и инструментом для развития ИИ-систем.

Охватив конкретные примеры, подведём системный итог. Таблица 2.2 суммирует области применения агентного моделирования, рассмотренные в данной лекции.

Глядя на этот список, нетрудно заметить общий паттерн: агентное моделирование особенно эффективно там, где система состоит из гетерогенных элементов с нетривиальным поведением, где взаимодействие носит локальный пространственный или топологический характер, а глобальные закономерности не могут быть непосредственно выведены из индивидуального поведения компонентов. Физическая среда — города, здания, биологические ткани — предоставляет контекст, в котором агентный подход раскрывает свой потенциал наиболее полно.

Таблица 2.2 — Обзор областей применения агентного моделирования

Область	Пример применения	Роль агентов
Эпидемиология	Моделирование COVID-19, гриппа	Люди с состояниями (S, E, I, R)
Сельское хозяйство	Распространение сорняков, обучение фермеров	Фермеры, семена, ГИС-среда
Динамика толпы	Метро, эвакуация, массовые мероприятия	Пешеходы с моделями избегания
Медицина (помещения)	Оптимизация госпиталя, маршруты персонала	Врачи, медсёстры, пациенты
Процессы лечения	Клинические пути, поток пациентов	Пациенты, персонал, ресурсы
Биомедицина	Рост ткани, рестеноз, клеточная динамика	Клетки в биологической среде
Цепочки поставок	Логистика авиазапчастей, производство	Склады, самолёты, подрядчики
Поведение клиентов	Маркетинг, анализ покупательского поведения	Покупатели с психологическими профилями
Городская среда	Скорая помощь, транспортные потоки	Машины, стационары, диспетчеры
ИИ / LLM	MetaGPT, ChatDev, агент в Minecraft	Специализированные LLM-агенты

2.7. Тенденции и перспективы

Анализируя историю и текущее состояние агентного моделирования, можно выделить несколько устойчивых тенденций, определяющих его развитие.

Первая тенденция — расширение областей применения. От изначальных задач в социальных науках и экологии агентное моделирование проникло в медицину, физику, биологию, экономику, информатику. Каждая новая область приносит собственные требования, методы валидации и источники данных — обогащая общую методологию.

Вторая тенденция — гибридизация. Агентное моделирование всё активнее интегрируется с машинным обучением (для описания поведения агентов), методами анализа данных (для калибровки и валидации), численными моделями (для описания физических процессов) и большими языковыми моделями (для реализации сложной когнитивной логики). Граница между «агентной моделью» и «ИИ-системой» становится всё менее определённой.

Третья тенденция — внимание к ответственному применению. Рост практических приложений агентных моделей в чувствительных областях (здравоохранение, управление городом, маркетинг, уголовное правосудие) ставит острые вопросы об объяснимости, прозрачности, воспроизводимости и этической приемлемости этих инструментов. Это не специфическая проблема агентного моделирования — она стоит перед всеми методами, претендующими на влияние на реальные решения, — но в агентных системах она усиливается непредсказуемостью эмерджентного поведения.

Четвёртая тенденция — конвергенция с концепцией цифровых двойников (digital twins). Создание высокодетализированных агентных моделей реальных систем — производственных предприятий, больниц, городских кварталов — которые работают в режиме реального времени, получая данные от датчиков и обновляя своё состояние, открывает принципиально новые возможности для оперативного управления.

2.8. Заключение

Путь агентного моделирования — от клеточных автоматов фон Неймана и Улама до LLM-агентов в Minecraft — отражает более широкую эволюцию науки о сложных системах. На каждом этапе методология обогащалась новыми идеями, инструментами и областями

применения, сохраняя при этом фундаментальный принцип: понимать сложное через простые локальные правила взаимодействия элементов.

Примеры, рассмотренные в этой лекции, демонстрируют зрелость подхода. Агентное моделирование больше не является экзотическим инструментом теоретиков — оно используется для проектирования реальных больниц, управления скорой помощью, оптимизации эвакуационных маршрутов, изучения механизмов эпидемий и создания команд программных разработчиков. Понимание истории этого развития помогает правильно оценивать возможности и ограничения метода, выбирать подходящие инструменты и ставить реалистичные ожидания.

Следующие лекции посвящены более детальному рассмотрению конкретных аспектов: архитектуры агентов (3), механизмов перцепции и принятия решений (4), принципов взаимодействия в многоагентных системах (5). Каждая из этих тем опирается на понятийный фундамент, заложенный первыми двумя лекциями.

Рекомендуемая литература и источники

1. Schelling T.C. Dynamic Models of Segregation / T.C. Schelling // Journal of Mathematical Sociology. — 1971. — Vol. 1, No. 2. — P. 143–186. — DOI: 10.1080/0022250X.1971.9989794. — URL: <https://www.tandfonline.com/doi/abs/10.1080/0022250X.1971.9989794>
2. Holland J.H. Adaptation in Natural and Artificial Systems / J.H. Holland. — Ann Arbor : University of Michigan Press, 1975; Cambridge : MIT Press, 1992. — 228 p. — ISBN 978-0-262-58111-5.
3. Axelrod R. The Evolution of Cooperation / R. Axelrod. — New York : Basic Books, 1984. — 241 p. — ISBN 978-0-465-02121-5.
4. Epstein J.M. Growing Artificial Societies: Social Science from the Bottom Up / J.M. Epstein, R. Axtell. — Cambridge : MIT Press, 1996. — 228 p. — ISBN 978-0-262-55025-3.
5. Helbing D. Social Force Model for Pedestrian Dynamics / D. Helbing, P. Molnár // Physical Review E. — 1995. — Vol. 51, No. 5. — P. 4282–4286. — DOI: 10.1103/PhysRevE.51.4282. — URL: <https://link.aps.org/doi/10.1103/PhysRevE.51.4282>
6. Patlolla P. Agent-Based Simulation of SEIR Model for Infectious Disease Spread / P. Patlolla [et al.] // Proceedings of the International Conference on Health Informatics. — 2004. — PMID: PMC2729742. — URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2729742/>
7. Kovalchuk S.V. A Concept of a Platform for Simulation-Based Optimization of Evacuation Management / S.V. Kovalchuk [et al.] // Procedia Computer Science. — 2016. — DOI: 10.1016/j.procs.2016.05.333. — URL: <https://doi.org/10.1016/j.procs.2016.05.333>
8. Qian C. ChatDev: Communicative Agents for Software Development / C. Qian [et al.]. — arXiv:2307.07924 [cs.SE], 2023. — DOI: 10.48550/arXiv.2307.07924. — URL: <https://arxiv.org/abs/2307.07924>

9. Wang G. JARVIS: A Neuro-Symbolic Commonsense Reasoning Framework for Conversational Embodied Agents / G. Wang [et al.]. — arXiv:2311.05997 [cs.AI], 2023. — DOI: 10.48550/arXiv.2311.05997. — URL: <https://arxiv.org/abs/2311.05997>
10. Rebaudo F. Agent-Based Modeling of Human-Induced Spread of Invasive Species in Agricultural Landscapes: Insights from the Potato Moth in Ecuador / F. Rebaudo [et al.] // Journal of Artificial Societies and Social Simulation. — 2011. — Vol. 14, No. 3. — Art. 7. — DOI: 10.18564/jasss.1802. — URL: <http://jasss.soc.surrey.ac.uk/14/3/7.html>
11. Bryan C. Combining Experiments with Multi-Cell Agent-Based Modeling to Study Biological Tissue Patterning / C. Bryan [et al.] // Briefings in Bioinformatics. — 2007. — Vol. 8, No. 4. — P. 245–257. — DOI: 10.1093/bib/bbm024. — URL: <https://doi.org/10.1093/bib/bbm024>
12. Hoekstra A.G. Towards the virtual artery: a multiscale model for vascular physiology at the physics–chemistry–biology interface / A.G. Hoekstra [et al.] // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. — 2016. — Vol. 374, No. 2080. — P. 20160146. — DOI: 10.1098/rsta.2016.0146. — URL: <https://doi.org/10.1098/rsta.2016.0146>

Лекция 3. Архитектура и характеристики агентов

3.1. Введение: агент в среде и понятие задачи

Как было установлено в предыдущих лекциях, агент представляет собой автономную сущность, действующую в рамках некоторой среды, принимающую решения и совершающую действия для достижения собственных целей. В настоящей главе мы рассмотрим вопрос о том, как устроен агент изнутри, какие принципы лежат в основе его архитектуры и каким образом различные архитектурные подходы влияют на поведение агента в среде.

Для того чтобы ответить на эти вопросы, необходимо прежде всего уточнить ключевые элементы взаимодействия агента и среды. Согласно классической схеме (рисунок 1.1), агент воспринимает окружающую среду посредством **сенсоров** и совершает действия в ней с помощью **эффекторов** (актуаторов). Сигналы, поступающие от сенсоров, называются **перцептами** (от лат. «perceptum» — воспринятое). Именно они составляют входные данные для внутренней логики агента, которая определяет, какое действие следует предпринять в той или иной ситуации.

Согласно определению С. Рассела и П. Норвига [1], **поведение агента** — это некоторая функция, сопоставляющая последовательность воспринятых перцептов с выбираемыми действиями из множества доступных. Иными словами: «the job of AI is to design the agent program that implements the agent function mapping percepts to actions». Проблема состоит в том, что подобная функция в общем случае крайне сложна для конструирования и реализации, особенно если агент должен решать нетривиальные задачи в динамичной и частично наблюдаемой среде.

Ключевую роль в описании поведения агента играет понятие **задачи**. Задача — это формализованная цель или набор целей, которые агент должен достичь или поддерживать в ходе своей работы. Задачи могут быть заданы внешним образом (например, все агенты в модели эвакуации должны добраться до выхода) или генерироваться самим агентом в ходе его функционирования (например, робот формирует задачу «зарядиться», когда уровень заряда падает ниже допустимого порога). Наконец, задачи могут формулироваться на мета-уровне, когда агент обучается и его деятельность направлена на приобретение знаний или навыков.

Таблица 3.1 — Ключевые характеристики задач агента

Характеристика	Возможные значения	Влияние на архитектуру
Тип	Достижение / Поддержка состояния	Сессионность / непрерывность работы агента
Количество	Одна / Несколько (последовательно / параллельно)	Необходимость очерёдности или приоритизации
Необходимость	Обязательная / Опциональная	Логика вознаграждений
Ценность	Одинаковая / Разнородная	Механизм выбора задачи из пула
Ограничения	Время / Ресурсы / Пространство	Планирование с учётом бюджетов

Свойства задач. Рассмотрим основные характеристики, по которым принято описывать задачи агента.

Тип задачи определяет отношение задачи к целевому состоянию. Принципиально различают задачи **достижения состояния** (агент должен однократно достичь целевой конфигурации мира) и задачи **поддержки состояния** (агент должен постоянно удерживать некоторую характеристику в заданном диапазоне, например, поддерживать уровень заряда аккумулятора). Это различие напрямую влияет на логику модели: в первом случае агент может «выключиться» после достижения цели, во втором — он функционирует непрерывно.

Количество задач. Агент может иметь единственную задачу, по завершении которой его жизненный цикл заканчивается, либо несколько задач, возникающих последовательно (одна порождает другую) или параллельно (агент располагает пулом задач и решает их, руководствуясь определённым приоритетом).

Необходимость. Задачи делятся на **обязательные** (их невыполнение означает несостоятельность агента в рамках модели) и **опциональные** (их выполнение приносит агенту некоторое вознаграждение, но не является критически необходимым). Опциональные задачи часто моделируются через механизм вознаграждения, характерный для обучения с подкреплением.

Ценность задачи — это абстрактная количественная характеристика важности задачи для агента. В отличие от простого приоритета (блокирующего признака при параллельном исполнении),

ценность позволяет учитывать расход ресурсов, временные ограничения и другие факторы при выборе между несколькими одновременно активными задачами.

Ограничения могут быть временными (дедлайны), ресурсными (ограниченный бюджет вычислений, энергии, памяти) или пространственными. Следует отметить, что работа с задачами сопряжена с рядом трудностей: задачи не всегда поддаются однозначной классификации и приоритизации, механизм их оценки может быть неявным, а неполнота информации о среде непосредственно влияет на оценку сложности и ценности задачи.

3.2. Понятие архитектуры агента

Понятие **архитектуры агента** вводится для того, чтобы описать логику агента в некоторой обобщённой, пригодной для повторного использования форме. Классическое определение Рассела и Норвига рассматривает агента как вычислительное устройство с физическими сенсорами и актуаторами («some sort of computing device with physical sensors and actuators»), однако подобный подход ничего не говорит нам о том, как именно устроена внутренняя логика агента. Именно здесь и появляется архитектура.

Полезной аналогией служит **архитектура программного обеспечения**. Подобно тому, как программная архитектура задаёт паттерны построения системы, определяет используемые компоненты и инструменты, архитектура агента задаёт «виртуальную машину», в которой исполняется программа агента. Ещё более точной является аналогия с **операционной системой**: та предоставляет набор атомарных операций (работа с файловой системой, устройствами ввода-вывода и т. д.), тогда как архитектура агента обеспечивает унифицированный доступ к перцепции, действиям и задачам.

С точки зрения объектно-ориентированного программирования архитектуру агента можно рассматривать как **класс программ-агентов**, определяющий набор методов — как модифицируемых пользователем, так и фиксированных архитектурой. Этот класс согласован со средой и множеством задач, которые агент способен решать.

Архитектура агента фиксирует следующие аспекты. Во-первых, это «**программа**» (**логика**) **агента** — паттерны взаимодействия с окружением и набор доступных методов. Во-вторых, **атомарные операции и доступные компоненты**, которыми агент может пользоваться. В-третьих, **уровень автоматизации** — какие процессы осуществляются без участия «программы» агента, а какие требуют

явного управления. В-четвёртых, **инструменты памяти, обучения и рассуждения**. Наконец, архитектура определяет **ограничения** — технические, логические и иные. Цель разработки архитектуры состоит в том, чтобы упростить реализацию программ-агентов: вместо того чтобы каждый раз строить агента с нуля, разработчик получает готовый фреймворк.

Важно также упомянуть понятие **когнитивной архитектуры** (cognitive architecture), которое тесно связано с архитектурой агента, но имеет самостоятельное значение. Когнитивные архитектуры описывают функциональную структуру познавательных процессов — прежде всего человека — с возможностью отображения этой структуры на интеллектуальных агентов. История когнитивных архитектур насчитывает несколько десятилетий: первые разработки относятся к 1960–1970-м годам, и ряд архитектур (например, SOAR) существует и активно развивается по сей день. Более подробно когнитивные архитектуры будут рассмотрены в главе 10 настоящего курса.

Принято выделять два базовых типа архитектур агентов: **реактивная** (reactive) и **рассуждающая** (deliberative). В русскоязычной литературе для второго типа используются также варианты: «мыслящий агент», «консультативный агент», «совещательный агент» — все они являются переводами английского термина «deliberative», который буквально означает «обдуманый», «взвешенный».

На основе этих двух базовых типов строятся более сложные **производные архитектуры**: гибридные системы, совмещающие свойства реактивных и рассуждающих агентов, а также многоагентные системы, в которых агенты различных типов взаимодействуют друг с другом. Рассмотрим базовые типы подробнее.

3.3. Реактивные агенты

Простой реактивный агент (simple reactive agent) реализует в наиболее прямолинейном виде идею отображения перцепции в действие. Его программа представляет собой набор правил вида «если воспринято A, то выполнить действие X». При этом никакой внутренней памяти, никакого представления о состоянии среды не предполагается: агент реагирует на текущий перцепт, не учитывая предыстории (рисунок 3.1). Формально функция выбора действия простого реактивного агента записывается так: *SelectAction: Event → Action*. То есть действие является функцией только текущего события (перцепта).



Рисунок 3.1 — Простой реактивный агент: единственный перцепт порождает единственное действие

Когда агент воспринимает несколько независимых сигналов одновременно, каждый из которых должен породить реакцию, возникает вопрос о том, как комбинировать полученные действия. Возможны три подхода, хорошо иллюстрируемые следующими схемами (рисунок 3.2).

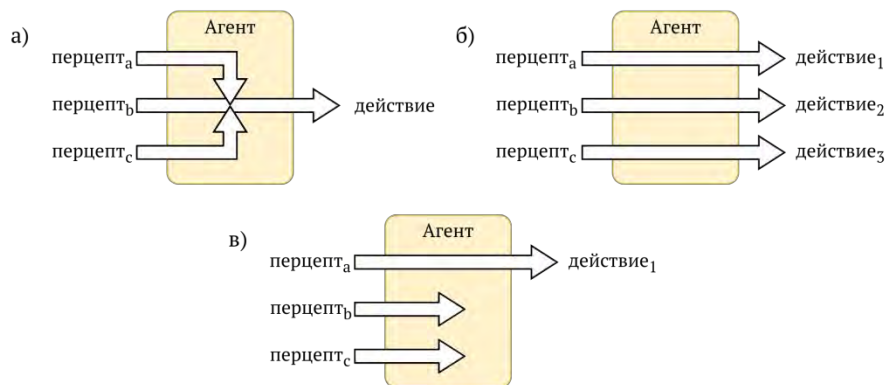


Рисунок 3.2 — Действия реактивных агентов при нескольких перцептах

Первый подход — **суммирование (комбинирование) действий**. Если действия имеют физический смысл вектора (например, ускорения в пространстве), их можно сложить и получить результирующее воздействие на агента. Второй подход — **параллельное выполнение**: агент одновременно совершает все порождённые реакции. Это возможно, когда действия не противоречат друг другу. Третий подход — **приоритизация**: из нескольких сработавших правил выбирается одно, наиболее приоритетное. Остальные правила откладываются до следующего цикла. Здесь важно учитывать, что пока выполняется первое действие, перцепты для оставшихся правил могут измениться, что повлияет на их срабатывание.

Пример: тележка Брайтенберга. Классическим примером простого реактивного агента служит **тележка Брайтенберга** (Braitenberg vehicle), предложенная итальянским нейробиологом Валентино Брайтенбергом. В своей книге [2] он описал мысленные эксперименты с примитивными роботами, у которых сенсоры напрямую соединены с двигателями колёс. Конструкция тележки чрезвычайно проста: два колеса, два сенсора (например, чувствительных к свету) и проводное соединение между ними. В

зависимости от типа соединения (прямого или перекрёстного) и знака связи (возбуждающей или тормозящей) тележка демонстрирует принципиально различное поведение по отношению к источнику света (рисунок 3.3).

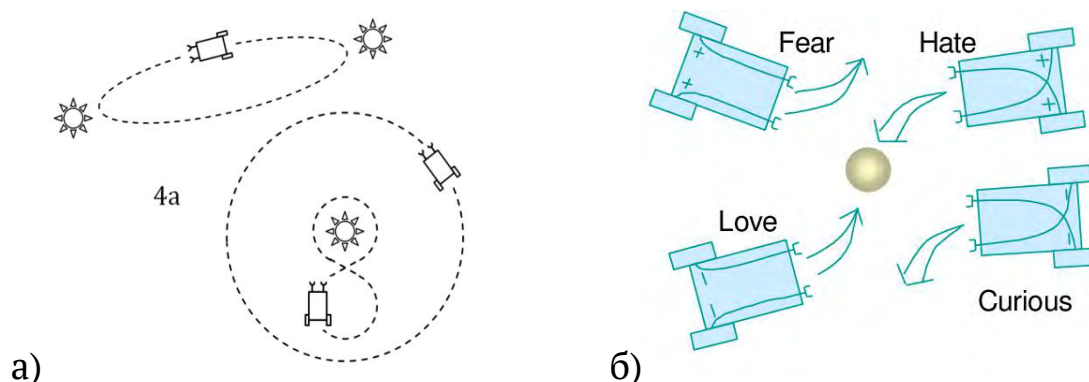


Рисунок 3.3 — Поведение тележки Брайтенберга а) траектории тележки при различных конфигурациях связей: тележка может двигаться к источнику света, вращаться вокруг него или обходить его стороной (Источник: [2]); б) четыре типа поведения тележки в зависимости от схемы соединения сенсоров и моторов: «страх», «ненависть», «любовь», «любопытство» (Источник: [12])

Броский парадокс эксперимента Брайтенберга состоит в следующем: тележка, лишённая какой-либо внутренней логики и программы, демонстрирует поведение, которое допускает весьма «человеческую» интерпретацию. Тележка с прямой возбуждающей связью «притягивается» к свету — её поведение напоминает «любопытство» или «любовь». Тележка с перекрёстной тормозящей связью, напротив, «убегает» от источника — что интерпретируется как «страх». Таким образом, уже через простейшие физические правила мы наблюдаем элементы **эмерджентного поведения** — поведения, не заложенного явно, но возникающего из взаимодействия простых компонентов с окружающей средой. Мысленные эксперименты Брайтенберга быстро переросли в реальные физические и компьютерные опыты, в которых исследователи варьировали число и типы сенсоров, схемы соединений и характеристики среды. Эта простая модель позволила подробно изучить диапазон поведений, достижимых на уровне простейших реактивных агентов.

Пример: модель Boids. Ещё одним ярким примером простого реактивного агента является **модель Boids**, разработанная Крейгом Рейнольдсом в 1986 году [3]. Модель описывает поведение стаи птиц (или косяка рыб) с помощью всего трёх правил, применяемых к каждому агенту («боиду», от «bird-oid»): **1. Разделение (Separation)** —

агент стремится избегать столкновения с ближайшими соседями: если сосед оказывается ближе допустимого расстояния, агент отклоняется от него. **2. Выравнивание (Alignment)** — агент согласовывает направление своего движения с соседями: вектор скорости агента стремится к среднему вектору скорости соседей в радиусе восприятия. **3. Сплочённость (Cohesion)** — агент стремится к центру группы: он движется в направлении среднего положения соседей. Каждое из трёх правил порождает вектор ускорения. Итоговое ускорение агента определяется как сумма трёх векторов с учётом ограничения на максимальное ускорение. Формально: $a_i = clip(w_1 \cdot F_{sep} + w_2 \cdot F_{align} + w_3 \cdot F_{cohes}, a_{max})$, где w_1, w_2, w_3 — веса правил, $F_{sep}, F_{align}, F_{cohes}$ — соответствующие векторы силы, a_{max} — максимально допустимое ускорение.

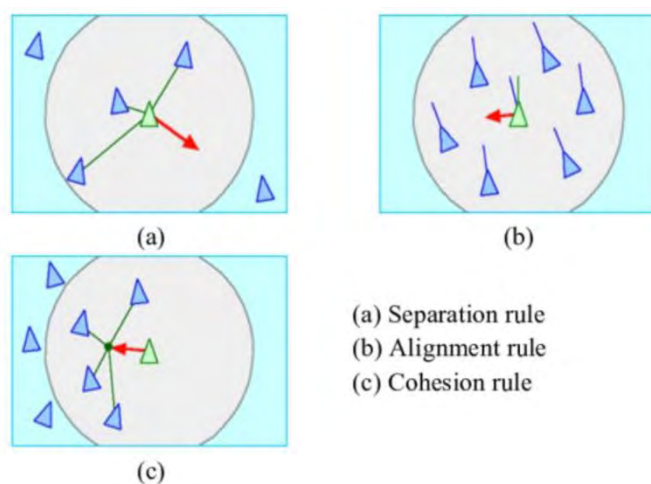


Рисунок 3.4 — Иллюстрация трёх правил модели Boids: каждый агент (зелёный треугольник) вычисляет результирующий вектор ускорения на основе трёх правил по отношению к соседям (синие треугольники)
(Источник: [3])

Несмотря на предельную простоту правил, модель воспроизводит сложное коллективное «стадное» поведение, наблюдаемое у птиц и рыб: образование стай, их деление и слияние, огибание препятствий. Агент в модели Boids не обладает ни памятью, ни представлением о среде: его единственная «программа» — это три вышеуказанных правила, отображающих воспринятые позиции соседей в вектор ускорения. Модель Boids является одним из наиболее известных примеров **эмерджентного коллективного поведения** в агентном моделировании.

Преимущества и ограничения простых реактивных агентов. Простые реактивные агенты обладают рядом существенных достоинств. Прежде всего, это **простота архитектуры**: реализация сводится к набору правил или непосредственному соединению сенсоров с

актуаторами, что позволяет воплощать агентов как программно, так и физически (в виде реального робота или электронной схемы). Следствием этой простоты является **высокая скорость реакции**: агент не тратит время на вычисления, что критически важно в системах реального времени. Кроме того, **не требуется явного представления внешней среды**: агент не хранит карту мира и не строит планов.

Вместе с тем простые реактивные агенты имеют принципиальные ограничения. **Фиксированность реакций** означает, что поведение агента детерминировано и не может адаптироваться к непредвиденным ситуациям. **Необходимость предварительного определения всех реакций** ограничивает область применения: разработчик должен заранее предусмотреть все возможные сочетания перцептов, что практически невозможно в сложных динамичных средах. Наконец, подобные агенты **не способны решать задачи**, требующие долгосрочного планирования, анализа альтернатив или работы с неопределённостью.

3.4. Реактивные агенты с внутренним состоянием

Введение памяти. Очевидный путь к преодолению ограничений простого реактивного агента — наделить его **внутренним состоянием** (памятью). Идея состоит в том, что на вход функции выбора действия подаётся не только текущий перцепт, но и текущее состояние агента, а на выходе формируется не только действие во внешней среде, но и новое состояние внутренней памяти (рисунок 3.5). Формально: $SelectAction: Event \times State \rightarrow Action \times State$.

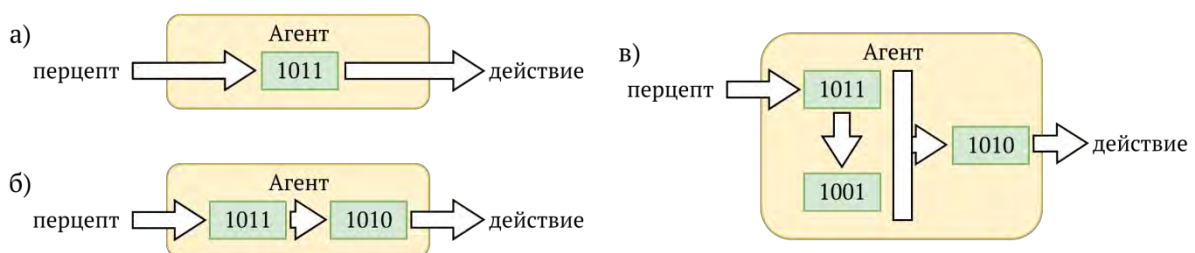


Рисунок 3.5 — Реактивный агент с внутренними состояниями: а) базовый вариант; б) изменение внутреннего состояния; в) несколько взаимосвязанных переменных

Введение памяти открывает значительно более широкие возможности. Во-первых, агент может **хранить историю перцептов**, что позволяет обнаруживать изменения в среде (например, зафиксировать, что объект исчез из поля зрения). Во-вторых, он может хранить **внутренние переменные** (счётчики, флаги), реализуя тем

самым конечные автоматы. При достаточно сложной организации памяти система становится эквивалентной **машине Тьюринга**, а значит, принципиально способна выполнять любую вычислимую программу. Различают **статическое** и **динамическое** внутреннее состояние. Статическое состояние задаётся один раз и не изменяется в ходе работы агента (например, фиксированные параметры). Динамическое состояние обновляется на каждом шаге, отражая текущие накопленные данные агента. Важная возможность, открываемая памятью, — **хранение истории воздействий на среду** и их результатов. Агент может различать, встречал ли он уже данное состояние среды и как в нём себя вёл, что позволяет накапливать своеобразный «опыт» и избегать повторных ошибок.

Проблема локальных минимумов. Одной из характерных трудностей реактивных агентов является **застревание в локальных минимумах** — ситуациях, когда силы, действующие на агента (притяжение к цели и отталкивание от препятствий), суммируются в нулевой вектор. Рассмотрим пример, показанный на рисунках ниже (рисунок 3.6). Агент движется к цели; встречая стену, перпендикулярную направлению движения, он оказывается в состоянии равновесия: сила притяжения к цели уравнивается реакцией препятствия. Агент «застревает» и не может продолжить движение.

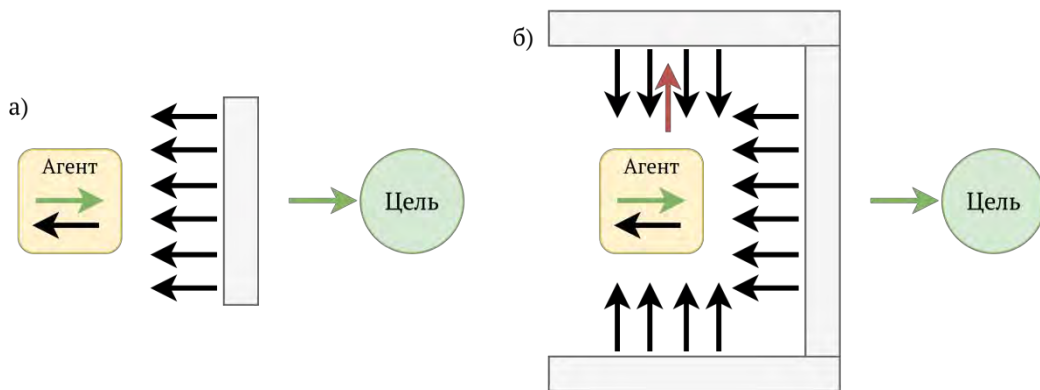


Рисунок 3.6 — Локальный минимум а) типа «стена»: агент застрял между стеной и целью, сумма векторов равна нулю; б) типа «тупик» (U-образное препятствие): агент попал в «карман», из которого простое правило отталкивания не помогает выйти

Для борьбы с застреванием применяются два основных подхода. Первый — **рандомизация**: к результирующему вектору действия добавляется случайная составляющая, которая «выбивает» агента из локального минимума. Метод прост в реализации и нередко

эффективен, однако приводит к неоптимальным траекториям. Второй — **использование памяти для «маркировки» негативных состояний**: посещённые позиции, в которых агент застрял или потратил много времени без продвижения к цели, помечаются как «нежелательные» и начинают «отталкивать» агента. Комбинация этих подходов позволяет реактивному агенту с памятью в конечном счёте достигать цели даже в сложных конфигурациях среды — пусть и неоптимальным путём.

Архитектура подчинения (Subsumption Architecture). Одним из значимых вкладов в развитие реактивных архитектур стала **архитектура подчинения** (Subsumption Architecture), предложенная Родни Бруксом в 1986 году [4]. Ключевая идея состоит в том, что поведение агента организуется как **иерархия модулей поведения**, каждый из которых реализован в виде расширенного конечного автомата (Augmented Finite State Machine, AFSM) - рисунок 3.7.

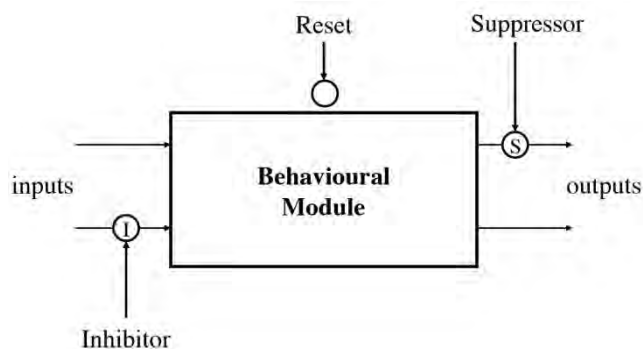


Рисунок 3.7 — Базовый модуль поведения в архитектуре подчинения: inputs/outputs — входные и выходные сигналы, Reset — сигнал сброса, Inhibitor — сигнал подавления входа, Suppressor — сигнал замещения выхода (Источник: [4])

Модули располагаются в **стеке**: более высокие уровни могут **ускорять** или **тормозить** нижние слои, а также **подавлять** их выходы (suppressor) или **замещать** входы (inhibitor). Тем самым низкоуровневые (примитивные) реакции продолжают функционировать в фоновом режиме, а высокоуровневые паттерны поведения переключают управление на себя только при необходимости.

Типичный пример применения архитектуры подчинения — команда роботов, задача которых состоит в сборе объектов в пространстве и доставке их на базу (рисунок 3.8). Иерархия уровней при этом может выглядеть следующим образом (снизу вверх): **Wander** (случайное блуждание по пространству) → **Avoiding** (избегание столкновений) → **Pickup** (захват обнаруженного объекта) → **Homing** (возвращение на базу с объектом). Каждый верхний уровень

«подчиняет» нижележащий лишь при срабатывании соответствующего условия. Достоинства такой организации очевидны: система надёжна (при сбое высокоуровневого модуля нижние уровни продолжают функционировать), модульна и допускает постепенное усложнение. Выходы модулей суммируются по аналогии с моделью Voids, что делает подход особенно удобным для физических роботов.

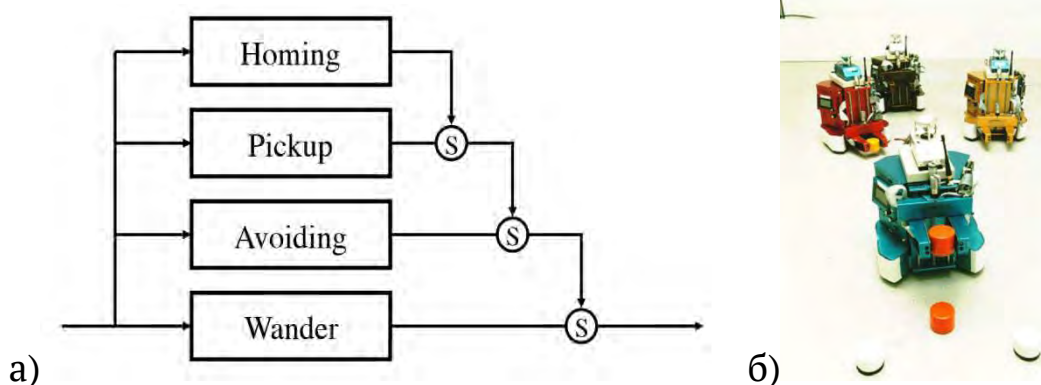


Рисунок 3.8 — Пример архитектуры подчинения для робота-собирающего объектов: а) четыре уровня (Wander, Avoiding, Pickup, Homing) иерархически управляют выходными сигналами посредством механизмов suppression (S); б) физические роботы, реализующие архитектуру подчинения: коллективная система для сбора и транспортировки объектов (Источник: [13])

Оценка реактивных архитектур. Обобщая рассмотренные подходы, можно сформулировать сильные и слабые стороны реактивных агентов в целом. К **преимуществам** относятся разнообразие и сложность порождаемого поведения (включая эмерджентное), простота разработки и реализации, пригодность для систем реального времени и возможность физического воплощения. **Ограничения** определяются принципиальной зависимостью от заранее прописанных правил: всё поведение должно быть предопределено разработчиком, новые непредвиденные состояния среды порождают ошибки, а для реализации по-настоящему сложной логики программа агента становится громоздкой и трудно поддерживаемой. Сложность работы с неопределённостью и принципиальное ограничение класса решаемых задач подводят нас к необходимости другого подхода — рассуждающих агентов.

3.5. Рассуждающие агенты

Рассуждающий агент (deliberative agent) принципиально отличается от реактивного тем, что осуществляет **явное рассмотрение**

альтернативных путей действия. Вместо того чтобы немедленно реагировать на перцепт, он строит гипотетические сценарии развития событий, оценивает их с точки зрения достижения цели и выбирает наилучший план (рисунок 3.9).

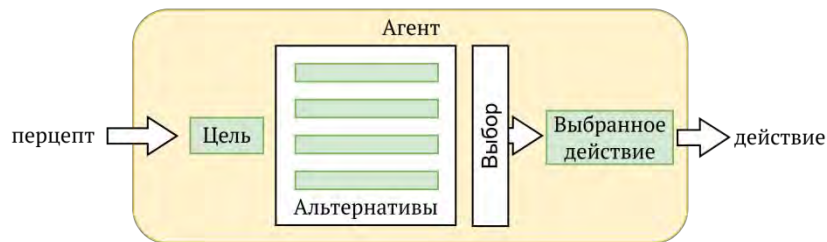


Рисунок 3.9 — Схема рассуждающего агента: перцепт преобразуется в цель (goal), затем генерируется множество альтернативных действий (alternatives), из которых выбирается (selection) наилучшее

Для того чтобы рассуждающий агент мог работать корректно, ему необходимы следующие возможности. Во-первых, **модель среды** — внутреннее представление мира, достаточно полное для планирования. Без такой модели агент не может «проигрывать» гипотетические сценарии. Во-вторых, **модель себя** — представление агента о своих возможностях, ресурсах и ограничениях. В-третьих, **механизм логического вывода** — способность генерировать и оценивать альтернативные цепочки действий, выбирая среди них предпочтительные. В-четвёртых, **работа с контрфактическими (противоречивыми, counterfactual) сценариями**: для правильной оценки планов агент должен уметь рассматривать не только хорошие варианты, но и их антиподы — ситуации, которые приводят к нежелательным исходам. Именно контрфактические сценарии задают «градиент», позволяющий оптимизировать выбор. В-пятых, **композиционная семантика** — способность интерпретировать состояния мира и цели на основе структурированной системы понятий. Для агента, получающего задачу в языковой форме (например, «передвинь жёлтый кубик ближе к зелёному»), это предполагает наличие некоторого символического представления объектов и их свойств.

Пример: задача коммивояжёра. Хрестоматийным примером задачи, решение которой иллюстрирует мышление рассуждающего агента, является **задача коммивояжёра** (Travelling Salesman Problem, TSP). Агент должен найти кратчайший замкнутый маршрут, проходящий через все заданные вершины графа ровно по одному разу. Задача является NP-сложной: перебор всех возможных маршрутов имеет факториальную сложность, что делает точное решение

неприемлемым при большом числе вершин. Тем не менее существуют эффективные **эвристические приближения**: алгоритм ветвей и границ, метод ближайшего соседа, алгоритмы муравьиных колоний и т. д. В агентном подходе итеративное улучшение маршрута можно интерпретировать как последовательность рассуждений агента о лучшем плане действий. Реактивный агент, напротив, справился бы с аналогичной задачей лишь за экспоненциально долгое время, если вообще справился бы.

Пример: робот Shakey. Исторически первым роботом, воплотившим концепцию рассуждающего агента, стал **Shakey** — проект Стэнфордского исследовательского института (SRI International), реализованный в 1966–1972 годах. Shakey стал первым роботом, способным рассуждать о своих действиях в реальном мире.

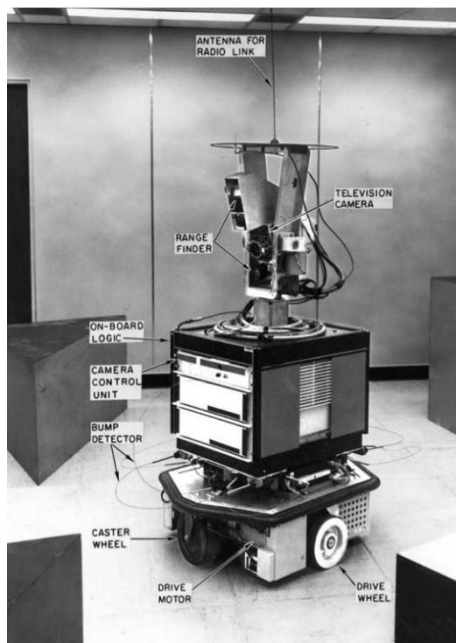


Рисунок 3.10 — Робот Shakey (SRI, 1966–1972): один из первых роботов, использовавших рассуждение о своих действиях. Оснащён телекамерой, дальномером, тактильными датчиками и антенной радиосвязи (Источник: [14])

Shakey был оснащён телевизионной камерой, дальномером и датчиками касания. В качестве вычислительной базы использовался суперкомпьютер PDP-10, работавший под управлением программы на языках Fortran и LISP. Ключевым достижением являлась возможность задавать роботу команды на естественном языке в абстрактной форме: «передвинь такой-то объект к такому-то месту». Для выполнения этой команды робот должен был самостоятельно разработать план действий, включая освобождение прохода, выбор подходящего объекта для

манипуляции и т. д. В основе логики Shakey лежала система планирования **STRIPS** (Stanford Research Institute Problem Solver), разработанная Файксом и Нильссоном в 1971 году. STRIPS использует символьную модель мира (формализованную в виде предикатов первого порядка), обновляемое представление о текущем состоянии мира (через частично наблюдаемые данные) и **регрессионное планирование** — построение плана от цели к начальному состоянию путём декомпозиции на подзадачи. Модель мира была статической, детерминированной, без внешних агентов — идеальные условия «классического» планирования.

Предположения классического планирования и их ограничения. Подход, реализованный в Shakey, опирался на ряд идеализирующих предположений. **Статичность мира** — среда не меняется в процессе работы агента. **Детерминированность** — каждое действие приводит к строго определённом результату. **Безошибочность воплощения плана** — спланированные действия всегда выполняются именно так, как задумано. **Неограниченное время планирования** — агент может сколь угодно долго строить и пересматривать план.

В реальных условиях ни одно из этих предположений, как правило, не выполняется. Среда **динамически изменяется** — и что особенно важно, она меняется уже в ходе восприятия, планирования и совершения действий. Действия могут **давать неожиданные результаты**. Планирование **ограничено во времени**. Наконец, данные о среде зачастую **неполны, неточны или ошибочны**.

Греческий философ Гераклит замечал, что в одну и ту же реку нельзя войти дважды. Применительно к рассуждающему агенту это означает: к тому моменту, когда агент завершает планирование, мир уже изменился по сравнению с тем, каким он был в начале планирования.

Существует четыре принципиальных подхода к работе с неидеальными данными о мире. Первый — **игнорирование**: агент продолжает действовать по плану, не замечая расхождений с реальностью. Это допустимо только при незначительных отклонениях. Второй — **улучшение перцепции**: установка более точных сенсоров или использование более детальной модели мира. Третий — **явное рассуждение о неопределённости**: агент формализует неопределённость (через вероятность, правдоподобие, риск) и включает её в процесс планирования. Четвёртый — **интерактивное перепланирование с малым шагом**: агент перестраивает план через

короткие промежутки времени, обновляя своё представление о мире. Основная сложность здесь — временные затраты на перепланирование.

Пример: боты для Quake. Ярким примером рассуждающего агента в практическом применении служат боты для многопользовательской игры Quake [8].

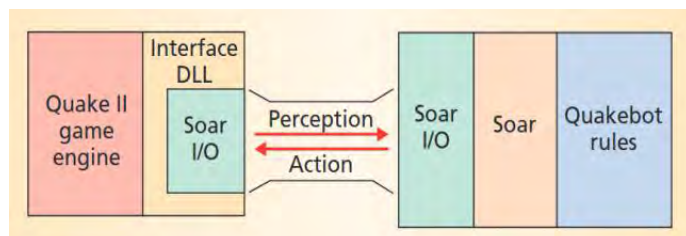


Рисунок 3.11 — Архитектура Quakebot: между игровым движком Quake II и агентом (Quakebot rules + SOAR) реализован интерфейс DLL для обмена перцептами и действиями (Источник: [8])

Боты получали те же входные данные, что и человек-игрок, и должны были действовать в условиях реального времени. Базовым инструментом рассуждения служила когнитивная архитектура **SOAR** (State, Operator And Result) — одна из наиболее известных когнитивных архитектур, предложенная Дж. Ленердом и А. Ньюэллом в 1987 году [15]. SOAR реализует цикл «воспринять — выбрать оператор — применить оператор», позволяя агенту строить иерархические цепочки рассуждений.

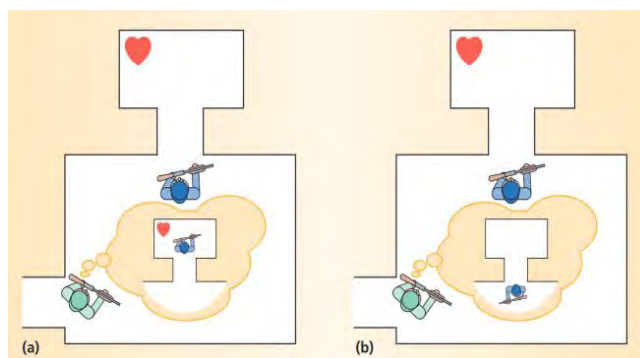


Рисунок 3.12 — Теория сознания (Theory of Mind) в Quakebot: агент строит модель поведения оппонентов, рассуждая о том, что они видят и что могут предпринять (Источник: [8])

Ключевым достижением проекта стало использование концепции **Theory of Mind** — агент рассуждал не только о своих действиях, но и о возможных действиях оппонентов. Он оценивал: если противник пойдёт в этом направлении, он окажется в поле моего зрения; значит, нужно занять позицию X. Такая логика в сочетании с иерархией стратегии, тактики и оперативного управления позволила ботам вести

себя вполне правдоподобно уже во второй версии игры, конкурируя с опытными игроками-людьми.

Проблема времени рассуждения. Принципиальной сложностью рассуждающих агентов является **вычислительная стоимость** построения и оценки альтернативных планов. По мере роста числа возможных альтернатив и горизонта планирования сложность задачи растёт экспоненциально, что в пределе приводит к «бесконечной генерации» сценариев. В непрерывных средах (таких как пространство физического движения) число альтернатив буквально бесконечно.

Дополнительную сложность создаёт необходимость учитывать **динамику мира**: для каждого возможного действия агента мир также эволюционирует, что порождает экспоненциально ветвящееся дерево состояний. Теоремой о «no free lunch» в оптимизации гарантируется, что не существует универсально оптимального алгоритма поиска, не предполагающего компромиссов.

Практических подходов к решению этой проблемы несколько. Первый — **игнорирование**: если пространство вариантов дискретно и невелико, полный перебор допустим. Второй — **эвристические методы и суррогатные модели**: вместо точного поиска оптимума используются приближённые методы (эвристики, нейросетевые суррогаты, методы прецедентов), снижающие вычислительную стоимость. Третий — подход **Empirical AI**: предварительная оценка сложности задачи и адаптивный выбор алгоритма в зависимости от прогнозируемого времени. Четвёртый — **алгоритмы с прерыванием** (Anytime Algorithms): алгоритм непрерывно генерирует и улучшает решения; в любой момент времени он может быть прерван, и будет возвращено лучшее из найденных на текущий момент. Такой подход позволяет рационально использовать вычислительный ресурс, отдавая агенту решение ровно того качества, которое соответствует отведённому на планирование времени.

Концепция BDI. Важнейшей концептуальной основой рассуждающих агентов является **модель BDI** (Belief–Desire–Intention — Убеждения–Желания–Намерения), предложенная Братманом [16] и получившая формализацию применительно к агентам в работах Рао и Джорджеффа [7]. **Убеждения** (Beliefs) — это представление агента о состоянии среды и о самом себе. Убеждения могут быть неполными или неточными; они обновляются по мере поступления новых перцептов. **Желания** (Desires) — это цели, к достижению которых стремится агент. Желания могут быть противоречивы: например, агент хочет одновременно зарядить аккумулятор и продолжить выполнение задачи.

Намерения (Intentions) — это активные цели, к реализации которых агент фактически приступает. Намерения представляют собой подмножество желаний, выбранных с учётом текущих убеждений и возможностей. Именно намерения определяют поведение агента в данный момент. Концепция BDI привлекает своей интуитивной близостью к описанию человеческого практического рассуждения и обеспечивает **прозрачность** поведения агента: наблюдатель может понять, почему агент поступает именно так, проследив цепочку от убеждений через желания к намерениям.

Пример: архитектура IRMA (Intelligent Resource-bounded Machine Architecture) была разработана Майклом Джорджеффом и Эми Лэндманн [6] как практическая реализация BDI-концепции с явным механизмом управления вычислительными ресурсами.

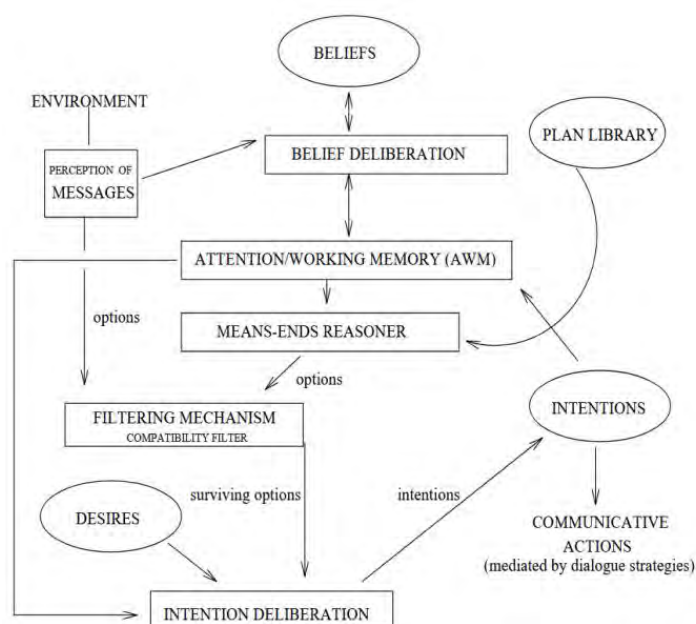


Рисунок 3.13 — Полная схема архитектуры IRMA: цикл включает восприятие среды (Perception of Messages), обновление убеждений (Belief Deliberation), рассуждение о средствах достижения целей (Means-Ends Reasoner), фильтрацию вариантов (Filtering Mechanism) и формирование намерений (Intention Deliberation). Ключевую роль играет рабочая память (Attention/Working Memory)
(Источник: [6])

Цикл работы IRMA состоит из следующих этапов. На первом этапе агент получает информацию о состоянии среды и обновляет свои убеждения. На втором — **механизм рассуждения о средствах и целях** (Means-Ends Reasoner) генерирует множество возможных планов достижения текущих желаний. На третьем — критически важный

механизм фильтрации (Filtering Mechanism) отсеивает несовместимые с текущими намерениями или заведомо нежелательные варианты. Оставшиеся «выжившие» варианты поступают на процедуру **рассуждения о намерениях** (Intention Deliberation), где из них выбирается итоговое намерение и строится план. Ключевое нововведение IRMA — механизм фильтрации — позволяет существенно снизить вычислительную сложность: вместо оценки всего пространства альтернатив агент работает лишь с их совместимым с текущими намерениями подмножеством. Это делает архитектуру приемлемой для практического применения в системах реального времени.

3.6. Гибридные архитектуры

Ни реактивные, ни рассуждающие архитектуры в чистом виде не обеспечивают оптимального баланса между скоростью реакции и качеством принятых решений. **Гибридные архитектуры** призваны сочетать преимущества обоих подходов. Ключевым вопросом при гибридизации является **организация взаимодействия компонентов**. Типовой подход — **многослойная (многоуровневая) архитектура**, в которой слои организованы иерархически. При этом возможны три варианта управления слоями (рисунок 3.14). Первый — **независимое (параллельное) управление**: каждый слой обрабатывает входные данные и генерирует действия параллельно; итоговое действие формируется как результат некоторой комбинации выходов всех слоёв.

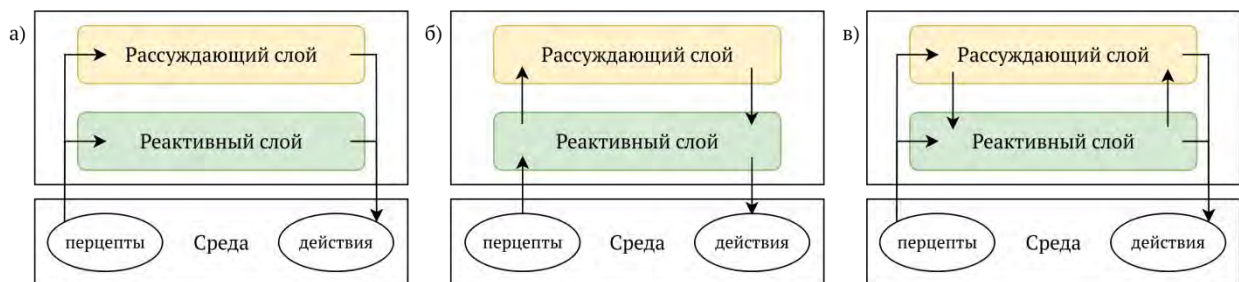


Рисунок 3.14 — Гибридные архитектуры а) с параллельным управлением; б) с иерархическим управлением; в) с конкурирующим управлением

Второй — **иерархическое управление**: нижний (реактивный) слой реагирует быстро и передает информацию вверх; верхний (рассуждающий) слой формирует высокоуровневые цели и ограничения, спускаемые вниз. Третий — **конкурирующее управление**: слои соревнуются за право принять решение; побеждает тот, чье «предложение» удовлетворяет некоторому критерию качества. Гибридные архитектуры позволяют одновременно обеспечить **быструю реакцию** на изменения среды (за счёт реактивного слоя) и

качественное долгосрочное планирование (за счёт рассуждающего слоя). Платой за это является повышенная сложность разработки, тестирования и настройки.

Пример: робот Xavier. Примером практической реализации гибридной многослойной архитектуры служит робот Xavier, разработанный в Университете Карнеги-Меллон [9]. Xavier оснащён ударными датчиками, одометром, камерой, четырьмя колёсами и голосовым интерфейсом.

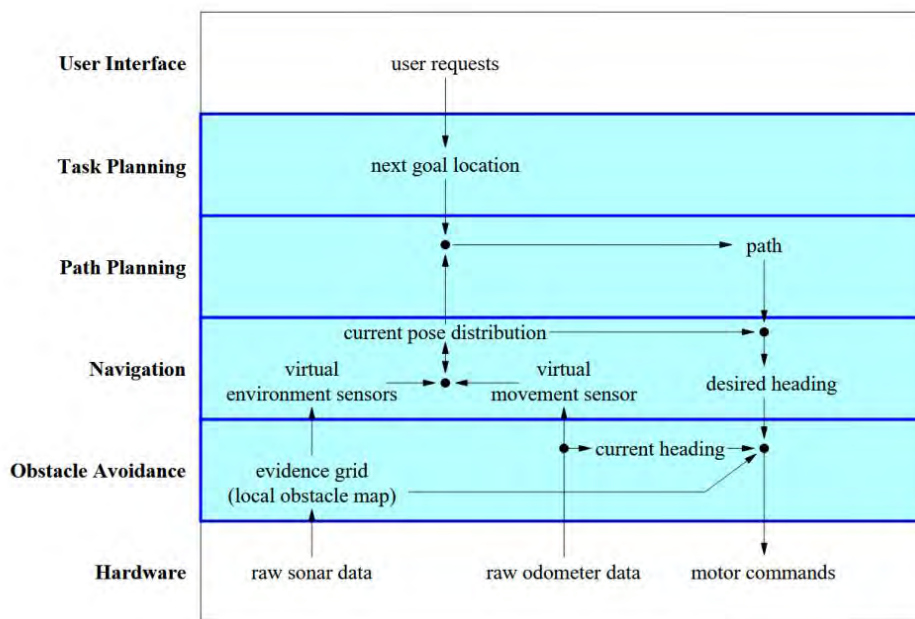


Рисунок 3.15 — Многослойная архитектура управления роботом Xavier: пять уровней (Hardware, Obstacle Avoidance, Navigation, Path Planning, Task Planning) с вертикальной передачей данных и управления (Источник: [9])

Архитектура Xavier включает пять уровней (рисунок 3.15). Нижний уровень (**Hardware**) — необработанные данные сенсоров (ультразвуковые данные, одометр) и команды моторам. Следующий уровень (**Obstacle Avoidance**) — реактивное избегание препятствий на основе локальной карты препятствий (evidence grid). Уровень **Navigation** осуществляет локализацию робота и вычисляет желаемый курс движения. Уровень **Path Planning** строит путь к следующей целевой точке. Верхний уровень (**Task Planning**) принимает запросы пользователя и декомпозирует их на последовательность целевых локаций. Такая вертикальная организация является образцом многослойной гибридной архитектуры и в той или иной форме воспроизводится в большинстве современных автономных роботов, включая бытовые роботы-пылесосы.

Пример: архитектура InteRRaP. Более сложным примером гибридной архитектуры, охватывающей также взаимодействие с другими агентами, служит InteRRaP [10]. Эта трёхуровневая архитектура объединяет реактивное поведение, планирование и кооперацию агентов (рисунок 3.16). Нижний уровень — **компонент на основе поведения** (Behaviour-based Component, BBC) — реализует реактивные паттерны поведения, задаваемые мировой моделью (ситуационным контекстом). Средний уровень — **план-ориентированный компонент** (Plan-based Component, PBC) — осуществляет локальное планирование и рассуждение с использованием базы локальных целей и планов. Верхний уровень — **компонент кооперации** (Cooperation Component, CC) — управляет взаимодействием с другими агентами, используя знания о совместных целях и социальном контексте. Все три уровня взаимодействуют через **интерфейс с миром** (WIF — World Interface), обеспечивающий единый доступ к перцепции, коммуникации и действиям. InteRRaP применяется в задачах коллективной навигации транспортных средств, управления группами роботов и моделирования кооперативного социального поведения.

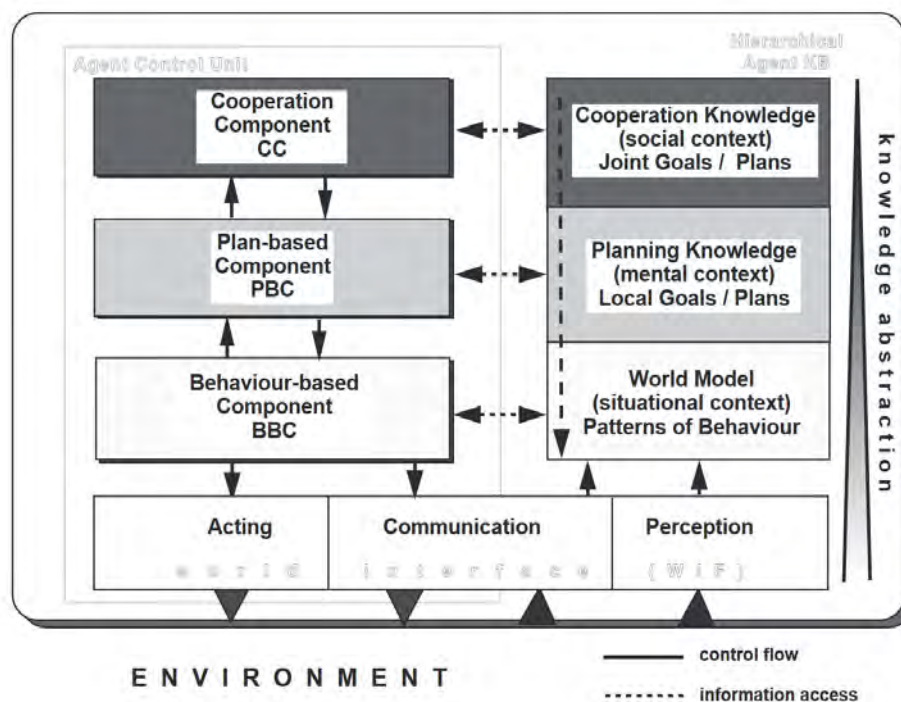


Рисунок 3.16 — Архитектура InteRRaP: три уровня управления (BBC — реактивное поведение, PBC — план-ориентированное, CC — кооперация) и соответствующие базы знаний (мировая модель, локальные цели/планы, совместные цели/планы). (Источник: [10])

3.7. Заключение

В настоящей лекции мы рассмотрели основные архитектурные подходы к построению агентов. Позволим себе обобщить ключевые результаты. **Архитектура агента** задаёт базовые принципы построения программы агента, определяет доступный функционал и ограничения. В терминах объектно-ориентированного программирования архитектура — это класс программ-агентов, согласованный со средой и множеством решаемых задач. **Реактивные агенты** реализуют непосредственное отображение перцепции в действие. Их сила — в простоте, быstroдействии и возможности порождать сложное (в том числе эмерджентное) поведение через простые правила, ограничение — в необходимости полного предварительного определения реакций и принципиальной неспособности работать с долгосрочными планами и неопределённостью. Введение внутреннего состояния (памяти) частично решает эти проблемы, расширяя класс реализуемых поведений. **Рассуждающие агенты** поднимаются на уровень явного планирования и оценки альтернатив. Они способны решать задачи в динамичных, неопределённых средах, строить долгосрочные планы и адаптировать их к изменениям. Платой за эти возможности является высокая вычислительная стоимость и сложность реализации. Ключевыми инструментами управления этой сложностью служат эвристические методы, алгоритмы с прерыванием и механизмы фильтрации альтернатив (как в IRMA). Концепция BDI предоставляет логически прозрачную основу для рассуждающих агентов. **Гибридные архитектуры** сочетают оба подхода, организуя их в многослойные иерархии с различными вариантами управления (независимым, иерархическим, конкурирующим). Именно гибридные архитектуры лежат в основе большинства современных автономных роботов и интеллектуальных агентов реального мира.

В следующей лекции мы подробнее остановимся на вопросах перцепции, действий и принятия решений агентов, рассматривая механизмы, с помощью которых агент получает информацию о среде и преобразует её в действия.

Таблица 3.2 — Сравнительная характеристика основных типов архитектур агентов

Архитектура	Принцип работы	Преимущества	Ограничения
Простой реактивный	Перцепт → Правило → Действие	Скорость, простота, эмерджентность	Нет памяти, фиксированные реакции
Реактивный с состоянием	Перцепт + Состояние → Действие + Состояние	Накопление опыта, история	Усложнение, ограниченный класс задач
Рассуждающий (deliberative)	Перцепт → Цель → Альтернативы → Отбор → Действие	Планирование, работа с неопределённостью	Вычислительные затраты, сложность
Гибридный	Реактивный + Рассуждающий (слои)	Баланс скорости и качества	Сложность разработки и настройки

Рекомендуемая литература и источники

1. Russell S.J. Artificial Intelligence: A Modern Approach / S.J. Russell, P. Norvig. — 4th ed. — Hoboken : Pearson, 2020. — 1132 p. — ISBN 978-0-13-461099-3. — URL: <https://aima.cs.berkeley.edu/>
2. Braitenberg V. Vehicles: Experiments in Synthetic Psychology / V. Braitenberg. — Cambridge : MIT Press, 1984. — 168 p. — ISBN 978-0-262-52112-3.
3. Reynolds C.W. Flocks, Herds and Schools: A Distributed Behavioral Model / C.W. Reynolds // Proceedings of ACM SIGGRAPH'87 (Computer Graphics). — New York : ACM, 1987. — Vol. 21, No. 4. — P. 25–34. — DOI: 10.1145/37402.37406. — URL: <https://dl.acm.org/doi/10.1145/37402.37406>
4. Brooks R.A. A Robust Layered Control System for a Mobile Robot / R.A. Brooks // IEEE Journal on Robotics and Automation. — 1986. — Vol. 2, No. 1. — P. 14–23. — DOI: 10.1109/JRA.1986.1087032. — URL: <https://doi.org/10.1109/JRA.1986.1087032>
5. Fikes R. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving / R. Fikes, N. Nilsson // Artificial Intelligence. — 1971. — Vol. 2, No. 3–4. — P. 189–208. — DOI: 10.1016/0004-3702(71)90010-5. — URL: [https://doi.org/10.1016/0004-3702\(71\)90010-5](https://doi.org/10.1016/0004-3702(71)90010-5)
6. Georgeff M.P. Reactive Reasoning and Planning / M.P. Georgeff, A.L. Lansky // Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87). — Seattle : AAAI Press, 1987. — P. 677–682.
7. Rao A.S. BDI Agents: From Theory to Practice / A.S. Rao, M.P. Georgeff // Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95). — San Francisco : AAAI Press, 1995. — P. 312–319.

8. Laird J.E. It Knows What You're Going to Do: Adding Anticipation to a Quakebot / J.E. Laird // AAAI Technical Report. — 2001. — URL: <http://dx.doi.org/10.1109/2.933506>
9. Simmons R. Xavier: An Autonomous Mobile Robot on the Web / R. Simmons [et al.] // IEEE Robotics & Automation Magazine. — 1998. — Vol. 5, No. 1. — P. 33–44. — DOI: 10.1145/272874.272878. — URL: <https://doi.org/10.1145/272874.272878>
10. Müller J.P. The Agent Architecture InteRRaP: Concept and Application : Technical Report RR-93-26 / J.P. Müller, M. Pischel. — Saarbrücken : German Research Center for Artificial Intelligence (DFKI), 1993.
11. Kotseruba I. 40 Years of Cognitive Architectures: Core Cognitive Abilities and Practical Applications / I. Kotseruba, J.K. Tsotsos // Artificial Intelligence Review. — 2020. — Vol. 53, No. 1. — P. 17–94. — DOI: 10.1007/s10462-018-9646-y. — URL: <https://doi.org/10.1007/s10462-018-9646-y>
12. Browning B. Neural Systems for Integrating Robot Behaviours / B. Browning, G. Wyeth // [Proceedings]. — 1998.
13. Mataric M.J. Interaction and Intelligent Behavior : Technical Report AI-TR-1495 / M.J. Mataric. — Cambridge : MIT AI Lab, 1994.
14. Application of Intelligent Automata to Reconnaissance : Technical Report, Project 5953 – Final Report / N.J. Nilsson [et al.]. — Menlo Park : Stanford Research Institute, 1968.
15. Laird J.E. SOAR: An Architecture for General Intelligence / J.E. Laird, A. Newell, P.S. Rosenbloom // Artificial Intelligence. — 1987. — Vol. 33, No. 1. — P. 1–64. — DOI: 10.1016/0004-3702(87)90050-6. — URL: [https://doi.org/10.1016/0004-3702\(87\)90050-6](https://doi.org/10.1016/0004-3702(87)90050-6)
16. Bratman M.E. Intention, Plans, and Practical Reason / M.E. Bratman. — Cambridge : Harvard University Press, 1987. — 200 p. — ISBN 978-0-674-45884-1.

Лекция 4. Перцепция, действия и решения агентов

Лекция посвящена трём ключевым механизмам, определяющим функционирование любого интеллектуального агента: перцепции, действиям и принятию решений. Понимание этих механизмов лежит в основе построения реалистичных агентных моделей и является необходимым условием для изучения многоагентных систем, обучения с подкреплением и когнитивных архитектур, которые будут рассматриваться в последующих главах курса. Агент существует в некоторой среде, с которой он взаимодействует посредством двух каналов: входного — через который он получает информацию о состоянии мира (перцепция), и выходного — через который он оказывает воздействие на среду (действия). Между этими двумя каналами располагается логика принятия решений, определяющая, какое именно действие агент должен предпринять в сложившейся ситуации. Качество и полнота функционирования каждого из трёх компонентов непосредственно влияет на эффективность и реалистичность поведения агента.

4.1. Перцепция агентов

Термин «перцепция» в контексте агентного моделирования обозначает процесс получения агентом информации об окружающей среде через сенсоры. Важно сразу оговориться: перцепция — это не просто пассивное «поступление данных», а целый цикл преобразований, начинающийся с физического сигнала и заканчивающийся внутренним представлением агента об окружающем мире. Формально, перцепция агента в момент времени t — это последовательность всех воспринятых им входных сигналов $p_1, p_2, \dots, p_t$, которую принято называть историей перцепции. Именно эта история (или её часть) лежит в основе любого последующего решения агента. Поведение агента, полностью определяемое его историей перцепции, называется табличным или перцепционным.

Модальность и источники сигналов. Важнейшей характеристикой перцепции является её модальность — тип информации, которую способны воспринимать сенсоры агента. В зависимости от сценария применения сенсоры могут работать с видеосигналом, аудиопотоком, текстовыми данными, числовыми показателями датчиков, бинарными сигналами или любой иной разновидностью информации. Модальность определяет не только то, что агент «видит», но и то, какие аспекты среды для него недоступны в

принципе. Например, агент, управляющий промышленным роботом, может получать информацию с камер (видеомодальность), датчиков силы и давления (тактильная модальность), акселерометров и гироскопов (модальность пространственного положения). Агент в финансовых системах, напротив, работает преимущественно с числовыми временными рядами рыночных котировок, объёмов торгов и макроэкономических показателей.

Зашумлённость, неопределённость и история перцепции.

Фундаментальная особенность перцепции — в отличие от абстрактной математической функции — состоит в том, что сигналы, получаемые от сенсоров, никогда не являются точным и полным отражением реального состояния среды. Этому способствуют несколько факторов. Во-первых, сенсоры обладают собственными техническими ограничениями: конечной разрешающей способностью, диапазоном чувствительности, систематической и случайной погрешностью измерений. Во-вторых, в процессе передачи сигнала от источника к сенсору неизбежно возникает шум — случайные искажения, вызванные физическими свойствами среды. В-третьих, агент воспринимает не текущее состояние мира, а состояние, зафиксированное в тот момент, когда сигнал достиг сенсора, — то есть всегда с некоторой задержкой. В динамических системах с высокой скоростью изменений это запаздывание может существенно влиять на корректность принимаемых решений. Помимо зашумлённости, важную роль играет история перцепции. Поскольку среда изменяется во времени, а каждый момент наблюдения даёт лишь частичный «срез» реальности, агент должен агрегировать накопленные наблюдения, чтобы поддерживать внутреннюю модель мира, достаточно точную для принятия решений.

Подходы к построению внутреннего образа: индукция и дедукция.

В теории агентного моделирования выделяют два принципиально различных подхода к тому, как агент выстраивает своё внутреннее представление о среде на основе поступающих сенсорных данных. Первый подход — **индуктивный**, или восходящий (bottom-up). Агент «собирает» образ окружающего мира снизу вверх, начиная с элементарных сигналов отдельных сенсоров и постепенно объединяя их в более сложные структуры. Такой подход характерен для нейросетевых систем восприятия, в которых признаки низкого уровня (пиксели, фонемы, числовые значения) объединяются в признаки высокого уровня (объекты, слова, паттерны рынка). Второй подход — **дедуктивный**, или нисходящий (top-down). Агент начинает с некоторого априорного представления о мире — «образца», «ожидания»

— и корректирует его по мере поступления новых сенсорных данных. Этот подход лучше соответствует байесовским методам обновления убеждений и моделям предсказательного кодирования, широко используемым в когнитивных науках. На практике оба подхода часто комбинируются: агент одновременно строит образ снизу вверх и сопоставляет его с предсказаниями, генерируемыми сверху вниз. Такая комбинация позволяет эффективнее обрабатывать неполную или зашумлённую информацию.

Внимание как характеристика перцепции. Существенной характеристикой перцепции является механизм внимания (attention). Внимание определяет, на каких аспектах получаемой информации агент концентрирует свои вычислительные ресурсы в конкретный момент времени. Это понятие связывает перцепцию с внутренней архитектурой агента и его текущими целями: агент обращает внимание на те элементы среды, которые наиболее релевантны для выполняемой задачи. Механизм внимания особенно важен при моделировании агентов с ограниченными вычислительными ресурсами или при работе с высокоразмерными сенсорными потоками, когда обработка всей доступной информации одновременно невозможна или нецелесообразна.

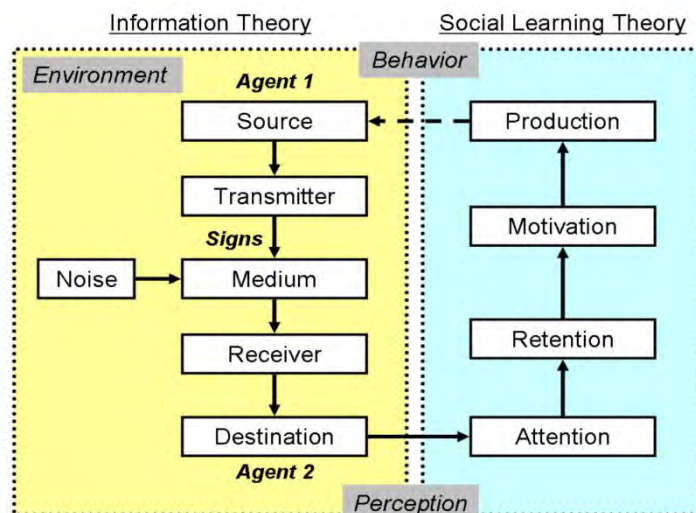


Рисунок 4.1 — Цикл перцепции в рамках теории информации и теории социального научения (Источник: [7])

На рисунке 4.1 изображена схема, объединяющая теорию информации и теорию социального научения в контексте перцепции агента. Левая часть (Information Theory) описывает физический канал передачи информации: среда (Environment) порождает источник сигнала (Source), который через передатчик (Transmitter) поступает в среду в виде знаков (Signs). Среда вносит шум (Noise), после чего сигнал

достигает приёмника (Receiver) и, наконец, пункта назначения (Destination) — то есть представления воспринятого агентом. Правая часть (Social Learning Theory, или, в более общем смысле, внутренняя архитектура поведения) описывает внутренние механизмы обработки: внимание (Attention) направляет полученную информацию к интерпретации, далее следует этап актуализации в памяти (Retention), на основе которого формируется мотивация (Motivation), завершающаяся выработкой поведенческого ответа (Production). Важно понимать, что программа агента «живёт» именно на стыке воспринятого представления и порождённого поведения. Всё, что происходит до этой границы, — область физической и информационной обработки сигналов; всё, что после — выходной канал воздействия на среду.

4.2. Психологические модели перцепции

Когда целью моделирования является воспроизведение поведения реальных агентов — прежде всего людей или человекоподобных систем — возникает необходимость выйти за рамки чисто технической трактовки перцепции и обратиться к психологическим моделям, описывающим когнитивные механизмы обработки информации.

Механизм внимания, памяти и мотивации. В психологических моделях поведения ключевую роль играют три взаимосвязанных компонента. **Внимание** (Attention) — способность агента выборочно концентрироваться на определённых стимулах из всего многообразия поступающих сигналов. Внимание является «фильтром», который определяет, какая часть сенсорного потока будет подвергнута глубокой когнитивной обработке, а какая отсеяна. В агентных моделях внимание часто реализуется через механизм весовых коэффициентов (attention weights), который лежит, в частности, в основе трансформерных архитектур в машинном обучении. **Удержание в памяти** (Retention) — процесс хранения, актуализации и извлечения информации из внутреннего хранилища агента. Это не просто пассивная «база данных», а активный механизм, связывающий вновь поступившую информацию с уже имеющимися знаниями и формирующий из них согласованное представление о мире. **Мотивация** (Motivation) — компонент, определяющий направление и интенсивность действий агента. В моделях поведения мотивация связывает внутреннее состояние агента (его потребности, цели, предпочтения) с процессом выработки ответного поведения. Именно мотивация объясняет, почему из нескольких возможных действий агент выбирает конкретное.

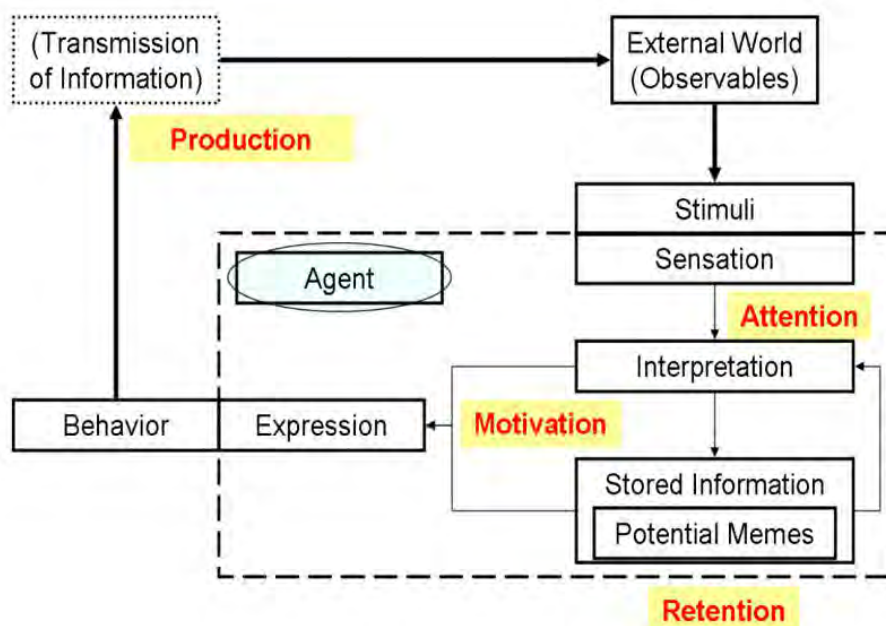


Рисунок 4.2 — Внутренние механизмы агента: внимание, удержание, мотивация и производство поведения (Источник: [9])

На рисунке 4.2 показана более детализированная схема внутренней архитектуры агента. Из внешнего мира (External World) поступают наблюдаемые стимулы, которые через процессы ощущения (Sensation) попадают в фокус внимания (Attention). Далее запускается интерпретация (Interpretation), которая обращается к хранилищу уже накопленной информации (Stored Information / Potential Memes) — этот цикл соответствует этапу удержания (Retention). На основе интерпретации и актуальных целей агента формируется мотивация (Motivation), которая управляет экспрессией (Expression) и итоговым поведением (Behavior). Результаты поведения возвращаются во внешний мир, образуя замкнутый контур взаимодействия.

Ресурсная теория внимания Канемана. Одной из наиболее влиятельных психологических моделей, применяемых при моделировании поведения агентов, является ресурсная теория внимания, разработанная Даниэлем Канеманом [2]. Согласно этой теории, внимание человека является ресурсно ограниченным: общий «бюджет» когнитивных ресурсов фиксирован, и распределение этого бюджета между одновременно выполняемыми задачами подчиняется определённым правилам.

Ключевое следствие ресурсной теории применительно к агентному моделированию состоит в том, что агент не способен одновременно в полной мере обрабатывать все поступающие сигналы. При перегрузке одного «канала» обработки качество другого снижается.

Это позволяет воспроизводить реалистичные паттерны человеческого поведения: ошибки в условиях высокой нагрузки, избирательное игнорирование части информации, деградацию качества решений в стрессовых ситуациях. Ресурсная теория внимания лежит в основе ряда когнитивных архитектур (ACT-R, SOAR), с которыми мы подробнее познакомимся в главе 10, посвящённой когнитивным состояниям агентов.

4.3. Действия агентов

Если перцепция представляет собой входной канал агента, то действия — его выходной канал, посредством которого агент оказывает воздействие на окружающую среду. Технически действия реализуются через актуаторы (actuators) — устройства или механизмы, преобразующие управляющие сигналы агента в физические или информационные изменения в среде.

Типы действий. В зависимости от природы взаимодействия со средой принято выделять два основных типа действий. **Физические действия** — это воздействия, связанные с перемещением агента или изменением физических объектов в среде. Для физических агентов (роботов, манипуляторов, транспортных средств) актуаторы имеют ограниченное число степеней свободы, работают в рамках физических ограничений по силе и скорости, обладают инертностью и могут вносить собственные погрешности. Например, рука робота с шестью степенями свободы не может точно воспроизвести команду при наличии люфта в сервоприводах. **Информационные действия** — это передача данных, сообщений, сигналов другим агентам или компонентам системы. Здесь основным ограничением выступает пропускная способность каналов связи и частота дискретизации. В большинстве абстрактных агентных моделей информационные действия полагаются мгновенными и безошибочными, однако при моделировании распределённых систем или систем с ненадёжными каналами это допущение необходимо снимать.

Свойства и ограничения действий. Действия агентов характеризуются рядом свойств, существенных для построения модели. Прежде всего, они могут быть **зашумлёнными** и **неопределёнными**: команда, отправленная агентом, не всегда исполняется точно так, как было задумано. Степень неопределённости варьируется от незначительной (в программных агентах) до весьма высокой (в реальных физических системах, работающих в открытой среде). Помимо этого, каждое действие обладает **стоимостью** (cost) —

ресурсом, который агент тратит на его выполнение (время, энергия, вычислительные мощности, денежные средства), — и **полезностью** (utility) — ожидаемым изменением состояния среды в пользу достижения целей агента. Правильный баланс между стоимостью и полезностью является одним из ключевых аспектов проектирования рациональных агентов.

В контексте многоагентных систем к свойствам действий добавляется возможность **коммуникации**: агент может не только воздействовать на физическую среду, но и передавать информацию другим агентам, координируя совместное поведение.

Важно отметить, что в большинстве задач агентного моделирования, прежде всего в моделях социальных и экономических систем, актуаторы значительно менее сложны, чем механизмы перцепции. Это обусловлено тем, что множество возможных действий, как правило, невелико и хорошо структурировано. Однако в задачах моделирования физических систем (роботов, клеточных ансамблей, биологических тканей) сложность актуаторов может быть весьма высокой и требовать специального внимания при построении модели.

4.4. Связанные свойства среды

Среда — это контекст, в котором функционирует агент: она предоставляет информацию для перцепции и воспринимает воздействия его актуаторов. Понимание свойств среды является необходимым условием правильного проектирования агента: разные комбинации свойств требуют принципиально различных архитектурных решений. В классической систематизации, предложенной Расселом и Норвигом [1], выделяют следующие базовые характеристики рабочей среды агента.

Свойства, связанные с перцепцией:

Наблюдаемость. Среда может быть полностью наблюдаемой (fully observable) — если сенсоры агента в каждый момент времени предоставляют полную информацию о текущем состоянии среды, необходимую для принятия решений. Среда является частично наблюдаемой (partially observable) — если агент имеет доступ лишь к ограниченному «окну» информации: либо из-за физических ограничений сенсоров, либо из-за принципиальной недоступности части состояния. Частичная наблюдаемость резко усложняет задачу планирования, вынуждая агента поддерживать вероятностные модели скрытого состояния среды.

Дискретность/непрерывность. Среда является дискретной, если множество её состояний и возможных воздействий конечно или счётно

(например, шахматная доска, состояние почтового ящика). Непрерывная среда характеризуется бесконечным, непрерывным пространством состояний (например, движение автомобиля в реальном пространстве). На практике непрерывная среда нередко дискретизируется для упрощения моделирования.

Статичность/динамичность. Статическая среда не изменяется, пока агент обдумывает своё следующее действие. Динамическая среда продолжает изменяться независимо от действий агента — время в ней течёт непрерывно. В полностью статических средах в реальном мире практически нет, однако на достаточно высоком уровне абстракции это допущение нередко оказывается приемлемым.

Детерминированность/стохастичность. В детерминированной среде следующее состояние однозначно определяется текущим состоянием и действием агента. В стохастической (недетерминированной) среде присутствует элемент случайности: одно и то же действие может приводить к различным результатам. Большинство реальных систем обладают стохастическими элементами, что обязывает агента учитывать вероятности исходов при планировании.

Свойства, связанные с действиями:

Возможность ошибки. Среда характеризует, насколько выполнение действия агентом гарантировано: в идеализированных средах действие всегда выполняется в точности как задумано, тогда как в реальных системах неизбежны ошибки исполнения.

Оценка **полезности** действий. Каждое действие оценивается с точки зрения того, насколько оно приближает агента к его цели. Формализация этой оценки через функцию полезности (utility function) является основой рационального поведения.

Оценка **стоимости** действий. Параллельно с полезностью каждое действие сопряжено с определёнными затратами ресурсов. Задача агента — найти такой план, при котором разница между полезностью и стоимостью максимальна.

Возможности **коммуникации**. В многоагентных средах агенты способны передавать информацию друг другу. Наличие или отсутствие коммуникации принципиально влияет на возможности координации поведения и коллективного принятия решений.

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Image-analysis	Fully	Deterministic	Episodic	Semi	Continuous	Single
Part-picking robot	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
Refinery controller	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Interactive English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Рисунок 4.3 — Примеры рабочих сред и их характеристики по классификации Рассела—Норвига (Источник: [1])

На рисунке 4.3 приведены примеры конкретных задач и их характеристики по каждому из перечисленных измерений. Так, решение кроссворда представляет собой полностью наблюдаемую, детерминированную, последовательную, статическую, дискретную задачу с одним агентом. Вождение автомобиля-такси, напротив, является частично наблюдаемой, стохастической, последовательной, динамической, непрерывной задачей с несколькими агентами. Медицинская диагностика располагается между этими полюсами и сочетает в себе частичную наблюдаемость, стохастичность и непрерывность временной динамики.

Эти примеры наглядно показывают, что конкретные характеристики среды существенно зависят не только от предметной области, но и от уровня абстракции, с которым мы её рассматриваем. Один и тот же объект реального мира — скажем, пациент с хроническим заболеванием — может трактоваться как статический (при анализе биопсии) или как динамический (при отслеживании прогрессирования болезни); как эпизодический (при каждом отдельном визите к врачу) или как последовательный (при долгосрочном ведении истории болезни).

4.5. Принятие решений агентом

Принятие решений (decision making) — это центральная функция любого агента, связывающая перцепцию (входной канал) с действиями (выходным каналом). По существу, именно здесь «живёт» программа агента. Вопрос о том, как агент принимает решения, является, пожалуй, наиболее фундаментальным во всей теории агентного моделирования.

В теории искусственного интеллекта традиционно противопоставляют два больших класса подходов к разработке агентов [1]: ориентированных на воспроизведение человеческого поведения и ориентированных на рациональность. Это разграничение наиболее чётко сформулировано в знаменитой четырёхчастной классификации Рассела и Норвига, которую называют «четырьмя целями ИИ»: рассуждать как человек, рассуждать рационально, действовать как человек, действовать рационально.

Рассуждение по-человечески (Thinking Humanly) — подход, основанный на когнитивных науках и нейронауках. Целью является воспроизведение процессов мышления человека, включая его ошибки, предубеждения и эвристики. Этот подход важен для создания агентов, имитирующих поведение людей в социальных или экономических системах.

Рациональное рассуждение (Thinking Rationally) — подход, опирающийся на формальную логику, теорию вероятностей и математическую оптимизацию. Агент строит точные модели мира и ищет формально оптимальные решения. Примеры: системы логического вывода, решатели задач удовлетворения ограничений.

Действие по-человечески (Acting Humanly) — подход, при котором внешнее поведение агента неотличимо от поведения человека. Классическим критерием здесь служит тест Тьюринга. Примеры: разговорные агенты, системы анализа и генерации естественного языка.

Рациональное действие (Acting Rationally) — подход, при котором агент совершает действия, наилучшие с точки зрения его целей и имеющейся информации. Это наиболее широко применяемая парадигма в современных системах ИИ, лежащая в основе обучения с подкреплением, планирования и теории решений.

Данная классификация не предполагает жёсткого выбора: на практике агенты нередко комбинируют элементы всех четырёх подходов. Ключевой вопрос при проектировании агента — где на условной шкале от «полностью рационального» до «полностью человекоподобного» находится требуемое поведение системы.

4.6. Рациональность агента

Рациональный агент — это агент, который из доступных ему действий выбирает то, которое максимизирует его ожидаемую производительность согласно имеющейся у него информации о состоянии среды. Ключевое слово здесь — «ожидаемую»: агент максимизирует не реальную производительность (которую он не может

знать заранее), а её ожидаемое значение, исходя из внутреннего представления мира.

Для концептуального описания рационального агента используется аббревиатура **PEAS**: Performance measure (мера производительности), Environment (среда), Actuators (актуаторы), Sensors (сенсоры). **Мера производительности** (Performance measure) определяет критерий успеха агента. Она является внешней по отношению к агенту: задаётся разработчиком или пользователем и отражает реальные цели задачи. Критически важно, что именно она задаёт горизонт рациональности: агент «рационален» строго относительно выбранной меры производительности. Выбор неверной или неполной меры производительности приводит к нежелательным результатам, даже если агент ведёт себя «оптимально» в рамках заданного критерия. **Среда** (Environment) — рабочая среда агента, охарактеризованная по рассмотренным выше измерениям. **Актуаторы** (Actuators) — выходные механизмы агента, через которые он воздействует на среду. **Сенсоры** (Sensors) — входные механизмы агента, через которые он получает информацию о среде.

Принципиально важно понимать, что рациональный агент — это не идеальный агент. Рациональный агент принимает наилучшее возможное решение исходя из имеющейся у него информации — то есть из того внутреннего представления среды, которое сформировано через его сенсоры. Это представление, как мы видели, неполно, зашумлено и запаздывает. Таким образом, рациональность агента всегда ограничена качеством его перцепции.

Ограниченная рациональность. Концепция ограниченной рациональности (bounded rationality), введённая лауреатом Нобелевской премии Гербертом Саймоном [3], фиксирует фундаментальное несоответствие между теоретически оптимальным и практически достижимым поведением. Агент в реальных условиях никогда не располагает полной информацией, неограниченными вычислительными ресурсами и бесконечным временем для принятия решений.

К факторам, ограничивающим рациональность, относятся: неполнота и зашумлённость доступной информации; ограниченность когнитивных или вычислительных ресурсов, не позволяющая рассмотреть все альтернативы; ограниченность времени для принятия решения; наличие социальных и психологических факторов, искажающих логику рассуждений; использование эвристик —

упрощённых правил принятия решений, работающих достаточно хорошо в типичных ситуациях, но не гарантирующих оптимальность.

Следствием ограниченной рациональности является то, что реальные агенты, как правило, принимают не оптимальные, а «достаточно хорошие» (satisficing, по терминологии Саймона) решения. Они «останавливаются» на первом варианте, удовлетворяющем некоторому порогу приемлемости, вместо того чтобы перебирать все возможные альтернативы в поисках глобального оптимума.

В контексте агентного моделирования ограниченная рациональность является ценным инструментом: именно она позволяет воспроизводить реалистичные паттерны человеческого поведения, включая типичные ошибки и систематические отклонения от оптимума. Модели с ограниченной рациональностью применяются в поведенческой экономике, социологии, когнитивной психологии и многих других областях.

Иррациональные агенты. На шкале от полностью рационального до полностью случайного поведения иррациональные агенты занимают промежуточное положение, которое соответствует поведению человека. Иррациональность в данном контексте — это не синоним «ошибочности» или «бессмысленности»; это обозначение систематических отклонений от логически оптимального поведения, обусловленных особенностями когнитивной архитектуры человека.

Часто формы иррационального поведения становятся объектом изучения в поведенческой экономике и психологии. **Когнитивные искажения** (cognitive bias) порождают систематически ошибочные суждения на основе нерелевантных факторов. **Эффект стадного поведения** (herding effect) заставляет людей некритично следовать решениям большинства. **Иррациональный оптимизм** (irrational exuberance) приводит к переоценке активов в условиях пузырей. **Ошибка невозвратных затрат** (sunk cost fallacy) вынуждает принимать решения на основе прошлых невосполнимых инвестиций, а не будущих перспектив. **Предвзятость к настоящему** (present bias) систематически занижает будущие результаты по сравнению с немедленным вознаграждением.

С точки зрения агентного моделирования иррациональные агенты представляют особую ценность именно там, где требуется воспроизвести реалистичное человеческое поведение, а не оптимизировать некоторую функцию полезности. Агенты финансовых рынков, модели потребительского поведения, агенты в социальных

симуляциях — все они требуют встраивания механизмов иррациональности.

Рациональная иррациональность и рациональное невежество. В завершение следует отметить два важных концептуальных понятия, возникающих на стыке ограниченной и иррациональной рациональности. **Рациональная иррациональность** (rational irrationality) — концепция, предложенная экономистом Брайаном Капланом [5], согласно которой поведение, внешне кажущееся иррациональным, может быть вполне рациональным, если учесть субъективную функцию полезности агента. Иными словами, агент следует своим внутренним эвристикам и предубеждениям не вопреки рациональности, а потому что это оптимально с точки зрения его собственных ценностей и убеждений.

Рациональное невежество (rational ignorance) — ситуация, при которой агент сознательно отказывается от получения дополнительной информации, поскольку стоимость её приобретения (временные, когнитивные или финансовые затраты) превышает ожидаемую ценность этой информации для принятия решений. Это явление широко изучается в политической экономике [6] и имеет прямое приложение к моделированию агентов с ограниченным бюджетом на сбор информации.

Оба понятия подчёркивают, что граница между «рациональным» и «иррациональным» поведением агента принципиально зависит от того, относительно какой функции полезности производится оценка. Это соображение имеет важное практическое значение: при проектировании агентных моделей разработчик должен явно специфицировать, какое поведение считается «оптимальным» для решаемой задачи.

Мета-рассуждения и самооценка агента. Особого внимания заслуживает концепция **мета-рассуждений** (meta-reasoning) — способности агента рассуждать о собственных рассуждениях. Это качество считается одной из отличительных черт человеческого когнитивного аппарата, выделяющей его среди других биологических агентов.

Мета-рассуждения предполагают, что агент не только принимает решения на уровне объектных знаний о мире, но и способен оценивать качество своих собственных рассуждений, выявлять ошибки и неопределённости в собственных умозаключениях, корректировать стратегии принятия решений на основе опыта и рефлексии. Применительно к агентному моделированию мета-рассуждения

открывают путь к самоадаптирующимся агентам, которые не просто выполняют фиксированную программу, но и способны улучшать свою собственную логику принятия решений.

В рамках когнитивных архитектур, таких как ACT-R или SOAR, мета-когнитивные механизмы реализуются через специальные модули мониторинга и управления вычислительными ресурсами. Этой теме мы посвятим отдельное внимание в лекции 10.

4.7. Заключение

Перцепция агента — это не просто «считывание» состояния среды, а сложный многоступенчатый процесс, включающий физический сбор сигналов через сенсоры, их обработку с учётом зашумлённости и задержек, а также построение внутреннего образа среды с помощью индуктивных или дедуктивных механизмов. Важнейшей характеристикой перцепции является механизм внимания, связывающий входящий поток информации с целями и когнитивным состоянием агента.

Действия агента реализуются через актуаторы и могут носить как физический, так и информационный характер. Каждое действие обладает стоимостью и полезностью; их соотношение является ключевым параметром при проектировании рационального поведения. В большинстве прикладных моделей действия значительно менее сложны, чем перцепция, хотя в задачах моделирования физических систем это соотношение может меняться.

Рабочая среда агента характеризуется по нескольким измерениям: наблюдаемость, дискретность, статичность, детерминированность, число агентов. Конкретные значения этих характеристик определяются не только предметной областью, но и уровнем абстракции модели и задачами исследования.

Принятие решений агентом может ориентироваться либо на воспроизведение человеческого поведения, либо на достижение формально оптимального результата. Рациональный агент максимизирует ожидаемую производительность согласно имеющейся у него информации — то есть всегда в рамках ограничений перцепции. Ограниченная рациональность фиксирует неизбежный разрыв между идеальной и практически достижимой рациональностью, а иррациональные агенты воспроизводят систематические когнитивные отклонения, характерные для поведения людей в реальных системах.

Понимание механизмов перцепции, действий и принятия решений является необходимой основой для перехода к изучению

многоагентных систем, обучения с подкреплением и реалистичных моделей поведения, которым посвящены последующие главы курса.

Рекомендуемая литература и источники

1. Russell S.J. Artificial Intelligence: A Modern Approach / S.J. Russell, P. Norvig. — 4th ed. — Hoboken : Pearson, 2020. — 1132 p. — ISBN 978-0-13-461099-3. — URL: <https://aima.cs.berkeley.edu/>
2. Kahneman D. Attention and Effort / D. Kahneman. — Englewood Cliffs : Prentice-Hall, 1973. — 246 p. — ISBN 978-0-13-050518-7.
3. Simon H.A. Models of Man: Social and Rational / H.A. Simon. — New York : Wiley, 1957. — 287 p.
4. Lashkari Y. Collaborative Interface Agents / Y. Lashkari, M. Metral, P. Maes // Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94). — Menlo Park : AAAI Press, 1994. — P. 444–449. — DOI: 10.1016/B978-0-08-051574-8.50084-4. — URL: <https://doi.org/10.1016/B978-0-08-051574-8.50084-4>
5. Caplan B. Rational Ignorance versus Rational Irrationality / B. Caplan // Kyklos. — 2001. — Vol. 54, No. 1. — P. 3–26. — DOI: 10.1111/1467-6435.00143. — URL: <https://doi.org/10.1111/1467-6435.00143>
6. Downs A. An Economic Theory of Democracy / A. Downs. — New York : Harper, 1957. — 310 p.
7. Bandura A. Social Learning Theory / A. Bandura. — Englewood Cliffs : Prentice-Hall, 1977. — 247 p. — ISBN 978-0-13-816748-1.
8. Wooldridge M. An Introduction to MultiAgent Systems / M. Wooldridge. — 2nd ed. — Chichester : Wiley, 2009. — 484 p. — ISBN 978-0-470-51946-2.
9. Nye B.D. Modeling Memes: A Memetic View of Affordance Learning / B.D. Nye. — 2011.

Лекция 5. Многоагентные системы

Многоагентные системы (Multi-Agent Systems, MAS) представляют собой одно из наиболее динамично развивающихся направлений в области вычислительного интеллекта и моделирования сложных систем. В отличие от традиционного монолитного подхода к моделированию, в котором система описывается как единое целое с помощью дифференциальных уравнений или агрегированных статистических моделей, многоагентный подход строится «снизу вверх»: поведение системы в целом возникает из взаимодействия множества автономных сущностей — агентов. Именно эта особенность делает МАС незаменимым инструментом для изучения систем, в которых коллективное поведение нельзя свести к простой сумме действий отдельных компонентов. Настоящая глава посвящена ключевым принципам устройства и функционирования многоагентных систем: их основным свойствам, механизмам коммуникации агентов, способам организации среды, типологии взаимодействий, а также практическим аспектам имитационного моделирования — верификации, валидации и реализации в современных инструментальных средствах. В заключение рассмотрены три классических примера, демонстрирующих богатство эффектов, возникающих в МАС: клеточный автомат «Жизнь», модели комплексных сетей и модель Изинга в применении к социальным системам.

5.1. Понятие и ключевые свойства многоагентных систем

Многоагентная система (МАС) — это вычислительная система, в которой одновременно функционируют несколько взаимодействующих автономных агентов, каждый из которых обладает собственным состоянием, восприятием среды и набором правил поведения. При этом «многоагентность» подразумевает не просто наличие нескольких агентов, но и системообразующее взаимодействие между ними, порождающее свойства, не присущие ни одному из агентов в отдельности.

С точки зрения состава популяции агентов МАС могут быть **гомогенными** (все агенты одинаковы по архитектуре и правилам поведения) или **гетерогенными** (агенты различаются по типу, роли или внутренней организации). Гомогенные системы значительно проще для теоретического анализа и верификации, тогда как гетерогенные позволяют воспроизвести более реалистичное разнообразие реальных систем — от финансовых рынков до биологических популяций.

Принципиально важным свойством МАС является наличие нескольких масштабов описания: **микроуровень** — уровень отдельного агента: его состояние, правила поведения, восприятие окружения и выполняемые действия; **макроуровень** — уровень системы в целом: агрегированные характеристики, статистические распределения, глобальные паттерны; **мезоуровень** — промежуточный уровень, соответствующий группам агентов, субпопуляциям или устойчивым паттернам взаимодействия. В ряде подходов именно на мезоуровне фиксируются эмерджентные структуры, не планировавшиеся при задании правил на микроуровне.

Определяющей особенностью многоагентного подхода является **асимметрия между уровнями описания**: правила задаются на микроуровне, а системные характеристики наблюдаются на макроуровне. Именно из этого разрыва возникают явления эмерджентности, которые будут подробнее рассмотрены в следующей главе.

По характеру координации поведение агентов в МАС может управляться централизованно — с помощью единого управляющего агента или координатора — либо децентрализованно, когда каждый агент принимает решения самостоятельно, опираясь лишь на локальную информацию. В последнем случае говорят о самоорганизующейся системе. Варианты взаимодействия между агентами охватывают широкий спектр: от полного игнорирования (агенты взаимодействуют только через среду) до активной кооперации (совместное решение общей задачи) и конкуренции (борьба за ресурсы или пространство).

5.2. Коммуникация агентов

Возможность прямого обмена информацией между агентами — одно из ключевых отличий МАС от более простых многокомпонентных моделей. Коммуникация позволяет агентам координировать действия, распределять задачи и достигать общих целей, недостижимых в одиночку.

Язык и протокол взаимодействия. Для того чтобы два агента могли обмениваться сообщениями, необходимо определить общий язык коммуникации. В формальном смысле язык агентов включает следующие компоненты. **Алфавит и словарь** — множество допустимых символов и терминов, из которых строятся сообщения. **Синтаксис** — правила построения допустимых выражений из элементов словаря. **Семантика** — соответствие между выражениями языка и их смыслом;

определяет, как агент должен интерпретировать полученное сообщение. **Онтология** — формальное описание понятий и отношений предметной области. Онтология обеспечивает единое понимание терминов и является обязательным компонентом в гетерогенных системах, где агенты могут иметь различные внутренние модели мира.

Наиболее распространённым стандартом для коммуникации программных агентов является FIPA ACL (Agent Communication Language), разработанный организацией Foundation for Intelligent Physical Agents. В нём сообщения структурируются как «речевые акты» (speech acts), отражающие намерения отправителя (INFORM, REQUEST, PROPOSE, AGREE, REFUSE).

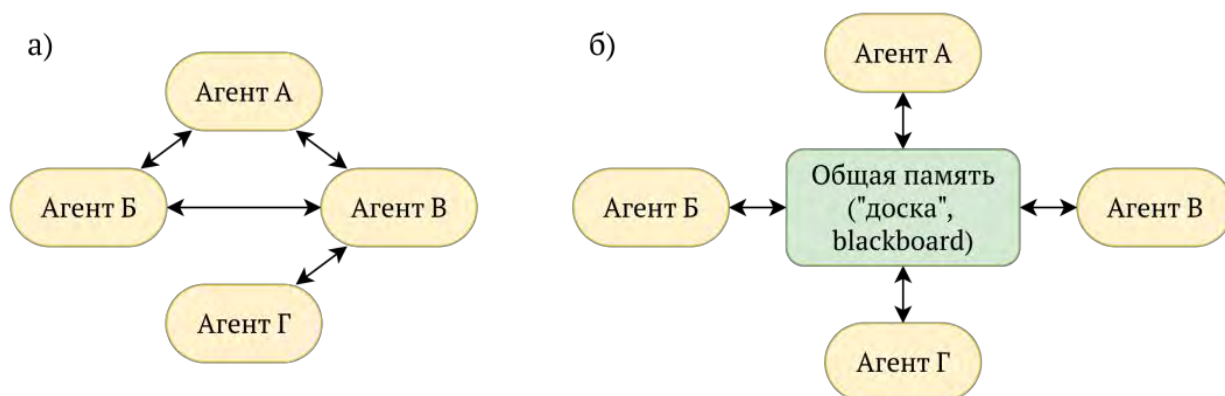


Рисунок 5.1 — Модели коммуникации агентов: а) индивидуальная; б) через общую доску объявлений

Типы коммуникации. По топологии и механизму передачи информации принято выделять три основных паттерна коммуникации (см. рисунок 5.1). **Массовая рассылка (broadcast).** Агент транслирует сообщение всем участникам системы одновременно. Паттерн прост в реализации, но порождает большую нагрузку на систему при увеличении числа агентов. Применяется в задачах оповещения и синхронизации. **Индивидуальная коммуникация (один-к-одному, peer-to-peer).** Агент направляет сообщение конкретному адресату. Это наиболее гибкий и распространённый вариант: реальная топология взаимодействия определяется структурой сети, в которой каждое ребро соответствует потенциальной коммуникационной связи. **Общая память (Blackboard).** Агенты не обмениваются сообщениями напрямую, а читают из общего структурированного хранилища и записывают в него. Это обеспечивает косвенную асинхронную координацию: агент публикует информацию, другой — считывает её в удобный момент. Паттерн широко применяется в экспертных системах и задачах планирования. Выбор паттерна коммуникации во многом

определяется характером задачи и вычислительными ограничениями. В реальных системах нередко сочетаются несколько паттернов: например, broadcast используется для глобального оповещения, тогда как детали переговоров проводятся через индивидуальные каналы.

5.3. Структурирование агентов: группы, роли, организация, среда

По мере роста числа агентов прямое описание всех попарных взаимодействий становится затруднительным. Для иерархической организации МАС применяется понятие группы. Группа — это подмножество агентов, связанных общей задачей, общим пространством взаимодействия или иным структурным критерием. Группы могут быть статическими (состав фиксирован на всё время моделирования) или динамическими (агенты вступают в группы и покидают их в процессе симуляции; сами группы могут создаваться и распадаться). Введение групп порождает иерархию: группа сама может выступать в роли агента более высокого уровня, а коммуникация между группами строится по тем же принципам, что и коммуникация между отдельными агентами.

Наиболее проработанной формальной моделью организации МАС является формализм «Агент–Группа–Роль» (AGR, Agent–Group–Role), предложенный Ferber и Gutknecht [2]. В нём каждый агент в каждый момент времени принадлежит одной или нескольким группам и исполняет в них определённые роли. Роль определяет набор прав, обязанностей и коммуникационных протоколов, доступных агенту, не задавая при этом его внутреннее устройство. Это позволяет строить организационные структуры, аналогичные корпоративной иерархии, с делегированием полномочий и разделением труда.

Другим важным аспектом структурирования является понятие ситуативного агента (situated agent) — агента, явно помещённого в физическое или абстрактное пространство, которое ограничивает его восприятие и взаимодействие. Ситуативность обеспечивает локальность: агент воспринимает лишь ближайшее окружение и взаимодействует лишь с соседями, что и порождает характерные для МАС эффекты самоорганизации.

По характеру управляющих правил поведения агентов в МАС можно классифицировать в соответствии с архитектурами агентов (см. лекцию 3). Часто, помимо реактивных и рассуждающих агентов, в отдельный класс выделяют **адаптивных агентов**, правила поведения которых могут изменяться в ходе моделирования. Это обучающиеся агенты, способные корректировать свою стратегию на основе

накопленного опыта. В строгих МАС с фиксированными правилами (как в «Жизни» или модели Изинга) адаптация отсутствует, тогда как в гибридных архитектурах она является ключевым механизмом.

Среда в МАС в целом соответствует общим понятиям среды агентного моделирования (см. лекцию 4).

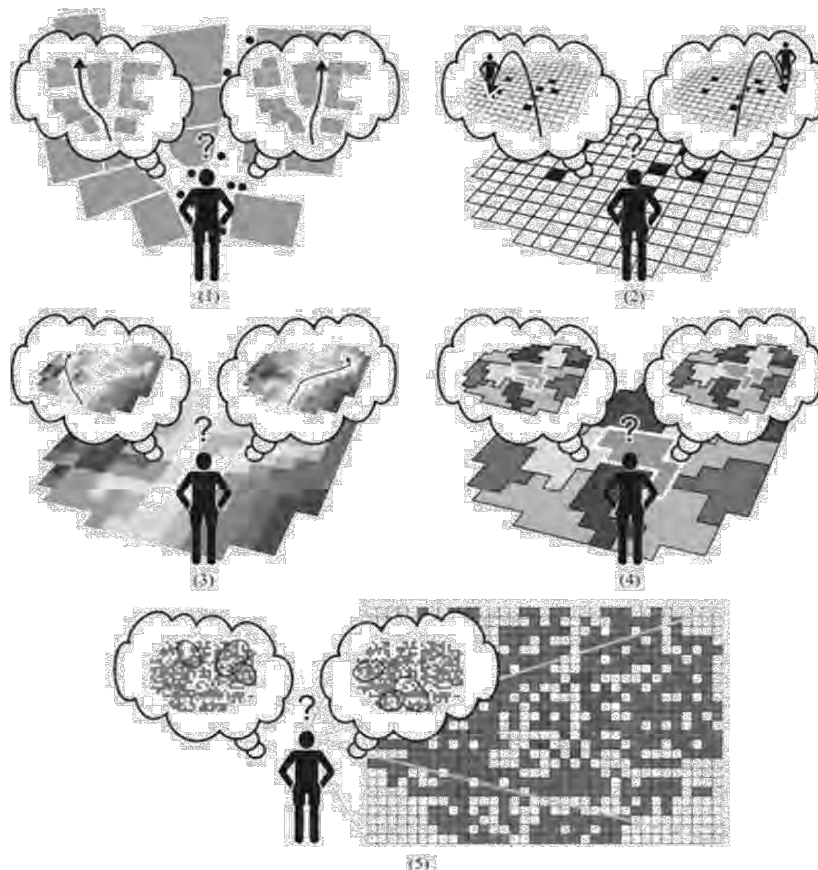


Рисунок 5.2 — Пять типов взаимодействия агент–среда в геоинформационном контексте (Источник: [10])

Тип взаимодействия агента со средой во многом определяет класс моделируемых явлений. Например, в контексте пространственных систем можно [10] выделять пять основных моделей взаимодействия агент–среда (Рисунок 5.2). (1) **Мобильность**. Ключевым действием агента является перемещение. Агент выбирает маршрут среди возможных траекторий, решает задачи навигации и диспетчеризации. Пространство может быть представлено дорожным графом или непрерывным полем. Типичные приложения — моделирование пешеходного трафика, автономных транспортных средств, эвакуации. (2) **Позиционирование**. Агент выбирает позицию в среде согласно определённым критериям (предпочтения, ресурсы, близость к другим агентам). Перемещение как таковое может не моделироваться — агент «телепортируется» из одной позиции в другую. Классическим примером

является модель сегрегации Шеллинга, демонстрирующая возникновение расовой сегрегации из сравнительно умеренных индивидуальных предпочтений. (3) **«Охотник-собиратель» (forager)**. Агент не только перемещается по среде, но и изменяет её ресурсы: потребляет, накапливает, истощает. Среда имеет состояние в каждой точке (ресурс, концентрация вещества и т.п.), которое обновляется под воздействием агентов. Модели муравейников и «сахарного ландшафта» (Sugarscape) относятся к этому классу. (4) **Принадлежность территории (land use)**. Агент структурирует среду, выделяя регионы и устанавливая отношения принадлежности или использования. Фермер, размечающий поля, городской планировщик, распределяющий функциональные зоны, кластеризационный агент в задаче сегментации — все они реализуют этот паттерн. Ключевым объектом является не траектория агента, а граница между регионами среды. (5) **Генерация среды (development)**. Агент создаёт новые объекты в среде, тем самым изменяя её топологию. Застройщик, строящий здание, клетка, делящаяся и порождающая дочерние клетки, нейрон, формирующий новые синаптические связи, — всё это примеры данного типа взаимодействия. Среда в этом случае является динамической и может существенно меняться в ходе симуляции.

В реальных прикладных моделях эти паттерны нередко сочетаются. Например, агент-горожанин может одновременно перемещаться (мобильность), выбирать место жительства (позиционирование) и оказывать влияние на рынок недвижимости (генерация среды).

5.4. Механизмы координации

Одним из центральных элементов архитектуры МАС является механизм координации — совокупность методов, обеспечивающих согласованность действий агентов при достижении как индивидуальных, так и системных целей (Рисунок 5.3).

Выделяют несколько уровней и типов координации. **Организационное структурирование (Organizational Structuring)**. Заранее определённая иерархия ролей и групп задаёт, кто кому подчинён и каковы каналы коммуникации. Это наиболее «жёсткий» вид координации — аналог корпоративной структуры. **Обмен метаинформацией (Meta-level Information Exchange)**. Агенты не только обмениваются данными о состоянии среды, но и информируют друг друга о своих намерениях, возможностях и планах. Это позволяет предотвращать конфликты и дублирование усилий. **Многоагентное**

планирование (Multiagent Planning). Агенты совместно строят план действий, согласовывая свои подзадачи так, чтобы результирующий план был глобально допустимым и эффективным. **Контрактная сеть (Contract Net Protocol, CNP).** Широко используемый протокол распределения задач: агент-менеджер объявляет тендер (call for proposals), агенты-исполнители подают заявки (proposals), менеджер выбирает лучшую и заключает контракт. Протокол эффективно распределяет задачи в динамической среде. **Переговоры (Negotiation).** Агенты с потенциально конфликтующими интересами достигают соглашения через итеративный обмен предложениями. Модели переговоров заимствуются из теории игр и экономической теории. **Рыночные механизмы (Market Mechanisms).** Координация через ценовые сигналы: агенты торгуются за ресурсы, а равновесная цена регулирует их распределение. Этот механизм является центральным в агентных моделях финансовых рынков.



Рисунок 5.3 — Архитектура МАС: агенты, их свойства и механизмы координации (адаптировано по [9])

5.5. Имитационное моделирование многоагентных систем

Построение МАС — лишь первая часть работы. Вторая, не менее важная, — запуск модели, то есть имитационное моделирование (simulation). В англоязычной литературе принято различать modeling (построение описания системы) и simulation (воспроизведение её динамики во времени). Это различие подчёркивает, что одна и та же модель может использоваться для генерации множества различных симуляций при разных начальных условиях или параметрах.

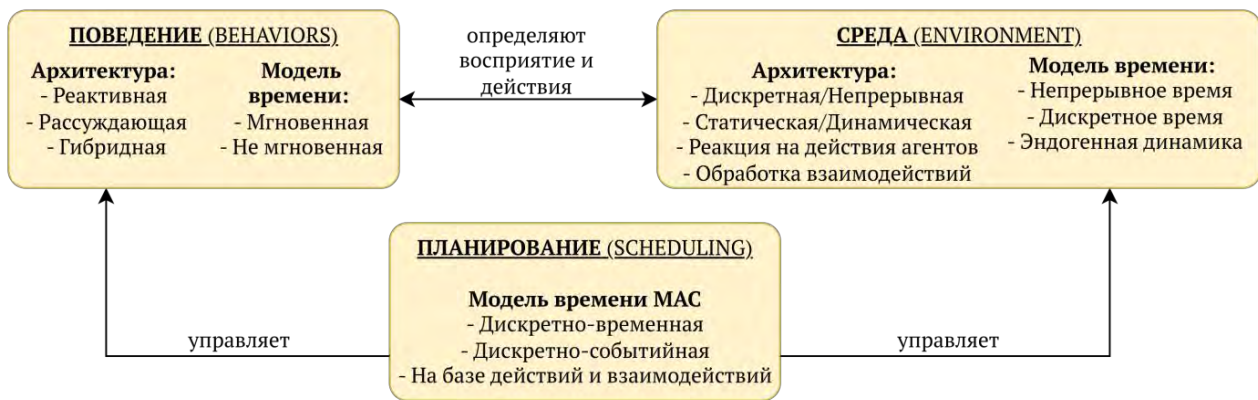


Рисунок 5.4 — Архитектура имитационного моделирования МАС: поведение агентов, среда и планировщик (адаптировано по [11])

Компоненты имитационной системы. Любая система имитационного моделирования МАС включает три взаимосвязанных компонента (Рисунок 5.4).

Поведение агентов (Behaviors). Определяет архитектуру агентов (реактивная, когнитивная, гибридная), их внутренние состояния и правила перехода. Модель времени агента может быть «мгновенной» (агент обновляет состояние мгновенно) или «не мгновенной» (действие занимает некоторое модельное время).

Среда (Environment). Описывает пространство и топологию, статические и динамические свойства среды, способ реакции среды на действия агентов и механизмы обработки взаимодействий между агентами.

Планировщик (Scheduler). Управляет продвижением модельного времени, активирует агентов, упорядочивает их действия и обеспечивает целостность состояния системы после каждого шага.

Модели времени. Выбор модели времени является критически важным решением при проектировании МАС:

Дискретное время (time-stepped). Модельное время делится на равные шаги (ticks). На каждом шаге планировщик последовательно (или параллельно) активирует всех агентов, каждый из которых обновляет своё состояние. Подход прост в реализации и отладке, но может быть вычислительно расточительным в случаях, когда между событиями проходит много «пустых» шагов.

Дискретно-событийное моделирование (Discrete Event Simulation, DES). Модельное время продвигается не равномерно, а от события к событию. Каждый агент, совершая действие, генерирует событие в будущем (например: «через 1 минуту 20 секунд агент переместится в точку В»). Эти события помещаются в приоритетную

очередь, и планировщик переходит непосредственно к следующему событию, пропуская «пустое» время. Дискретно-событийный подход значительно эффективнее с вычислительной точки зрения и хорошо подходит для систем с неравномерной активностью агентов.

Следует особо отметить, что порядок активации агентов внутри одного шага влияет на результаты симуляции. Три основных подхода: синхронная активация (все агенты обновляются одновременно), асинхронная последовательная (агенты активируются по одному в заданном порядке) и случайная активация (порядок случаен на каждом шаге). Выбор подхода должен соответствовать природе моделируемой системы.

Помимо управления временем, планировщик отвечает за **жизненный цикл агентов**: их создание (рождение), активную фазу и удаление (смерть). В биологических и демографических моделях появление и исчезновение агентов является неотъемлемой частью динамики. В информационных системах агент может «войти» в систему (например, пользователь подключился к сети) и «выйти» из неё. Корректная обработка этих событий — уничтожение ссылок, обновление топологии связей — является важной реализационной задачей.

Разработка многоагентной модели — итеративный процесс, включающий не только конструирование, но и проверку качества. Здесь принципиально важно различать два понятия.

Верификация (Verification) отвечает на вопрос: «Правильно ли реализована модель?» — то есть соответствует ли поведение программы задуманному дизайну. Верификация обнаруживает ошибки в коде, алгоритмах и спецификациях.

Валидация (Validation) отвечает на вопрос: «Ту ли модель мы построили?» — то есть насколько адекватно модель воспроизводит поведение реальной системы. Валидация требует сравнения с эмпирическими данными.

Стандартный конвейер проверки качества модели включает четыре последовательных этапа (Рисунок 5.5). 1) **Лицевая валидность (Face Validation)**. Разработчик визуально наблюдает за поведением агентов в симуляции и оценивает, выглядит ли оно качественно правдоподобно. Этот шаг относится в большей степени к верификации: он позволяет обнаружить грубые ошибки ещё до сбора статистики. 2) **Анализ чувствительности (Sensitivity Analysis)**. Исследуется, как меняются выходные характеристики модели при вариации входных параметров. Цели двойные: убедиться, что модель не «разваливается»

при малых изменениях параметров (устойчивость), и выявить наиболее влиятельные параметры для последующей калибровки. 3) **Калибровка (Calibration)**. Параметры модели настраиваются так, чтобы выходные данные соответствовали реальным наблюдениям. В случае МАС это особенно сложно: параметры задаются на микроуровне (правила агентов), а данные для сравнения имеются на макроуровне (агрегированные статистики). 4) **Выходная валидация (Output Validation)**. Количественное сравнение предсказаний модели с независимыми эмпирическими данными. Только на этом шаге можно говорить о полностью валидированной модели.



Рисунок 5.5 — Конвейер верификации и валидации многоагентной модели: четыре этапа от лицевой валидности до полной выходной валидации (адаптировано по [12])

Принципиальная трудность валидации МАС состоит в «**микро-макро-разрыве**»: нет формально гарантированного алгоритма перехода от корректных микроправил к нужному макроповедению. Исследователь вынужден проходить весь цикл несколько раз, корректируя правила агентов до достижения приемлемого соответствия между симуляцией и реальностью. Это существенно отличает МАС от традиционных подходов, где параметры модели можно подобрать аналитически или с помощью стандартных методов оптимизации.

Важно также разграничивать ситуацию, когда среда и агенты задаются разработчиком (designed), и ситуацию, когда они анализируются по реальным данным (analysed). В первом случае верификация опирается на теоретическое сравнение и репликацию; во втором — на количественные критерии соответствия данным, что требует иных инструментов разработки.

Более подробно общие принципы моделирования также будут рассмотрены в лекции 11.

5.6. Кросс-дисциплинарность многоагентного моделирования

Многоагентное моделирование является подчёркнуто кросс-дисциплинарной областью. В этом состоит одновременно и её сила (широта применений, богатство методов), и слабость (отсутствие единой теоретической базы, сложность переноса результатов между дисциплинами). На формирование теории и практики МАС оказали влияние следующие направления. **Объектно-ориентированное программирование** предоставило концептуальный каркас для описания агентов как объектов с инкапсулированным состоянием и методами. **Распределённые системы и распределённый искусственный интеллект** сформировали подходы к декомпозиции задач, протоколы коммуникации (FIPA ACL, KQML) и механизмы координации. **Экономика и теория игр** снабдили МАС формальными моделями рационального поведения, аукционных и рыночных механизмов, переговорных протоколов. **Психология и когнитивные науки** оказали влияние на проектирование архитектур агентов с убеждениями, желаниями и намерениями (BDI-архитектура). **Социология** предоставила концепции социальных сетей, нормативных систем, коллективного поведения толп. **Биология и физика** дали начало классу «социофизических» моделей (модели эпидемий, стайного поведения, самоорганизации), где правила поведения агентов напрямую транслируются из физических или биологических принципов. Именно это многообразие источников объясняет, почему различные инструментальные средства для МАС сильно отличаются по акцентам: одни ориентированы на программных агентов со сложной архитектурой, другие — на пространственное моделирование, третьи — на сетевые системы.

5.7. Инструментальные средства многоагентного моделирования

Для реализации МАС существует широкий спектр программных инструментов — от специализированных GUI-платформ до

низкоуровневых библиотек. Полный обзор доступен на странице Wikipedia: «Comparison of agent-based modeling software» (https://en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software). В таблице представлены наиболее часто используемые в академической практике платформы.

Таблица 5.1 - Примеры платформ для создания МАС

Платформа	Язык	Основные возможности	Применение
NetLogo http://www.netlogo.org/	NetLogo DSL	Клеточные автоматы, встроенная визуализация, обширная библиотека моделей	Обучение, исследования
Repast Symphony https://repast.github.io/	Java/Groovy	Высокая гибкость, GIS, статистические модули, HPC	Научные исследования
AnyLogic https://www.anylogic.com/	Java	GUI, агентные + SD + DES модели, GIS, коммерческая	Бизнес, промышленность
GAMA http://gama-platform.org/	GAML DSL	Глубокая интеграция GIS, 3D-среды	Пространственные симуляции
Mesa https://mesa.readthedocs.io/stable/	Python	Лёгкость интеграции с ML/данными, веб-визуализация	Исследования, ML-интеграция

NetLogo, разработанный в Northwestern University, остаётся наиболее популярным инструментом для учебных целей благодаря низкому порогу входа и обширной библиотеке готовых моделей. Repast Symphony обеспечивает промышленный уровень производительности и масштабируемости, поддерживая распределённые вычисления на суперкомпьютерах. AnyLogic привлекает бизнес-пользователей благодаря графическому интерфейсу и возможности комбинировать агентное моделирование с системной динамикой и дискретно-событийным моделированием в одном проекте.

5.8. Классические примеры многоагентных моделей

Игра «Жизнь». Клеточный автомат «Жизнь» (Conway's Game of Life), предложенный математиком Джоном Хортоном Конвеем в 1970 году, является, пожалуй, самой известной демонстрацией эмерджентного поведения в МАС. Система определяется тремя правилами на бесконечной двумерной сетке. **Рождение:** в пустой

(мёртвой) клетке, рядом с которой ровно три живые соседки, зарождается жизнь. **Выживание:** живая клетка с двумя или тремя живыми соседями продолжает жить. **Смерть:** живая клетка с числом соседей менее двух («одиночество») или более трёх («перенаселённость») умирает.

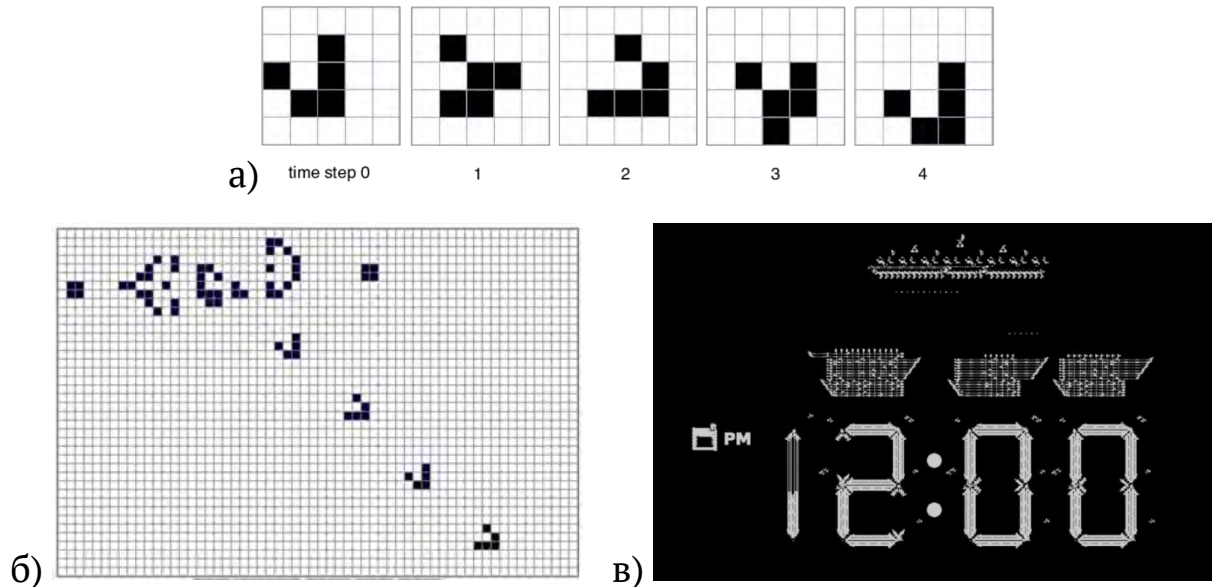


Рисунок 5.6 — Игра «Жизнь»: а) конфигурация «glider» (4 временных шага); б) пример симуляции более сложной конфигурации «gosper gun» (доступ онлайн: <http://www.michurin.net/online-tools/life-game.html>); в) симуляция «часы» (запись онлайн: <https://www.youtube.com/watch?v=3NDAZ5g4EuU>)

Из этих трёх простых локальных правил возникает удивительное разнообразие поведения (рисунок 5.6). Простейший нетривиальный паттерн — «глайдер» (glider): конфигурация из пяти клеток, которая за четыре шага воспроизводит себя, смещаясь на одну клетку по диагонали. Глайдер демонстрирует стабильность (паттерн сохраняется), движение (он перемещается по полю) и несводимость к правилам (ни одно правило не упоминает «движение»). Это классический пример эмерджентного поведения.

Более сложные конструкции — «пушка глайдеров» (glider gun), периодически генерирующая глайдеры, и «пожиратели» (eaters), уничтожающие их, — позволяют строить схемы, вычисляющие логические функции. В пределе исследователи построили в «Жизни» полноценный универсальный компьютер с памятью, вычислителем и вводом-выводом. Это свидетельствует о том, что «Жизнь» является вычислительно универсальной — то есть, в принципе, способна воспроизвести любой алгоритм.

С точки зрения МАС «Жизнь» — предельно простая гомогенная система в клеточной среде с реактивными агентами (клетками), состояние которых бинарно (жива/мертва). Тем не менее системные паттерны чрезвычайно богаты. Этот пример наглядно иллюстрирует принцип, лежащий в основе всего многоагентного подхода: сложное макроповедение порождается простыми микроправилами.

Комплексные сети. Многие реальные сети — интернет, сети цитирования, социальные сети, сети соавторства — демонстрируют характерное свойство: распределение степеней вершин подчиняется степенному закону (рисунок 5.7).

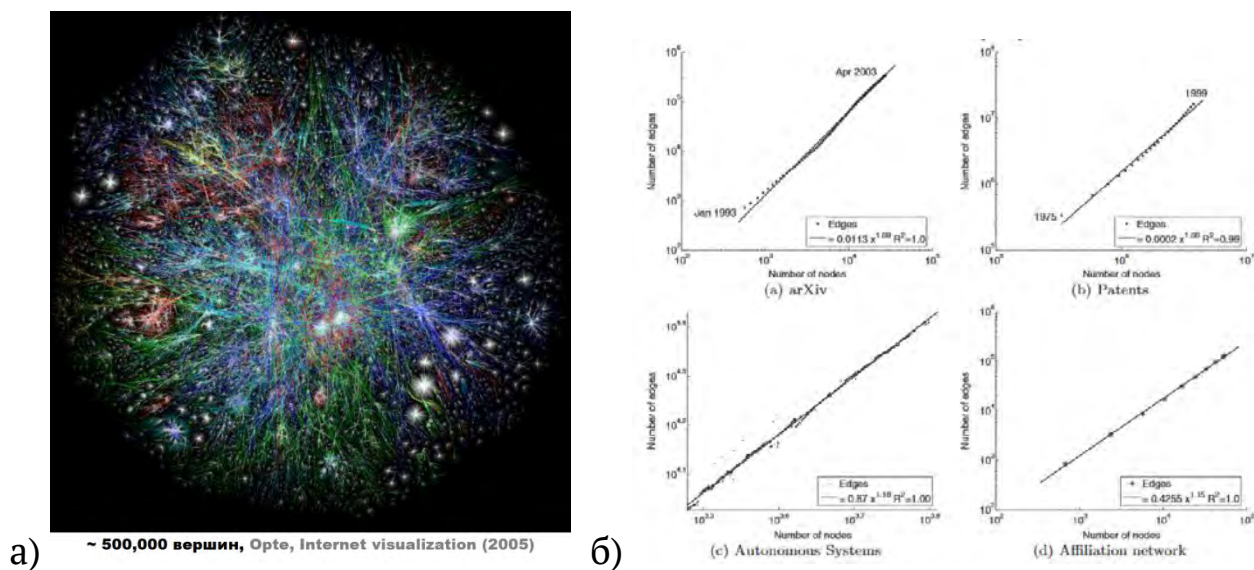


Рисунок 5.7 — Безразмерные сети: а) визуализация интернета (Источник: проект Opte - <https://www.opte.org/>); б) пример распределения степеней вершин для различных социальных сетей (Источник [13])

Иначе говоря, если обозначить через $P(k)$ долю вершин со степенью k , то $P(k) \sim k^{-\gamma}$, где γ — характеристический показатель (наиболее интересны сети с $2 < \gamma < 3$). В двойной логарифмической шкале это выглядит как прямая линия. Такие сети называют безразмерными (scale-free): у них нет характерного масштаба — в отличие, например, от случайных сетей Эрдёша–Рени с пуассоновским распределением степеней. Альберт-Ласло Барабаши и Реа Альберт в 1999 году предложили механизм, объясняющий возникновение безразмерности: **предпочтительная привязанность (preferential attachment)**. Суть состоит в следующем: сеть растёт, на каждом шаге добавляется новый агент (вершина), который присоединяется к уже существующей вершине с вероятностью, пропорциональной её степени: $p_i = k_i / \sum_j k_j$.

Здесь p_i — вероятность присоединения к вершине i , k_i — её текущая степень. Таким образом, «богатые становятся богаче»: высокосвязные вершины притягивают ещё больше связей, порождая неравномерное («степенное») распределение. Принципиально, что это не статическая структура, а динамический агентный процесс: ключевым является именно последовательное появление и присоединение новых агентов.

Другой важной характеристикой сетей является кластеризация — склонность соседей одной вершины быть связанными между собой (образовывать «треугольники»). Высокая кластеризация при малом среднем расстоянии между вершинами характерна для сетей «малого мира» (small-world networks) Уоттса–Строгаца. Реальные сети нередко совмещают свойства безразмерности и тесного мира. Агентный подход к построению комплексных сетей позволяет имитировать их рост и эволюцию, что открывает путь к изучению распространения информации, влияния, болезней и финансовых рисков в сетевых структурах.

Модель Изинга, первоначально предложенная в 1925 году Эрнстом Изингом для описания взаимодействия магнитных диполей в кристаллической решётке, стала классическим примером «социальной физики» — направления, переносящего физические модели на социальные явления. В исходной физической постановке каждый узел решётки имеет спин $\sigma_i \in \{-1, +1\}$; взаимодействие соседей определяется гамильтонианом системы, минимум которого соответствует упорядоченному состоянию (все спины сонаправлены).

В социальной интерпретации спин агента соответствует его мнению (например, за кандидата А или кандидата В на выборах). Вероятность изменения мнения агента i определяется двумя следующими конкурирующими механизмами. **Поддержка (supportiveness)** s_{ij} — тенденция агента сохранить своё мнение под влиянием единомышленников в окружении. Вклад этого компонента активен, когда мнения агента i и соседа j совпадают ($\sigma_i = \sigma_j$). **Убеждение (persuasiveness)** p_{ij} — склонность агента сменить мнение под давлением несогласных соседей. Активен при $\sigma_i \neq \sigma_j$.

Вероятность переключения мнения агента i на шаге t описывается формулой: $I_i = [\sum_j p_{ij}/d_{ij}^\alpha \cdot (1 - \sigma_i \sigma_j)] - [\sum_j s_{ij}/d_{ij}^\alpha \cdot (1 + \sigma_i \sigma_j)]$, где d_{ij} — расстояние между агентами i и j (в пространственной версии), а α — параметр, управляющий скоростью убывания влияния с расстоянием. Расширения базовой модели включают фазы формирования мнения (нерешительность $\rightarrow$ убеждённость), перемещение агентов в

пространстве (смена соседей), а также влияние «внешнего поля» (медиа, лидер мнений). Совокупность этих механизмов позволяет моделировать реалистичные сценарии предвыборных кампаний, включая эффект «последнего слова» (bandwagon effect) и поляризацию электората.

Модель Изинга наглядно демонстрирует, как физический формализм, описывающий магнетизм, может быть переосмыслен в терминах социальных взаимодействий — и при этом давать количественно проверяемые предсказания о коллективной динамике мнений. Это один из наиболее убедительных аргументов в пользу универсальности подхода МАС.

5.9. Заключение

Многоагентные системы представляют собой мощный инструмент для изучения широкого класса явлений — от физических и биологических до экономических и социальных. Ключевая особенность МАС состоит в принципиальной асимметрии между уровнями описания: правила задаются на микроуровне (отдельный агент), а интерес представляют свойства макроуровня (система в целом). Именно из этого разрыва рождаются эмерджентные паттерны, не заданные явно ни в одном из правил.

Фундаментальными элементами любой МАС являются агенты (с их архитектурой, состоянием и правилами поведения), среда (пространственная, сетевая или абстрактная), топология взаимодействия (определяющая, кто с кем может взаимодействовать) и механизм координации (регулирующий коллективное поведение). Коммуникация между агентами — через языки, протоколы и различные паттерны (broadcast, peer-to-peer, blackboard) — является важнейшим инструментом достижения согласованных коллективных действий.

Практическая реализация МАС требует решения ряда технических вопросов: выбора модели времени (дискретное vs. событийное), порядка активации агентов, представления среды и механизмов проверки корректности модели. Верификация и валидация — наиболее сложные этапы работы с МАС, обусловленные принципиальной трудностью аналитического перехода от микроправил к макронаблюдениям.

Три рассмотренных примера — «Жизнь», комплексные сети и модель Изинга — при всём своём различии объединены общей идеей: простые локальные взаимодействия агентов порождают нетривиальное и содержательное коллективное поведение. Именно эта способность

выявлять системные эффекты «снизу вверх» делает многоагентный подход незаменимым в современной науке о сложных системах.

Рекомендуемая литература и источники

1. Wooldridge M. An Introduction to MultiAgent Systems / M. Wooldridge. — 2nd ed. — Chichester : Wiley, 2009. — 484 p. — ISBN 978-0-470-51946-2.
2. Ferber J. A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems / J. Ferber, O. Gutknecht // Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98). — Paris : IEEE Computer Society, 1998. — P. 128–135.
3. Macal C.M. Tutorial on Agent-Based Modelling and Simulation / C.M. Macal, M.J. North // Journal of Simulation. — 2010. — Vol. 4, No. 3. — P. 151–162. — DOI: 10.1057/jos.2010.3. — URL: <https://doi.org/10.1057/jos.2010.3>
4. Barabási A.-L. Emergence of Scaling in Random Networks / A.-L. Barabási, R. Albert // Science. — 1999. — Vol. 286, No. 5439. — P. 509–512. — DOI: 10.1126/science.286.5439.509. — URL: <https://www.science.org/doi/10.1126/science.286.5439.509>
5. Galam S. Sociophysics: A Physicist's Modeling of Psycho-political Phenomena / S. Galam. — New York : Springer, 2012. — 439 p. — ISBN 978-1-4614-2031-6. — DOI: 10.1007/978-1-4614-2032-3. — URL: <https://doi.org/10.1007/978-1-4614-2032-3>
6. Bandini S. Agent Based Modeling and Simulation: An Informatics Perspective / S. Bandini, S. Manzoni, G. Vizzari // Journal of Artificial Societies and Social Simulation. — 2009. — Vol. 12, No. 4. — Art. 4. — URL: <http://jasss.soc.surrey.ac.uk/12/4/4.html>
7. Agent-Based Models of Geographical Systems / ed. by A.J. Heppenstall [et al.]. — Dordrecht : Springer, 2012. — 762 p. — ISBN 978-90-481-8926-7. — DOI: 10.1007/978-90-481-8927-4. — URL: <https://doi.org/10.1007/978-90-481-8927-4>
8. Oliya S. Agent-Based Model for Voter Opinion Dynamics / S. Oliya [et al.] // Lecture Notes in Computer Science. — Cham : Springer, 2020. — DOI: 10.1007/978-3-030-30244-3_51. — URL: https://doi.org/10.1007/978-3-030-30244-3_51
9. Kishore R. Enterprise Integration Using the Agent Paradigm: Foundations of Multi-Agent-Based Integrative Business Information Systems / R. Kishore [et al.] // Decision Support Systems. — 2006. — Vol. 42, No. 1. — P. 48–78. — DOI: 10.1016/j.dss.2004.09.011. — URL: <https://doi.org/10.1016/j.dss.2004.09.011>
10. O'Sullivan D. Agent-Based Models – Because They're Worth It? / D. O'Sullivan [et al.] // Agent-Based Models of Geographical Systems / ed. by A. Heppenstall [et al.]. — Dordrecht : Springer, 2012. — P. 109–123. — DOI: 10.1007/978-90-481-8927-4_6. — URL: https://doi.org/10.1007/978-90-481-8927-4_6
11. Drogoul A. Multi-Agent Systems and Simulation: A Survey from the Agent Community's Perspective / A. Drogoul, F. Michel, J. Ferber // Multi-Agent Systems. — 2009.
12. Ngo T.A. Calibration and Validation of Agent-Based Models of Land Cover Change / T.A. Ngo, L. See // Agent-Based Models of Geographical Systems / ed. by A. Heppenstall [et al.]. — Dordrecht : Springer, 2012. — P. 181–196. — DOI: 10.1007/978-90-481-8927-4_10. — URL: https://doi.org/10.1007/978-90-481-8927-4_10
13. Wu J. Power Laws in Social Networks / J. Wu // Social Network Computing. — Singapore : Springer, 2025. — DOI: 10.1007/978-981-97-4084-0_8. — URL: https://doi.org/10.1007/978-981-97-4084-0_8

Лекция 6. Сложность, эмерджентность и коллективное поведение

Настоящая лекция посвящена одному из наиболее фундаментальных явлений, с которыми сталкивается исследователь агентных систем: феномену сложности и вытекающему из неё эмерджентному поведению. Именно здесь формальная логика локальных взаимодействий встречается с богатством глобальных паттернов, которые нельзя «считать» из поведения отдельного агента. Понимание этих концепций является ключом к моделированию реальных социальных, экономических и биологических систем.

6.1. Классы сложности: от Уивера к современным подходам

Понятие сложности интуитивно знакомо каждому, однако его строгое определение долгое время оставалось предметом дискуссий. Ключевое наблюдение специалистов по сложным системам состоит в следующем: большие системы — то есть системы из множества элементов — не обязательно являются сложными. Число элементов само по себе не определяет «сложность» в научном смысле этого слова.

Одну из первых и наиболее влиятельных классификаций предложил американский математик и учёный **Уоррен Уивер** (Warren Weaver) в 1948 году [1]. Он выделил три типа систем, с которыми имеет дело наука. **Организованная простота** (Organized Simplicity) — это детерминированные системы, поведение которых полностью описывается небольшим числом уравнений. Примерами служат маятник, планетарные орбиты, электрические цепи. Наука XIX века в значительной мере занималась именно такими системами: всё определено, всё предсказуемо, причинно-следственные связи прозрачны. **Неорганизованная сложность** (Disorganized Complexity) — это системы с большим числом элементов, ведущих себя случайным образом. Ключевое свойство таких систем — возможность описания с помощью статистических методов и законов больших чисел. Молекулы газа в сосуде, случайные блуждания частиц, популяции с пуассоновским распределением — всё это неорганизованная сложность. Статистическая физика и теория вероятностей, активно развивавшиеся с начала XX века, — инструменты именно для этого класса. **Организованная сложность** (Organized Complexity) — это тот класс явлений, который не попадает ни в первую, ни во вторую категорию. Такие системы состоят из множества элементов, взаимодействующих нелинейно и порождающих устойчивые паттерны и структуры, которые

невозможно предсказать, исходя лишь из поведения отдельных составляющих. Это и есть область эмерджентности — биологические, социальные, экономические системы. Активное изучение таких систем началось приблизительно с середины XX века.

Джеральд Вайнберг (Gerald M. Weinberg) дополнил классификацию Уивера [2], сделав акцент на перспективе наблюдателя: сложность системы зависит не только от самой системы, но и от того, как наблюдатель смотрит на неё и какие вопросы он задаёт. Важный вклад в операционализацию понятия сложности внёс анализ её проявлений в разных «измерениях» наблюдаемой системы [4]. Для временных рядов организованная простота соответствует регулярным, детерминированным сигналам (синусоида, осциллятор); неорганизованная сложность — белому шуму; организованная сложность — сигналам с несколькими масштабами изменчивости, выраженными паттернами. Для пространственных структур аналогия очевидна: упорядоченные регулярные паттерны, хаотическое случайное распределение и, наконец, многомасштабные неоднородные структуры. Для топологии взаимодействий эта триада воплощена в регулярных сетях, случайных сетях и безмасштабных сетях (рисунок 6.1).

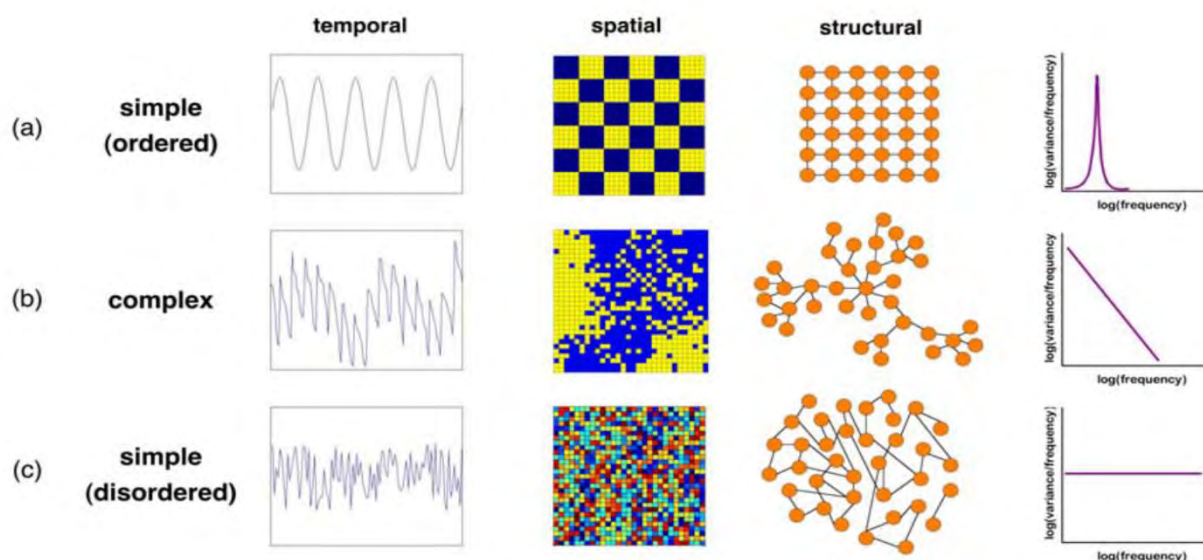


Рисунок 6.1 — Временные, пространственные и структурные характеристики простых упорядоченных (а), сложных (б) и простых неупорядоченных (с) систем — для каждого случая приведён также частотный анализ (Источник: [4])

6.2. Свойства сложных систем

Множество дисциплин — физика, биология, экономика, социология — дали собственные определения сложных систем.

Несмотря на различия в терминологии и акцентах, можно выделить несколько ключевых свойств, присутствующих во всех подходах и описанных, в частности, в монографии Нино Боккара [3].

Первое и наиболее очевидное свойство — **множество взаимодействующих агентов** (элементов, частиц). Важно подчеркнуть, что само по себе большое число агентов не делает систему сложной: требуется именно взаимодействие, обмен информацией, ресурсами или влиянием между составляющими системы.

Второе, и центральное, свойство — **эмерджентное поведение** (emergent behavior), то есть поведение системы, которое напрямую не следует из поведения отдельного агента. Это переход от микроуровня (правил поведения каждого агента) к макроуровню (свойств системы в целом). Ни одному отдельному агенту не «предписано» создавать те паттерны, которые возникают на системном уровне.

Третье свойство — **отсутствие централизованного управления**. В сложной системе нет «главного агента», раздающего команды остальным. Системные структуры и паттерны возникают благодаря локальным взаимодействиям, то есть через самоорганизацию.

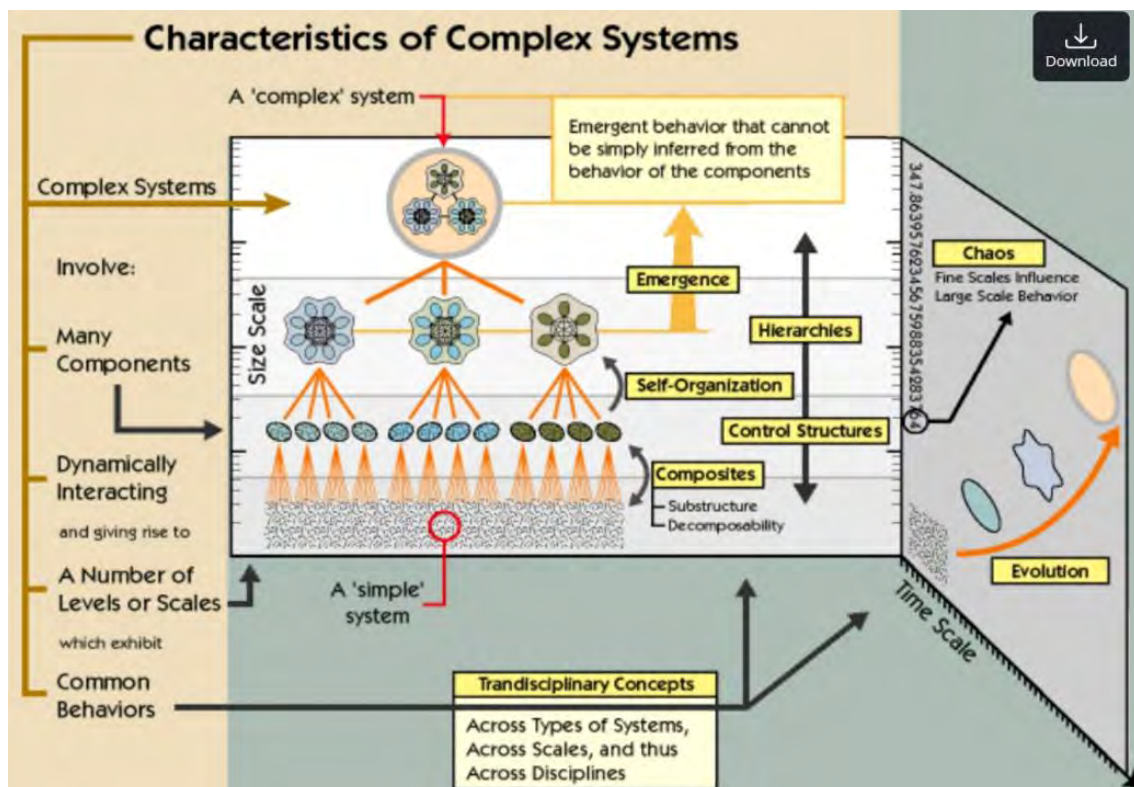


Рисунок 6.2 — Характеристики сложных систем: многоуровневая структура, самоорганизация, иерархии, эмерджентность и эволюция (Источник: [18])

Помимо перечисленных, существуют три ключевых условия, при которых сложность и эмерджентность реализуются в полной мере. Во-первых, это достаточно **большое число элементов**: при двух-трёх агентах говорить о сложных системах не приходится. Во-вторых, **дальние связи между элементами** — взаимодействия, не ограниченные ближайшим окружением агента. В моделях мобильности это выражается в возможности агента оказаться далеко от исходной позиции; в сетевых моделях — в связях между узлами, расположенными на большом топологическом расстоянии. В-третьих, **многомасштабная изменчивость**: процессы в системе разворачиваются одновременно на нескольких пространственно-временных масштабах, а взаимодействия могут охватывать сразу несколько уровней иерархии (рисунок 6.2).

6.3. Квадранты риска по Нассиму Талебу

Ещё один взгляд на проблему сложности предлагает **Нассим Николас Талеб** — статистик и финансовый эксперт, наиболее известный своей концепцией «Чёрного лебедя». В книге «Антихрупкость» [5] он разрабатывает так называемую **матрицу рисков**, которая позволяет понять, в каких системах традиционные методы прогнозирования работают, а в каких они принципиально бесполезны.

Матрица строится на пересечении двух осей: тип исходов (простые/сложные) и тип распределения результатов (нормальное/степенное). Это порождает четыре квадранта (рисунок 6.3). Первый квадрант: простые исходы с нормальным распределением. Сюда попадают монетка, выборы, спортивные матчи, казино. Риски здесь поддаются точному расчёту с помощью стандартных статистических методов. Второй квадрант: сложные исходы с нормальным распределением. Это болезни, компании, люди, продукты — системы, внутренне сложные, однако подчиняющиеся усреднённым закономерностям. Случайность здесь «смирная», контролируемая. Третий квадрант: простые исходы со степенным распределением. Землетрясения, распределение доходов, слава — здесь возникают редкие, но сильные выбросы. Вероятность события можно оценить, однако хвост распределения «тяжёлый». Четвёртый квадрант (наиболее важный для нас): сложные исходы со степенным распределением. Именно сюда попадают финансовые рынки, толпы, пандемии, конфликты, экосистемы. Здесь последствия редких событий колоссальны, а статистическое прогнозирование принципиально ненадёжно. Это и есть мир «Чёрных лебедей» — редких, непредсказуемых событий с огромными последствиями. Этот квадрант

Талеб называет «Extremistan» в противовес предсказуемому «Mediocristan».



Рисунок 6.3 — Матрица рисков по Нассиму Талебу: распределение типов систем по осям «тип исходов» и «тип распределения». Выделенные красным — системы IV квадранта, для которых эмерджентность и непредсказуемость являются нормой (Источник: [5])

Принципиальный вывод Талеба: в IV квадранте предсказание невозможно по определению — систему нельзя «просчитать», с ней нужно уметь взаимодействовать, строя «антихрупкие» структуры, способные не только выдерживать неожиданные шоки, но и извлекать из них выгоду. Именно системы IV квадранта характеризуются той организованной сложностью, которая является объектом агентного моделирования.

6.4. Сложные адаптивные системы

Сложные адаптивные системы (Complex Adaptive Systems, CAS) — это подкласс сложных систем, которые не только демонстрируют эмерджентные свойства, но и способны адаптироваться к изменяющимся условиям среды. Термин введен и систематически разработан в Институте Санта-Фе (Santa Fe Institute) в 1980–1990-е годы. Биологические организмы — наиболее наглядный пример CAS. Именно поэтому понятие «таксономии сложности», предложенное в [6], особенно хорошо иллюстрируется биологическими примерами. Согласно этой классификации, CAS характеризуются тремя видами сложности.

Конститутивная сложность (Constitutive Complexity): система состоит из множества элементов, расположенных неслучайно — органы, ткани, клетки организованы в иерархические структуры. «Целое» не является просто суммой частей.

Динамическая сложность (Dynamic Complexity): функциональные процессы в системе нелинейны и включают множество обратных связей. Метаболизм, иммунный ответ, нейронная активность — примеры процессов с динамической сложностью.

Эволюционная сложность (Evolutionary Complexity): структура системы зависит от её истории. Адаптивные решения, выработанные в процессе эволюции или обучения, обусловлены прошлым опытом — и в этом смысле система «помнит» свой путь.

Для многоагентных систем концепция CAS принципиально важна: агенты в таких системах не просто следуют фиксированным правилам, но меняют своё поведение в ответ на состояние среды и действия других агентов. Возникающие структуры устойчивы, но не статичны — они могут медленно эволюционировать, адаптируясь к новым условиям. Примерами CAS в широком смысле являются иммунная система, рыночная экономика, интернет, экосистемы.

6.5. Эмерджентность: определение и механизмы

Эмерджентность (Emergence) — это возникновение у системы свойств, которых нет ни у одного из её элементов по отдельности и которые не могут быть непосредственно выведены из свойств этих элементов. Термин происходит от латинского *emergere* — «всплывать», «появляться». Различают **слабую эмерджентность** (weak emergence), когда системное свойство в принципе можно вывести из микроправил, но лишь путём сложного вычисления, и **сильную эмерджентность** (strong emergence), когда системное свойство принципиально не редуцируемо к характеристикам компонентов.

Для агентного моделирования ключевой является слабая эмерджентность: именно её мы наблюдаем и изучаем в компьютерных симуляциях. Логика проста: мы задаём правила поведения агентов на микроуровне и наблюдаем, какие паттерны, структуры и закономерности возникают на макроуровне. При этом макроуровневые свойства не закодированы в правилах явно — они возникают как результат взаимодействия.

Эмерджентность может проявляться в разных формах. Эмерджентные паттерны — устойчивые пространственные или временные структуры (например, волны на фондовом рынке). Эмерджентные функции — способности, которых нет у отдельного агента (например, коллективный интеллект роя). Эмерджентные структуры — возникновение иерархий, кластеров, коалиций из неструктурированного первоначального состояния. В совокупности все

эти формы образуют то, что называют самоорганизацией (Self-Organization) — стихийным формированием упорядоченных структур без внешнего управления.

6.6. Примеры эмерджентного поведения в многоагентных моделях

Модель *Boids*: коллективное движение. Одним из самых известных и наглядных примеров эмерджентного поведения является модель *Boids* («птицеподобные объекты»), разработанная Крейгом Рейнольдсом (Craig Reynolds) в 1987 году. Модель воспроизводит поведение косяка рыб или стаи птиц на основе трёх локальных правил, которыми руководствуется каждый агент (см. раздел 3.3).

Поразительно, что из этих трёх простых правил возникает богатство коллективного поведения — стая движется как единый организм, огибает препятствия, реагирует на помехи, не распадаясь. Никакого «лидера», задающего направление, не существует. Важно подчеркнуть: мы наблюдаем не просто согласованное движение в одном направлении, а появление устойчивых групповых формаций, которые самоподдерживаются.

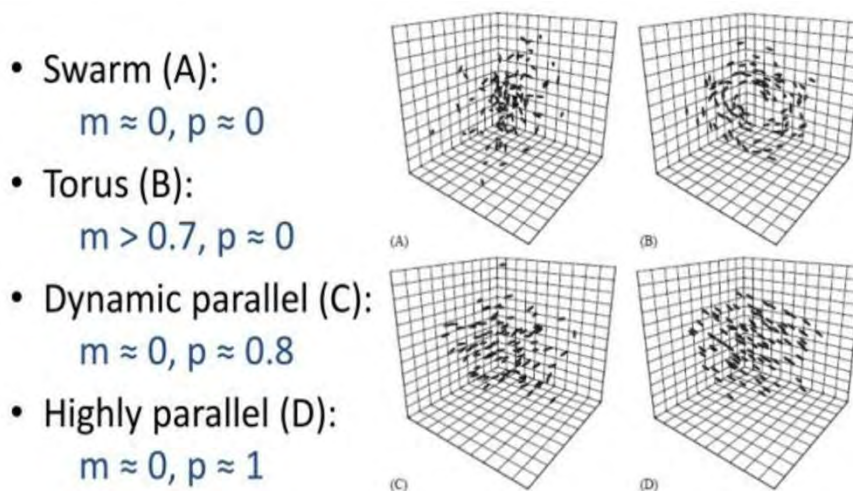


Рисунок 6.4 — Четыре типа формаций в модели *Boids* согласно классификации Кузина (Couzin): рой (A), тор (B), динамически параллельное (C) и высокопараллельное (D) движение. Параметры m и p — нормированные меры группового вращения и поляризации (Источник: [8])

Исследования И. Кузина [8] показали, что в зависимости от соотношения интенсивности правил сепарации и выравнивания система *Boids* переходит между четырьмя принципиально разными режимами коллективного движения: рой (*Swarm*) — несогласованное движение в рамках одного объёма; тор (*Torus*) — вращение по

замкнутой окружности (феномен, наблюдаемый у некоторых видов рыб); динамически параллельное движение — согласованное, но с периодическими флуктуациями; высокопараллельное движение — максимально скоординированный «полёт клином» (рисунок 6.4).

Особенно интересен феномен тора у муравьёв: если колонна муравьёв по случайности замыкается в кольцо, особи начинают ходить по кругу всё быстрее, оставляя всё более сильный феромонный след, и теоретически могут двигаться по нему до полного истощения — настолько мощным оказывается эффект самоподдерживающейся эмерджентной структуры.

Динамика толпы. Моделирование поведения людей в толпе — область, в которой переход от «организованной простоты» к «организованной сложности» особенно нагляден. Ранние подходы описывали толпу как поток жидкости (гидродинамические модели): это разумно для очень плотных, однородных и больших толп на открытом пространстве. Однако для большинства реальных сценариев — эвакуация из здания, движение через турникеты метро, перемещение на вокзале — гидродинамика оказывается невалидной: толпа слишком неоднородна, препятствия слишком специфичны, а индивидуальные различия людей слишком существенны.

Агентный подход позволяет учесть гетерогенность: каждый человек является отдельным агентом с собственными характеристиками скорости, цели, «нетерпеливости». Одной из наиболее популярных моделей является модель **социальных сил** (social forces) [9], в которой на каждого агента действуют «силы» — притяжение к целевой точке и отталкивание от других агентов и физических препятствий. Математически движение агента описывается уравнением Ньютона: $dv/dt = (v_0 \cdot e - v) / \tau + \sum f$, где v_0 — желаемая скорость, e — единичный вектор к цели, τ — время реакции, f — суммарная сила отталкивания.

Параметры модели могут быть индивидуализированы: скорость пожилого человека с тростью отличается от скорости бегущего студента, и эта разница радикально меняет динамику всей толпы. На макроуровне возникают эмерджентные явления: спонтанное расслоение встречных потоков на параллельные полосы, образование пробок у «бутылочного горлышка», колебания плотности у выходов. Именно такие модели применяются при проектировании станций метро, стадионов, торговых центров и при разработке планов эвакуации.

Модель Лотки–Вольтерры: хищник–жертва. Модель хищник–жертва (Lotka–Volterra model), независимо предложенная Альфредом Лоткой (1925) и Вито Вольтеррой (1926), описывает динамику двух взаимозависимых популяций — жертвы (V) и хищника (P). В своей простейшей форме система записывается как:

$$\begin{aligned} dV/dt &= \alpha V - (\beta/N) \cdot PV, \\ dP/dt &= \varepsilon \cdot (\beta/N) \cdot PV - \delta P, \end{aligned}$$

где α — скорость естественного прироста жертв, β — коэффициент «встречаемости» хищника и жертвы, ε — эффективность конверсии пищи, δ — естественная смертность хищников, N — нормировочный множитель. В базовом случае решение системы — замкнутые циклические траектории на фазовой плоскости (P , V): рост популяции хищников опережает рост числа жертв, затем нехватка корма приводит к спаду числа хищников, что в свою очередь даёт жертвам восстановиться — и цикл повторяется (рисунок 6.5).

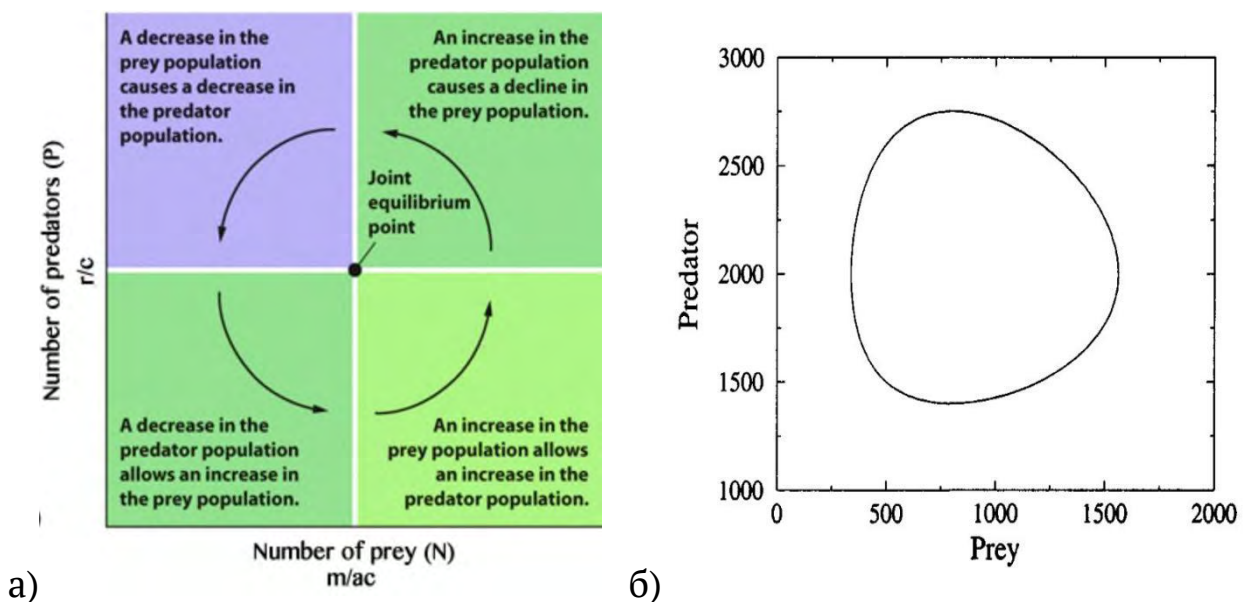


Рисунок 6.5 — Модель «хищник-жертва» (модель Лотки–Вольтерры):
а) циклическая динамика модели: четыре квадранта фазового пространства иллюстрируют взаимную обусловленность роста и спада численности хищников и жертв; б) фазовый портрет модели Лотки–Вольтерры: замкнутые траектории вокруг точки равновесия
(Источник: [18])

Однако подлинная организованная сложность возникает, когда в модель добавляется пространство и случайные флуктуации. Агентная версия модели, в которой хищники и жертвы перемещаются по среде, вступают во взаимодействие только при непосредственном контакте и реагируют на локальные условия, порождает пространственные

паттерны — динамически меняющиеся регионы с преобладанием хищников или жертв. Эти пятна не заданы в правилах модели — они возникают из локального взаимодействия, являясь чистой эмерджентностью. Пространственная модель хищника–жертвы стала хрестоматийным примером эмерджентных пространственных структур в теории сложных систем.

6.7. Коллективное поведение: толпы, кооперация и конкуренция, управление

Джеймс Шуровьески в книге **«Мудрость толпы»** [14] анализирует ситуации, в которых агрегированное мнение большой группы оказывается точнее, чем суждение любого эксперта в отдельности. Классический исторический эксперимент, описанный в книге: в 1906 году на сельскохозяйственной ярмарке в Англии 800 посетителей написали на бумажках своё предположение о весе быка. Среднее значение всех предположений оказалось точнее, чем оценка любого из присутствовавших мясников или скотоводов. Аналогичный эффект наблюдался при поиске пропавшей американской подводной лодки USS Scorpion в 1968 году: команда независимых экспертов из разных областей (гидрологи, метеорологи, капитаны подводных лодок) дала разрозненные прогнозы, но их усреднение указало почти точно на место гибели субмарины. Ключевые условия «мудрости толпы», по Шуровицкому: разнообразие мнений (diversity) — каждый участник несёт уникальную часть информации; независимость — суждения формируются без взаимного давления; децентрализация — решение не диктуется «сверху»; наличие механизма агрегации — способа свести индивидуальные мнения в коллективное. Когда эти условия нарушаются, коллективный разум превращается в коллективное безумие - **«безумие толпы»**. Именно это систематически описывает Чарльз Маккей в своей книге 1841 года [15]: тюльпановая мания, Компания Южных морей, алхимические поветрия — все эти случаи объединяет то, что Шиллер позднее назвал «нарративным пузырьём»: самоподдерживающийся поток информации, разрывающий связь между ценой и реальной ценностью. В агентных терминах это ситуация, когда разнообразие мнений исчезает — агенты начинают имитировать поведение соседей, независимость нарушается, и система переходит в режим стадного поведения. Именно эта двойственность коллективного поведения — мудрость при соблюдении условий разнообразия и независимости vs. безумие при их нарушении — является одним из

центральных объектов изучения в теории сложных систем и многоагентного моделирования.

В многоагентных системах отношения между агентами не ограничиваются взаимодействием как таковым — существенную роль играет его тип: **кооперация или конкуренция**. Эти стратегии поведения сами по себе являются источником эмерджентных явлений и часто рассматриваются в рамках эволюционной теории игр. Классическая «дилемма заключённого» показывает, что в одиночном взаимодействии доминирующей стратегией для каждого агента является предательство, хотя взаимная кооперация давала бы лучший результат для обоих. В итерированной версии игры, когда агенты встречаются многократно и «помнят» историю взаимодействий, эволюционно стабильной стратегией оказывается «Tit-for-Tat» («Зуб за зуб»): начать с кооперации и затем повторять ход противника. Результаты турниров Роберта Аксельрода [13] стали одним из первых систематических исследований коэволюции стратегий в многоагентной среде.

В пространственных многоагентных моделях кооперация может поддерживаться даже без механизма памяти — за счёт образования кластеров кооперирующих агентов, которые меньше взаимодействуют с конкурентами (**network reciprocity**). Именно это объясняет устойчивость кооперативного поведения в природных популяциях, несмотря на очевидные эволюционные преимущества предательства.

Понимание механизмов эмерджентности открывает прагматическую перспективу: если мы умеем воспроизводить в модели реальные системы, мы можем использовать эту модель для поиска управленческих решений. Именно в этом состоит логика **Simulation-Based Policymaking** — разработки политик (правил, регуляций, норм) на основе агентного моделирования. Центральный принцип подхода: в реальных социальных и экономических системах мы, как правило, не можем изменить агентов (людей, компании), но можем изменить среду или правила взаимодействия. В модели мы задаём поведение агентов на микроуровне и наблюдаем макроуровень; в реальности ситуация обратная — мы наблюдаем макропоказатели и должны подобрать правила (meso-level), при которых система ведёт себя желаемым образом. Модель Шеллинга (см. раздел 1.6) даёт хрестоматийный пример: нельзя изменить индивидуальные предпочтения горожан мгновенно, но можно изменить жилищную политику (квоты, субсидии, зонирование), то есть ввести правило на мезоуровне. Моделирование позволяет заранее оценить, к какой степени интеграции или сегрегации

приведёт та или иная политическая мера. В здравоохранении Simulation-Based Policymaking применяется для оценки мер по контролю эпидемий: модель даёт возможность сравнить эффективность карантина, вакцинации и ограничений на передвижение ещё до их реального введения. В транспортном планировании — для оценки влияния изменений дорожной инфраструктуры на образование пробок. В финансовом регулировании — для моделирования последствий новых требований к ликвидности или ограничений на короткие продажи.

6.8. Заключение

Понятие сложности — не просто философская категория, но практический инструмент анализа систем, поведение которых не поддаётся редукции к линейным зависимостям. Специфику сложных систем определяют три фундаментальных условия: множество взаимодействующих элементов, дальние связи между ними и многомасштабная изменчивость процессов.

Ключевым свойством, отличающим организованно сложные системы от простых, является эмерджентность — возникновение на макроуровне структур, функций и паттернов, не заданных явно на микроуровне. Именно эмерджентность делает многоагентное моделирование незаменимым инструментом для изучения биологических популяций, социальных и экономических процессов, финансовых рынков, динамики толпы.

Рассмотренные примеры — Voids, хищник–жертва, Шеллинг, финансовые рынки — объединяет общая логика: простые локальные правила порождают сложное глобальное поведение, которое невозможно предсказать аналитически, но можно изучить через симуляцию. С использованием подходов теории сложных систем становится возможным исследование нелинейных, неочевидных феноменов не только в уже существующих системах, но и в проектируемых искусственных средах.

Если мы можем воспроизвести сложное поведение в модели, мы можем его регулировать. Именно в этом состоит конечная цель агентного моделирования сложных систем: не только понять, как работает мир, но и понять, как его изменить — аккуратно, через воздействие на правила взаимодействия, не нарушая автономии агентов.

Рекомендуемая литература и источники

1. Weaver W. Science and Complexity / W. Weaver // American Scientist. — 1948. — Vol. 36, No. 4. — P. 536–544. — URL: <https://www.jstor.org/stable/27826250>
2. Weinberg G.M. An Introduction to General Systems Thinking / G.M. Weinberg. — New York : Wiley, 1975. — 279 p. — ISBN 978-0-471-92563-3.
3. Boccara N. Modeling Complex Systems / N. Boccara. — 2nd ed. — New York : Springer, 2010. — 490 p. — ISBN 978-1-4419-6561-5. — DOI: 10.1007/978-1-4419-6562-2. — URL: <https://doi.org/10.1007/978-1-4419-6562-2>
4. Riera-Aroca M. Signatures of Simple Ordered and Disordered Systems versus Complex Systems / M. Riera-Aroca [et al.] // Diversity. — 2010. — Vol. 2, No. 3. — P. 397–411. — DOI: 10.3390/d2030395. — URL: <https://doi.org/10.3390/d2030395>
5. Taleb N.N. Antifragile: Things That Gain from Disorder / N.N. Taleb. — New York : Random House, 2012. — 519 p. — ISBN 978-1-4000-6782-4.
6. Mitchell M. Complexity: A Guided Tour / M. Mitchell. — Oxford : Oxford University Press, 2009. — 349 p. — ISBN 978-0-19-512441-5.
7. Reynolds C.W. Flocks, Herds, and Schools: A Distributed Behavioral Model / C.W. Reynolds // Proceedings of ACM SIGGRAPH'87 (Computer Graphics). — New York : ACM, 1987. — Vol. 21, No. 4. — P. 25–34. — DOI: 10.1145/37402.37406. — URL: <https://dl.acm.org/doi/10.1145/37402.37406>
8. Couzin I.D. Collective Memory and Spatial Sorting in Animal Groups / I.D. Couzin [et al.] // Journal of Theoretical Biology. — 2002. — Vol. 218, No. 1. — P. 1–11. — DOI: 10.1006/jtbi.2002.3065. — URL: <https://doi.org/10.1006/jtbi.2002.3065>
9. Helbing D. Social Force Model for Pedestrian Dynamics / D. Helbing, P. Molnár // Physical Review E. — 1995. — Vol. 51, No. 5. — P. 4282–4286. — DOI: 10.1103/PhysRevE.51.4282. — URL: <https://link.aps.org/doi/10.1103/PhysRevE.51.4282>
10. Lotka A.J. Elements of Physical Biology / A.J. Lotka. — Baltimore : Williams & Wilkins, 1925. — 460 p.
11. Schelling T.C. Dynamic Models of Segregation / T.C. Schelling // Journal of Mathematical Sociology. — 1971. — Vol. 1, No. 2. — P. 143–186. — DOI: 10.1080/0022250X.1971.9989794. — URL: <https://doi.org/10.1080/0022250X.1971.9989794>
12. Benenson I. Agent-Based Modeling of Residential Dynamics / I. Benenson [et al.] // Journal of Artificial Societies and Social Simulation. — 2002. — DOI: 10.18564/jasss.1873. — URL: <https://doi.org/10.18564/jasss.1873>
13. Axelrod R. The Evolution of Cooperation / R. Axelrod. — New York : Basic Books, 1984. — 241 p. — ISBN 978-0-465-02121-5.
14. Surowiecki J. The Wisdom of Crowds / J. Surowiecki. — New York : Doubleday, 2004. — 306 p. — ISBN 978-0-385-50386-0.
15. Mackay C. Extraordinary Popular Delusions and the Madness of Crowds / C. Mackay. — London : Richard Bentley, 1841. — 2 vol.
16. Cont R. Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues / R. Cont // Quantitative Finance. — 2001. — Vol. 1, No. 2. — P. 223–236. — DOI: 10.1088/1469-7688/1/2/201. — URL: <https://doi.org/10.1088/1469-7688/1/2/201>
17. Liu X.-Y. FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance [Electronic resource] / X.-Y. Liu [et al.]. — 2020. — URL: <https://github.com/AI4Finance-Foundation/FinRL>

18. Bonsall M.B. Predator–Prey Interactions / M.B. Bonsall, M.P. Hassell // Theoretical Ecology: Principles and Applications / ed. by R. May, A.R. McLean. — 3rd ed. — Oxford : Oxford University Press, 2007. — DOI: 10.1093/oso/9780199209989.003.0008. — URL: <https://doi.org/10.1093/oso/9780199209989.003.0008>

Лекция 7. Введение в обучение с подкреплением

Обучение с подкреплением (Reinforcement Learning, RL) представляет собой один из ключевых подходов в машинном обучении, приобретший особое значение в контексте агентного моделирования. Данная парадигма позволяет создавать интеллектуальных агентов, способных самостоятельно обучаться эффективному поведению через непосредственное взаимодействие со средой — без явного программирования целевого поведения. Именно это свойство делает обучение с подкреплением особенно ценным инструментом для моделирования сложных систем, в которых исчерпывающая формализация задачи затруднена или невозможна.

Тема обучения с подкреплением широка настолько, что ей посвящаются отдельные академические курсы. В рамках данной главы мы сосредоточимся на ключевых концепциях, формальной постановке задачи и месте данного подхода в общей архитектуре агентного моделирования. Настоящее введение опирается прежде всего на фундаментальный труд Р. Саттона и Э. Барто «Reinforcement Learning: An Introduction» [1], который по сей день остаётся основным учебником по данной теме.

7.1. Обучение с подкреплением и агентное моделирование

Схема взаимодействия агента и среды, принятая в обучении с подкреплением, достаточно точно воспроизводит общую архитектуру интеллектуального агента, которую мы рассматривали в предыдущих главах курса (рисунок 7.1). В каждый момент времени t агент воспринимает наблюдение O_t о состоянии среды, совершает действие A_t и получает от среды скалярный сигнал вознаграждения R_t . Отличительной чертой парадигмы обучения с подкреплением является именно этот сигнал вознаграждения: он не задаёт агенту конкретных инструкций о том, что делать, но косвенно указывает, насколько хорошим или плохим оказалось совершённое действие. Именно на основе накопленного опыта таких взаимодействий агент постепенно формирует и уточняет свою стратегию поведения.

Таким образом, если в классическом агентном моделировании мы описываем, как агент действует и воспринимает среду, то в обучении с подкреплением мы добавляем механизм автоматического совершенствования поведения агента на основе обратной связи от среды. Это превращает агента из статически запрограммированного актора в обучающуюся систему, способную адаптироваться к условиям

среды и находить эффективные стратегии действий даже в тех ситуациях, которые не были предусмотрены разработчиком явно.



Рисунок 7.1 – Общая схема взаимодействия агента и среды в обучении с подкреплением: агент получает наблюдение O_t , выполняет действие A_t и получает вознаграждение R_t от среды (адаптировано по [1])

7.2. Формальная постановка задачи

Гипотеза вознаграждения. Всё обучение с подкреплением строится на одной фундаментальной гипотезе, которую принято называть гипотезой вознаграждения (reward hypothesis): единственной целью агента является максимизация ожидаемого суммарного будущего вознаграждения. Формально суммарная отдача (return) G_t определяется как: $G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots$. Это весьма сильное предположение: оно постулирует, что любую цель агента можно свести к максимизации некоего скалярного показателя. На практике цели реальных агентов нередко многокритериальны и не всегда поддаются однозначной скаляризации. Тем не менее в большинстве прикладных задач такая редукция оказывается возможной и продуктивной: вознаграждение может трактоваться как денежная прибыль, уровень удовлетворённости пользователя, счёт в игре или любой другой показатель, который мы хотим оптимизировать.

Пример: робот-пылесос. Для иллюстрации формальных концепций рассмотрим классический пример из книги Саттона и Барто — простого робота-пылесоса, задача которого состоит в уборке помещения с учётом заряда батареи. Состояние системы характеризуется уровнем заряда: «высокий» (high) или «низкий» (low). Доступные действия — «искать мусор» (search), «ждать» (wait) и «перезаряжаться» (recharge). При этом действие «искать мусор» в состоянии высокого заряда с вероятностью α сохраняет высокий заряд и приносит вознаграждение r_{search} , а с вероятностью $(1-\alpha)$ заряд снижается до низкого; при низком же заряде поиск мусора с вероятностью $(1-\beta)$ рискует привести к полной разрядке (с вознаграждением -3 — штрафом за «аварийную» ситуацию).

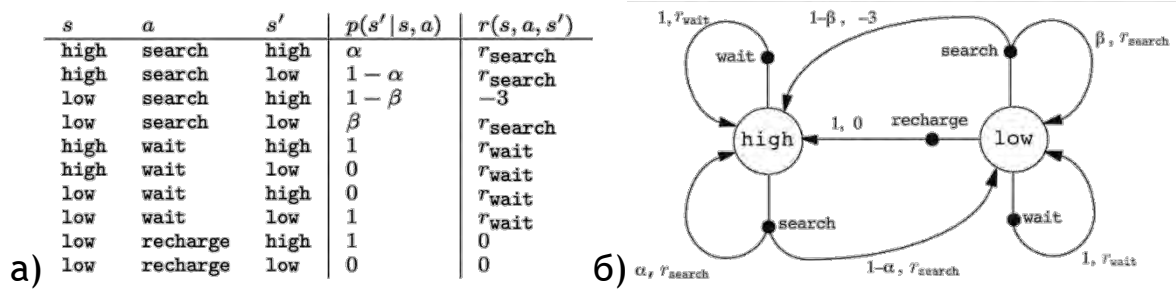


Рисунок 7.2 – Таблица переходов (а) и граф состояний (б) робота-пылесоса: для каждой пары (состояние, действие) задаётся вероятность перехода и вознаграждение (Источник: [1])

Данный пример хорошо демонстрирует структуру задачи обучения с подкреплением: у нас есть конечное множество состояний, конечное множество действий, вероятностные переходы между состояниями и скалярное вознаграждение за каждый переход. Задача агента — научиться выбирать действия так, чтобы максимизировать суммарное вознаграждение за длительный период работы.

7.3. Марковские процессы принятия решений

Свойство Маркова и MDP. Второй фундаментальный принцип обучения с подкреплением — предположение о марковости среды. **Свойство Маркова** («марковость», Markov property) означает, что вся информация, необходимая для предсказания будущего поведения системы, содержится в её текущем состоянии, а история предшествующих состояний и действий не имеет дополнительной предсказательной силы. Формально это записывается следующим образом:

$$P(S_{t+1} | S_t) = P(S_{t+1} | S_t, S_{t-1}, \dots, S_0)$$

$$P(R_t | S_t, A_t) = P(R_t | S_t, A_t, S_{t-1}, A_{t-1}, \dots, S_0, A_0)$$

Система, удовлетворяющая свойству Маркова, описывается марковским процессом принятия решений (Markov Decision Process, MDP). MDP формально задаётся пятёркой $\langle S, A, P, R, \gamma \rangle$, где S — пространство состояний, A — пространство действий, $P(s'|s, a)$ — функция переходов (вероятность попасть в состояние s' из состояния s при выполнении действия a), $R(s, a)$ — функция вознаграждения, γ — коэффициент дисконтирования. История взаимодействия агента со средой представляет собой последовательность $H_t = O_0, A_0, R_1, O_1, A_1, \dots, O_{t-1}, A_{t-1}, R_t, O_t$. Формально для принятия решений история не требуется

(достаточно текущего состояния), однако для обучения она является ценнейшим источником опыта.

Проблема частичной наблюдаемости. В классической постановке MDP предполагается, что среда полностью наблюдаема: наблюдение агента O_t совпадает с полным состоянием среды S_t^e . На практике же агент, как правило, видит лишь часть состояния системы. Такая ситуация описывается частично наблюдаемым MDP (Partially Observable MDP, POMDP), в котором состояние агента S_t^a и состояние среды S_t^e могут существенно различаться.

Рассмотрим задачу поиска пути в лабиринте. Агент воспринимает лишь небольшую окрестность вокруг себя (например, квадрат 3×3 клетки), и может оказаться, что несколько различных позиций дают идентичные локальные наблюдения. Если агент принимает решения только на основе текущего наблюдения, игнорируя историю своего пути, он нарушает свойство Маркова и рискует принять неверное решение.

Существует два основных способа справиться с проблемой частичной наблюдаемости. Первый — включение полной истории наблюдений в состояние агента. Это технически корректно, однако крайне расточительно с вычислительной точки зрения, поскольку история может быть сколь угодно длинной. Второй, более предпочтительный, подход — построение внутренней модели среды: агент формирует и постепенно уточняет своё представление о среде, которое служит достаточной статистикой для принятия решений. Именно этот подход реализуется в архитектурах агентов с памятью, о которых шла речь в предыдущих главах: сохранённое внутреннее состояние позволяет агенту «помнить» контекст своего существования и принимать обоснованные решения даже при ограниченном непосредственном восприятии.

7.4. Ценность, вознаграждение и стратегия

Мгновенное вознаграждение и функция ценности. В обучении с подкреплением необходимо чётко различать два понятия: вознаграждение (reward) и ценность (value). Вознаграждение R_t — это мгновенный сигнал, получаемый агентом от среды после совершения действия в момент времени t . Это непосредственная, локальная оценка качества одного шага. Ценность же (value) $v(s)$ — это агрегированный показатель, отражающий долгосрочные перспективы. Ценность состояния s определяется как математическое ожидание суммарного

будущего вознаграждения при условии, что агент находится в этом состоянии:

$$v_{\pi}(s) = E(G_t | S_t=s, \pi) = E(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t=s, \pi)$$

Аналогичным образом определяется ценность действия $q(s, a)$ — ожидаемое суммарное вознаграждение при выполнении действия a в состоянии s :

$$q(s, a) = E(G_t | S_t=s, A_t=a)$$

Интуиция здесь такова: состояние может иметь низкое мгновенное вознаграждение, но высокую ценность — если из него открывается путь к будущим высоким вознаграждениям. Именно поэтому цель агента — максимизировать не мгновенное, а суммарное дисконтированное вознаграждение, что делает функцию ценности центральным объектом в задаче обучения.

Коэффициент дисконтирования. В формуле суммарного вознаграждения вводится коэффициент дисконтирования $\gamma \in [0, 1]$, который определяет, насколько высоко агент ценит будущие вознаграждения по сравнению с немедленными. При $\gamma = 0$ агент является «близоруким» — он оптимизирует только ближайшее вознаграждение. При $\gamma \rightarrow 1$ агент уделяет всё большее внимание отдалённым последствиям своих действий. Выбор $\gamma < 1$ принципиально важен по ряду причин: во-первых, он обеспечивает сходимость бесконечной суммы (при $\gamma < 1$ и ограниченном вознаграждении ряд сходится); во-вторых, при $\gamma = 1$ многие алгоритмы обучения могут не сходиться или заикливаться; в-третьих, коэффициент дисконтирования отражает естественное предпочтение более ранних вознаграждений — принцип, хорошо известный из теории принятия решений и экономики.

Стратегия (policy) π — это, пожалуй, центральное понятие в обучении с подкреплением. Стратегия задаёт поведение агента: она отображает состояния в действия или в распределения вероятностей над действиями. Детерминированная стратегия — это однозначное отображение $\pi: S \rightarrow A$, то есть из каждого состояния агент всегда выбирает ровно одно действие. Стохастическая стратегия задаёт условное распределение вероятностей над действиями для каждого состояния: $\pi(a|s) = P(A_t = a | S_t = s)$. Конечная цель обучения с подкреплением, как правило, — найти оптимальную стратегию π^* , максимизирующую ожидаемую суммарную дисконтированную отдачу.

7.5. Уравнение Беллмана и оптимальные стратегии

Ключевым математическим инструментом в обучении с подкреплением является уравнение Беллмана (Bellman equation), сформулированное Ричардом Беллманом в начале 1950-х годов [2]. Уравнение выражает рекурсивную связь между ценностью текущего состояния и ценностью последующих состояний:

$$v_{\pi}(s) = E(R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s, A_t \sim \pi(s)) = E(R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s, A_t \sim \pi(s))$$

Это уравнение имеет принципиальное значение: оно показывает, что ценность состояния определяется как сумма мгновенного вознаграждения и дисконтированной ценности следующего состояния, усреднённая по всем возможным действиям и переходам. Аналогичное уравнение записывается для функции ценности действий $q_{\pi}(s, a)$. Для оптимальной стратегии π^* выполняется уравнение оптимальности Беллмана:

$$v^*(s) = \max_a E(R_{t+1} + \gamma v^*(S_{t+1}) \mid S_t = s, A_t = a)$$

Если функция ценности $v^*(s)$ найдена, оптимальная стратегия строится жадным образом: в каждом состоянии агент выбирает действие, ведущее в наиболее ценное следующее состояние. Рисунок 7.3 наглядно демонстрирует эту идею на примере лабиринта: оптимальная стратегия $\pi(s)$ (стрелки на левой панели) полностью определяется функцией ценности состояний $v_{\pi}(s)$ (числовые значения на средней панели), которая, в свою очередь, задаётся функциями переходов и вознаграждений $P(s, a, s')$, $R(s, a)$ (правая панель).

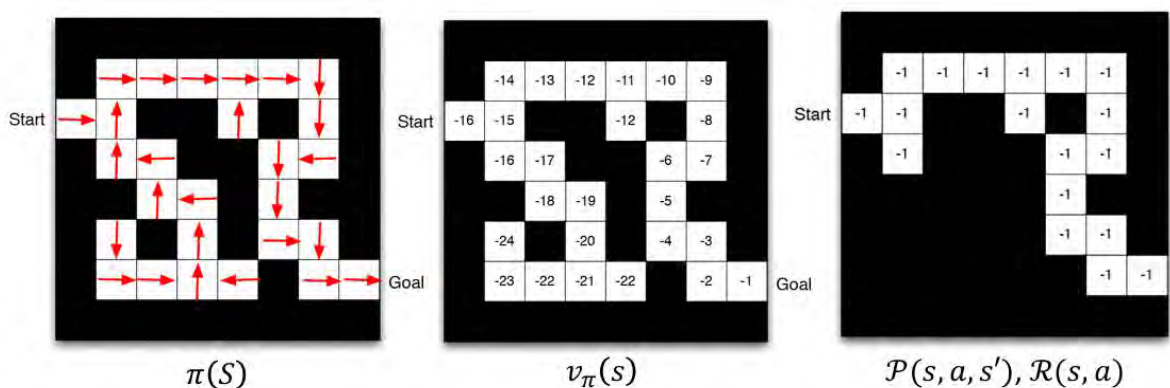


Рисунок 7.3 – Оптимальная стратегия $\pi(s)$, функция ценности состояний $v_{\pi}(s)$ и модель среды $P(s, a, s')$, $R(s, a)$ для задачи навигации в лабиринте (вознаграждение -1 за каждый шаг) (Источник: [1])

Видно, что значения ценности убывают по мере удаления от целевого состояния (Goal), суммируя количество шагов до цели с

отрицательным знаком. Оптимальная стратегия однозначно следует из этих значений: из каждой клетки агент движется в направлении наибольшей ценности. Важно отметить, что такая «карта ценностей» позволяет агенту найти оптимальный путь из любой начальной точки, что делает функцию ценности чрезвычайно мощным инструментом для задач навигации и планирования.

Существенным преимуществом использования функции ценности является её обобщаемость: зная полную карту ценностей всей среды, агент способен принимать оптимальные решения при старте из произвольной позиции, не перестраивая план заново. Именно это свойство отличает обучение с подкреплением от классического поиска пути: агент усваивает не конкретный маршрут, а «структуру ценностей» среды.

7.6. Модель среды

Модель среды в представлении агента — это его внутренние знания о том, как среда реагирует на действия. Формально модель среды включает две функции: функцию переходов $P(s'|s, a)$ и функцию вознаграждения $R(s, a)$. Если агент располагает точной моделью среды, он способен прогнозировать последствия своих действий, не взаимодействуя физически со средой, и тем самым планировать поведение заранее.

Наличие или отсутствие модели среды кардинально влияет на подход к обучению. Агенты, использующие модель среды (model-based), могут планировать на несколько шагов вперёд и требуют значительно меньшего количества взаимодействий со средой для достижения хороших результатов. Агенты без модели (model-free) обучаются непосредственно из опыта — методом проб и ошибок, — что делает их более универсальными, но менее эффективными с точки зрения использования данных. В реальных задачах модель среды чаще всего неизвестна и может быть либо задана заранее (например, в виде правил игры), либо выучена самим агентом в процессе взаимодействия.

7.7. Методы обучения с подкреплением

Методы обучения с подкреплением классифицируются по нескольким ортогональным критериям, что отражает диаграмма Венна на рис. 7.4. Прежде всего, методы делятся на основанные на функции ценности (**value-based**) и основанные на стратегии (**policy-based**). Value-based методы явно оптимизируют функцию ценности $v(s)$ или $q(s, a)$ и извлекают стратегию из неё жадным образом. Policy-based

методы напрямую параметризуют и оптимизируют стратегию, не строя явной функции ценности. Пересечение этих подходов порождает семейство методов актор-критик (actor-critic): актор (actor) — это компонент, отвечающий за выбор действий (стратегию), а критик (critic) — компонент, оценивающий качество действий (функцию ценности). Оба компонента обучаются совместно, взаимно усиливая друг друга.

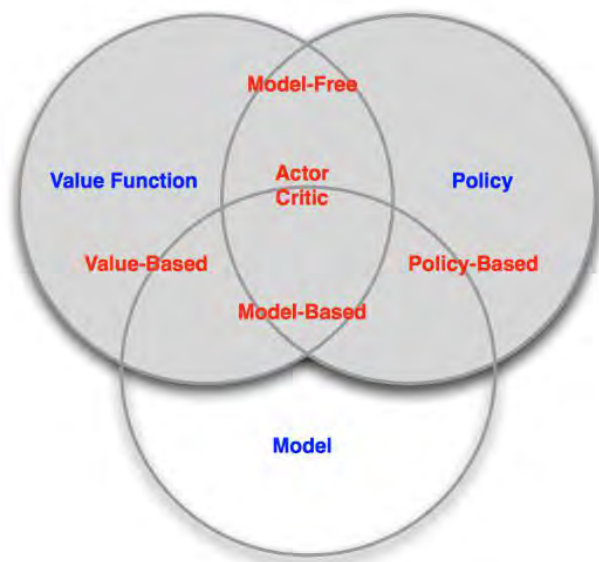


Рисунок 7.4 — Таксономия методов обучения с подкреплением.

Методы классифицируются по наличию модели среды (model-based / model-free), объекту оптимизации (value function, policy) и архитектуре (actor-critic) (Источник: [4])

Второй ключевой критерий классификации — наличие модели среды. **Model-based** методы используют явную модель переходов и вознаграждения для планирования, тогда как model-free методы обучаются непосредственно из траекторий взаимодействия со средой. Model-based подходы значительно эффективнее с точки зрения числа необходимых взаимодействий (sample efficiency), однако требуют наличия точной или обучаемой модели среды. Model-free подходы более универсальны и применимы в ситуациях, когда среда слишком сложна для явного моделирования.

Необходимо отдельно упомянуть важное направление — **обратное обучение с подкреплением** (Inverse Reinforcement Learning, IRL). Если в прямом обучении с подкреплением мы ищем оптимальную стратегию при известной функции вознаграждения, то в IRL задача инвертируется: по наблюдаемому поведению опытного агента (эксперта) мы стремимся восстановить функцию вознаграждения, которой этот агент руководствуется. Это особенно актуально для моделирования

поведения людей в задачах поведенческого клонирования (behavioral cloning), о которых пойдёт речь в следующей главе.

Исследование и использование. Одна из центральных дилемм обучения с подкреплением — конфликт между **исследованием** (exploration) и **использованием** (exploitation). Использование предполагает, что агент поступает согласно своим текущим знаниям, выбирая действия с максимальной предполагаемой ценностью. Исследование же означает выбор потенциально субоптимальных действий с целью получения новой информации о среде. Агент, который всегда эксплуатирует накопленные знания, рискует застрять в локальном оптимуме; агент, который всегда исследует, не извлекает пользы из уже приобретённого опыта.

Рассмотрим иллюстративный пример с мышью и двумя рычагами — чёрным и белым (задача многорукого бандита, multi-armed bandit). Нажатие рычагов приносит либо положительное вознаграждение (сыр, +1), либо отрицательное (удар током, -1). Предположим, что мышь последовательно нажала чёрный рычаг (получила -1), затем белый (+1), а потом трижды белый (-1, -1, -1). Если использовать среднее накопленное вознаграждение для оценки ценности рычагов, получится: $Q(\text{чёрный}) = -1$, $Q(\text{белый}) = (+1 - 1 - 1 - 1)/4 = -0.5$. Агент будет продолжать нажимать белый рычаг (ведь он «лучше»), но из-за единственного положительного результата в начале это среднее никогда не упадёт до -1, и агент будет систематически ошибаться. Ситуация показывает, что простое накопление статистики без целенаправленного исследования не позволяет корректно оценить ценность действий.

Наиболее известный способ управления балансом исследования и использования — **ϵ -жадная стратегия** (ϵ -greedy). При данном подходе с вероятностью $(1 - \epsilon)$ агент выбирает действие с максимальной текущей оценкой ценности (exploitation), а с вероятностью ϵ — случайное действие (exploration). Подбор значения ϵ зависит от задачи: при слишком малом ϵ агент может пропустить важные области пространства состояний; при слишком большом — расточительно тратить время на случайные действия. Нередко применяется убывающий ϵ , который постепенно снижается по мере накопления опыта.

Более продвинутый подход к исследованию основан на явном **учёте неопределённости**. Вместо точечной оценки ценности $Q(a)$ поддерживается распределение вероятностей над возможными значениями ценности, и исследование осуществляется там, где

неопределённость максимальна. Один из широко используемых критериев — UCB (Upper Confidence Bound): агент выбирает действие с максимальной верхней доверительной границей оценки.

Подход «**сожаления**» (regret) измеряет суммарную разницу между фактически полученным вознаграждением и тем, которое было бы получено при знании оптимального действия заранее; цель — минимизировать это сожаление с течением времени. Доказано, что определённые алгоритмы (в частности, UCB-1) обеспечивают логарифмический рост сожаления, что является асимптотически оптимальным [10].

Динамическое программирование (Dynamic Programming, DP) предоставляет базовый набор алгоритмов для точного решения задачи обучения с подкреплением при условии, что модель среды (функции переходов P и вознаграждений R) полностью известна. Идея состоит в итеративном применении уравнений Беллмана для вычисления и уточнения оценок функций ценности.

Итеративная оценка стратегии (policy evaluation) состоит в многократном применении уравнения Беллмана до сходимости: начиная с произвольной инициализации v_0 , на каждом шаге пересчитывается $v_{k+1}(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) [r + \gamma v_k(s')]$. Итерация стратегии (policy iteration) чередует оценку текущей стратегии и её жадное улучшение: стратегия улучшается путём выбора в каждом состоянии действия, максимизирующего правую часть уравнения Беллмана. Доказано, что при $\gamma < 1$ этот процесс сходится к оптимальной стратегии за конечное число итераций. Итерация ценности (value iteration) объединяет эти два шага в один, непосредственно применяя оператор оптимальности Беллмана.

Классический пример применения динамического программирования — задача перемещения агента по сеточной среде с двумя поглощающими (терминальными) состояниями. Агент получает вознаграждение -1 за каждый шаг, пока не достигнет терминального состояния. Инициализируя ценность всех состояний нулём и итеративно применяя уравнение Беллмана, мы наблюдаем, как после первой итерации все нетерминальные состояния получают значение -1 , после второй — соседи нетерминальных становятся -2 и т.д. При сходимости значения образуют «карту расстояний» до ближайшего терминального состояния, а оптимальная стратегия — следовать в направлении возрастания ценности — автоматически даёт кратчайшие пути.

Ограничение динамического программирования — необходимость полного перебора всех состояний и действий на каждой итерации. Если пространство состояний велико (а тем более непрерывно), вычислительная сложность становится запретительной. Асимптотика составляет $O(|S|^2 \cdot |A|)$ за итерацию для итерации стратегии и аналогична для итерации ценности. В задаче управления играми Atari, где «состояние» — это все пиксели экрана во всех возможных конфигурациях, прямое динамическое программирование практически неприменимо, что и мотивировало развитие методов глубокого обучения с подкреплением.

7.8. Глубокое обучение с подкреплением

Аппроксимация функций в RL. Когда пространство состояний и/или действий слишком велико для точного табличного представления, на помощь приходят методы аппроксимации. Идея состоит в том, чтобы параметрически аппроксимировать ключевые функции обучения с подкреплением: функцию ценности $v^\theta(s) \approx v(s)$, функцию ценности действий $q^\theta(s, a) \approx q(s, a)$, стратегию $\pi^\theta(a|s)$ или модель среды $P^\theta(s'|s, a)$. Параметры θ (веса нейронной сети) обучаются методами стохастического градиентного спуска на основе накопленного опыта.

Глубокое обучение с подкреплением (Deep Reinforcement Learning, Deep RL) использует глубокие нейронные сети в качестве аппроксиматоров. Это позволяет работать с высокоразмерными входами (изображения, звук) и непрерывными пространствами действий. Однако вместе с возможностями глубокое обучение приносит и ряд серьёзных проблем: нестационарность обучающих данных (среда «глазами агента» меняется по мере обучения), корреляции между последовательными наблюдениями, риск переобучения на ограниченном опыте и нестабильность обучения.

Deep Q-Network (DQN). Одним из наиболее значимых прорывов в глубоком обучении с подкреплением стала архитектура Deep Q-Network (DQN), предложенная исследователями DeepMind в 2013–2015 годах [3]. DQN аппроксимирует функцию ценности действий $Q(s, a; \theta)$ глубокой свёрточной нейронной сетью, принимающей на вход «сырые» пиксели игрового экрана. Обученная таким образом система смогла достичь сверхчеловеческого уровня игры в большинстве игр консоли Atari 2600 — примечательно, что одна и та же архитектура без каких-либо модификаций обучалась на 49 различных играх.

Для стабилизации обучения DQN использует две ключевые технические инновации. Первая — буфер воспроизведения опыта

(experience replay buffer): сохранённые переходы (s, a, r, s') случайно перемешиваются и повторно используются для обучения, что разрывает временные корреляции в данных. Вторая — отдельная «целевая» сеть (target network), чьи веса обновляются не непрерывно, а периодически, что стабилизирует цели обучения и предотвращает расходимость.

Знаковым примером является обучение агентов DeepMind игре в Atari исключительно на основе видеосигнала и игрового счёта (использованного как вознаграждение). На вход агенту подавалась только последовательность кадров экрана, а на выходе формировалась оценка ценности возможных нажатий кнопок. Таким образом, агент не получал никакой дополнительной информации об игровых правилах или структуре уровней — и, тем не менее, сумел освоить стратегии, превосходящие человеческий уровень [3].

Алгоритмы оптимизации стратегии: PPO и DPO. Алгоритм PPO (Proximal Policy Optimization) [5] — один из наиболее широко применяемых алгоритмов глубокого обучения с подкреплением, относящийся к классу методов актор-критик. Идея PPO состоит в ограничении «шага» оптимизации стратегии: чтобы обновление параметров не уводило стратегию слишком далеко от текущей, вводится ограничение на отношение вероятностей нового и старого распределений действий (clip-ограничение). Это обеспечивает стабильное обучение даже в условиях шумных оценок функции ценности. PPO нашёл широкое применение как в классических задачах управления и игровых средах, так и в системах на основе больших языковых моделей.

DPO (Direct Preference Optimization) [6] — более недавний подход, предложенный как альтернатива PPO для задач дообучения языковых моделей. В отличие от PPO, который явно оптимизирует функцию вознаграждения, DPO напрямую работает с попарными предпочтениями человека (какой из двух ответов модели лучше?) и не требует отдельного обучения модели вознаграждения. Хотя DPO изначально позиционировался как более простая и стабильная альтернатива PPO для языковых моделей, дискуссия об их сравнительных преимуществах продолжается.

Практические инструменты и среды. Для экспериментирования с алгоритмами обучения с подкреплением разработан обширный арсенал программных средств. Наиболее известной стандартной библиотекой остаётся OpenAI Gym (с 2023 года поддерживается под названием Gymnasium), предоставляющая унифицированный интерфейс к обширной коллекции сред — от простейших задач

(балансировка маятника CartPole, управление в пространстве состояний MountainCar) до сложных (игры Atari, симуляторы роботов в MuJoCo). Единый API среды (методы `reset()`, `step(action)`, `render()`) позволяет легко подключать произвольные алгоритмы и сравнивать их на стандартных бенчмарках [11].

CartPole — один из классических тестовых примеров: тележка, способная двигаться влево и вправо, несёт вертикальный маятник, который необходимо удерживать в равновесии. Наблюдение представляет собой четырёхмерный вектор (положение тележки, скорость тележки, угол маятника, угловая скорость). Вознаграждение равно +1 за каждый шаг, пока маятник не упал; при падении эпизод завершается. Несмотря на простоту, эта задача хорошо иллюстрирует ключевые свойства обучения с подкреплением: необходимость многошагового планирования, деликатный баланс между немедленными и отложенными вознаграждениями, важность исследования пространства состояний.

7.9. Многоагентное обучение с подкреплением

Многоагентное обучение с подкреплением (Multi-Agent Reinforcement Learning, MARL) расширяет классическую постановку MDP на случай, когда в одной среде действуют несколько обучающихся агентов одновременно. Это порождает принципиально новые вызовы: с точки зрения каждого отдельного агента среда становится нестационарной, поскольку её динамика определяется не только физическими законами, но и изменяющимися стратегиями других агентов. Традиционные гарантии сходимости алгоритмов обучения с подкреплением при этом нарушаются.

Взаимодействие агентов в MARL может принимать различные формы: полностью кооперативное (агенты максимизируют общее вознаграждение), полностью конкурентное (игра с нулевой суммой), а также смешанные сценарии — что отражает реальные ситуации, в которых частичное сотрудничество сосуществует с конкуренцией. Кроме того, важны вопросы централизации управления (все агенты руководствуются единой политикой против полностью децентрализованного обучения), коммуникации (есть ли у агентов канал для обмена информацией и каков его протокол) и разделения знаний (могут ли агенты делиться функциями ценности, параметрами сети, траекториями опыта).

Среди подходов к MARL выделяются: независимое Q-обучение (Independent Q-Learning, IQL), где каждый агент обучает собственную Q-

функцию, игнорируя существование других агентов; COMA (Counterfactual Multi-Agent Policy Gradient) — алгоритм актор-критик с контрфактической базовой функцией для атрибуции вклада каждого агента; MACKRL (Common Knowledge Reinforcement Learning), использующий разделяемые знания в качестве основы для координации; DecPOMDP (Decentralised Partially Observable MDP) — формализм для децентрализованных задач с частичной наблюдаемостью; LOLA (Learning with Opponent-Learning Awareness), учитывающий факт, что противники тоже обучаются.

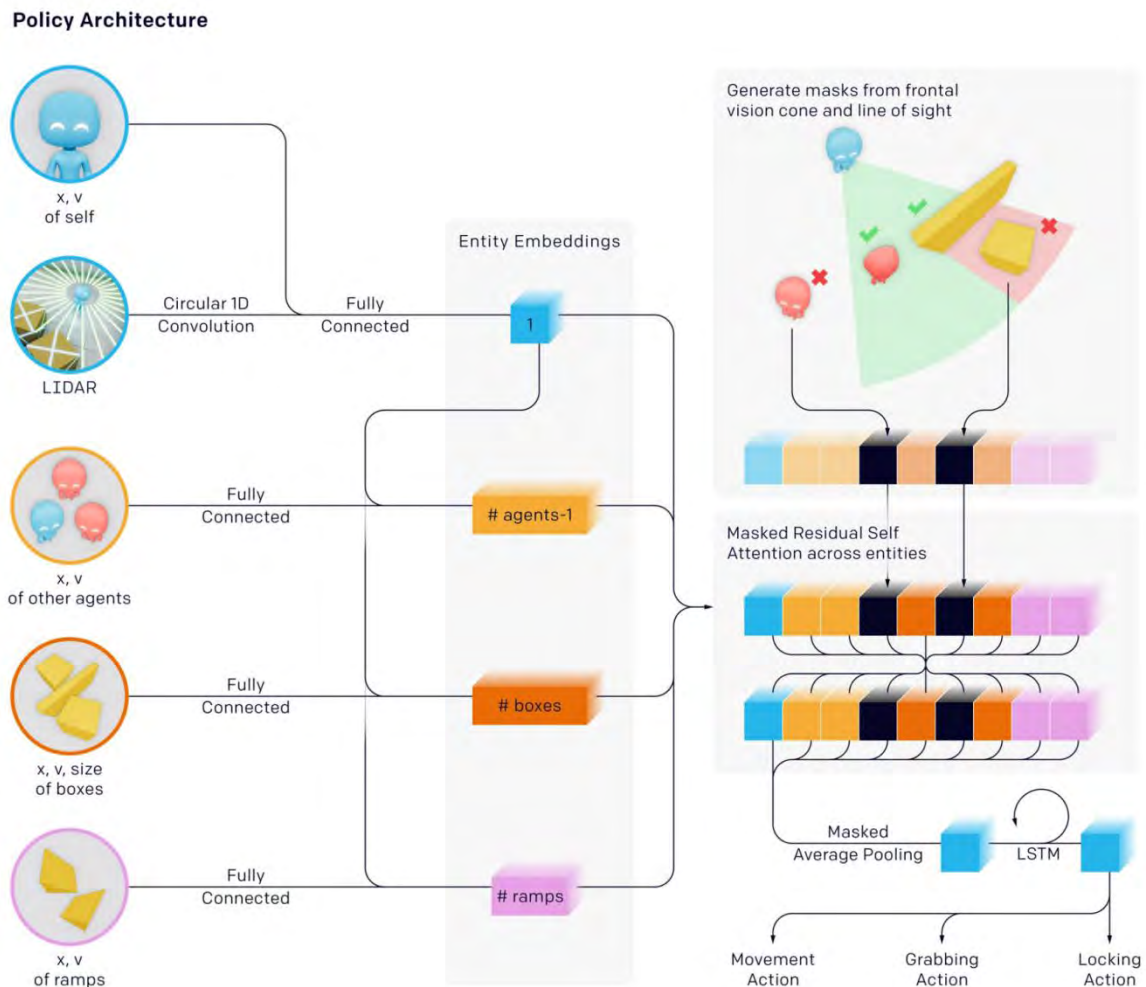


Рисунок 7.5 — Архитектура политики в задаче MARL «прятки». Агент обрабатывает информацию о своём положении, данные лидара, координаты других агентов, ящиков и рамп, используя внимание и LSTM для выработки действий (Источник: [7], <https://openai.com/index/emergent-tool-use/>)

Одним из наиболее впечатляющих примеров многоагентного обучения с подкреплением является эксперимент OpenAI с агентами, играющими в прятки [7]. В среде с разнообразными объектами (ящики, рампы, двери) команды прячущихся и ищущих агентов обучались без

каких-либо явных инструкций о стратегии. В ходе обучения из 75 миллионов эпизодов у агентов последовательно возникли шесть фаз эмерджентного поведения: от простого преследования до строительства укреплений из ящиков, контрмер по устранению укреплений, взбирания на рампы, кражи рамп и, наконец, обнаружения «бага» среды (движения на поверхности ящика) для преодоления защиты.

Этот пример демонстрирует явление, которое в главе 6 мы называли эмерджентностью: высокоуровневые, сложные стратегии возникают из простых правил взаимодействия без какого-либо явного проектирования. Агенты «изобретают» инструментальное использование объектов среды исключительно под давлением игровой конкуренции, причём каждая новая стратегия одной стороны побуждает другую сторону адаптироваться, порождая своего рода «гонку вооружений». Это делает MARL исключительно плодотворной парадигмой для моделирования эволюционных и социальных процессов.

7.10. Обучение с подкреплением на основе обратной связи от человека

Особого внимания заслуживает подход, получивший широкую известность в связи с развитием больших языковых моделей — обучение с подкреплением на основе обратной связи от человека (Reinforcement Learning from Human Feedback, RLHF). Ключевая идея RLHF состоит в использовании субъективных оценок человека в качестве сигнала вознаграждения для обучения агента, когда явно запрограммировать желательное поведение затруднительно.

Типичная схема RLHF включает три этапа. На первом этапе обучается базовая модель на большом корпусе данных (supervised pretraining). На втором собирается датасет предпочтений: людей-аннотаторов просят оценить пары ответов модели и выбрать лучший. На основе этих предпочтений обучается отдельная модель вознаграждения (reward model), которая предсказывает «человеческий рейтинг» ответа. На третьем этапе базовая модель дообучается алгоритмом PPO так, чтобы максимизировать предсказанное моделью вознаграждения, при этом не слишком отклоняясь от исходной модели (KL-дивергенция используется как регуляризация).

Проект WebGPT [8] продемонстрировал применение RLHF для обучения системы вопросно-ответного поиска на основе GPT-3. Аннотаторы оценивали качество ответов по бинарной шкале (правда/неправда, хорошо/плохо), а эти оценки использовались для

обучения через PPO. Результатом стала система, генерирующая ответы, практически неотличимые от человеческих, с обоснованными ссылками на источники.

Проект StackLLaMA [9] использовал неявные человеческие предпочтения — голоса пользователей платформы StackExchange — в качестве сигнала вознаграждения для дообучения языковой модели LLaMA. Это показало принципиальную возможность автоматического сбора сигнала вознаграждения из уже существующих источников человеческих оценок, избавляя от дорогостоящей ручной аннотации.

В обоих случаях возникает принципиальная техническая проблема, связанная с особенностями языковых моделей-трансформеров: человек оценивает готовый ответ как единое целое, тогда как модель генерирует его токен за токеном. Необходимо распространить итоговую оценку на все промежуточные шаги генерации. Это делается с помощью дисконтированного вознаграждения: итоговый балл «размазывается» по всей последовательности токенов, убывая по мере удаления от финального токена. С формальной точки зрения генерацию трансформера можно рассматривать как задачу обучения с подкреплением: состояние — текущий контекст (ранее сгенерированные токены), действие — следующий токен, вознаграждение — оценка человека за итоговый текст.

7.11. Границы применимости и открытые вопросы

Несмотря на впечатляющие успехи, обучение с подкреплением по-прежнему сопряжено с серьёзными ограничениями и открытыми вопросами. Марковское допущение — хотя и мощный инструмент упрощения — нарушается в большинстве реальных задач. Задачи с частичной наблюдаемостью, долгосрочными зависимостями и сложной временной структурой выходят за рамки классической MDP-постановки. Гипотеза скалярного вознаграждения также порождает трудности: для задач с множеством несоизмеримых целей единая функция вознаграждения не всегда может быть сконструирована без потери информации.

Другая фундаментальная проблема — низкая эффективность использования данных (sample inefficiency). Современные алгоритмы глубокого обучения с подкреплением могут требовать сотен миллионов взаимодействий со средой, прежде чем достигнут приемлемого поведения. В задачах, где взаимодействие с реальной средой дорого (робототехника, медицина, финансы), это является серьёзным практическим препятствием. Model-based методы частично решают эту

проблему за счёт «воображаемых» взаимодействий с моделью среды, однако и здесь возникают вопросы точности и обобщаемости обученной модели.

Наконец, проблема безопасности и устойчивости агентов с подкреплением остаётся во многом нерешённой. Агент, обученный максимизировать определённое вознаграждение, может находить неожиданные способы это делать, не соответствующие намерениям разработчика (так называемое «reward hacking»). Обеспечение безопасного и предсказуемого поведения обучающихся агентов в реальном мире — один из центральных вопросов современной науки об агентных системах и безопасном искусственном интеллекте.

7.12. Заключение

В настоящей главе мы рассмотрели основные концепции, методы и приложения обучения с подкреплением в контексте агентного моделирования. Мы выяснили, что обучение с подкреплением представляет собой формальный аппарат для описания обучающихся агентов, действующих в марковских средах с целью максимизации суммарного дисконтированного вознаграждения. Центральными понятиями являются функция ценности состояний $v(s)$ и действий $q(s,a)$, стратегия π , уравнение Беллмана как рекурсивное определение ценности, и коэффициент дисконтирования γ , балансирующий между немедленными и отложенными вознаграждениями.

Обучение с подкреплением применимо тогда, когда задача формулируется как MDP (или POMDP), цель сводится к скалярному вознаграждению, и агент имеет возможность накапливать опыт через взаимодействие со средой. В таких условиях RL способен находить эффективные стратегии поведения без явного программирования — от игры в Atari до управления роботами и дообучения языковых моделей. Параллельное развитие многоагентных вариантов RL открывает возможности для моделирования коллективного поведения и эмерджентных социальных явлений.

Главным ограничением остаются требования марковости и скалярного вознаграждения, а также высокая стоимость накопления опыта. Тем не менее сочетание теоретической строгости уравнений Беллмана с мощностью современных нейронных архитектур делает обучение с подкреплением одним из наиболее перспективных инструментов в арсенале исследователя агентных систем.

Рекомендуемая литература и источники

1. Sutton R.S. Reinforcement Learning: An Introduction / R.S. Sutton, A.G. Barto. — 2nd ed. — Cambridge : MIT Press, 2018. — 552 p. — ISBN 978-0-262-03924-6. — URL: <http://incompleteideas.net/book/the-book-2nd.html>
2. Bellman R. Dynamic Programming / R. Bellman. — Princeton : Princeton University Press, 1957. — 342 p.
3. Mnih V. Human-Level Control through Deep Reinforcement Learning / V. Mnih [et al.] // Nature. — 2015. — Vol. 518. — P. 529–533. — DOI: 10.1038/nature14236. — URL: <https://doi.org/10.1038/nature14236>
4. Silver D. Lectures on Reinforcement Learning [Electronic resource] / D. Silver. — London : University College London, 2015. — URL: <https://www.davidsilver.uk/teaching/>
5. Schulman J. Proximal Policy Optimization Algorithms / J. Schulman [et al.]. — arXiv:1707.06347 [cs.LG], 2017. — DOI: 10.48550/arXiv.1707.06347. — URL: <https://arxiv.org/abs/1707.06347>
6. Rafailov R. Direct Preference Optimization: Your Language Model is Secretly a Reward Model / R. Rafailov [et al.]. — arXiv:2305.18290 [cs.LG], 2023. — DOI: 10.48550/arXiv.2305.18290. — URL: <https://arxiv.org/abs/2305.18290>
7. Baker B. Emergent Tool Use from Multi-Agent Autocurricula [Electronic resource] / B. Baker [et al.]. — OpenAI, 2020. — URL: <https://openai.com/research/emergent-tool-use>
8. Nakano R. WebGPT: Browser-Assisted Question-Answering with Human Feedback / R. Nakano [et al.]. — arXiv:2112.09332 [cs.LG], 2021. — DOI: 10.48550/arXiv.2112.09332. — URL: <https://arxiv.org/abs/2112.09332>
9. Beeching E. StackLLaMA: A Hands-On Guide to Train LLaMA with RLHF [Electronic resource] / E. Beeching [et al.] // Hugging Face Blog. — 2023. — URL: <https://huggingface.co/blog/stackllama>
10. Auer P. Finite-Time Analysis of the Multiarmed Bandit Problem / P. Auer, N. Cesa-Bianchi, P. Fischer // Machine Learning. — 2002. — Vol. 47, No. 2–3. — P. 235–256. — DOI: 10.1023/A:1013689704352. — URL: <https://doi.org/10.1023/A:1013689704352>
11. Brockman G. OpenAI Gym / G. Brockman [et al.]. — arXiv:1606.01540 [cs.LG], 2016. — DOI: 10.48550/arXiv.1606.01540. — URL: <https://arxiv.org/abs/1606.01540>

Лекция 8. Модели принятия решений

8.1. Принятие решений как центральная задача агентного моделирования

Принятие решений — один из центральных элементов поведения любого агента. Именно способность делать выбор между возможными действиями, опираясь на информацию о состоянии мира и собственные цели, отличает агента от простого реактивного объекта. Архитектура агента, будь то реактивная (reactive) или когнитивная (deliberative), в конечном счёте реализует некоторую стратегию принятия решений — правило, по которому агент в каждый момент времени выбирает действие из доступного множества.

Формально процесс принятия решений описывается в рамках марковского процесса принятия решений (Markov Decision Process, MDP). MDP задаётся четвёркой $M = (S, A, T, R)$, где S — пространство состояний, A — пространство действий, $T: S \times A \rightarrow P(S)$ — функция переходов (стохастическая модель среды), $R: S \times A \rightarrow \mathbb{R}$ — функция вознаграждения. В рамках этого формализма задача агента формулируется как нахождение стратегии (policy) $\pi: S \rightarrow A$, максимизирующей ожидаемое суммарное вознаграждение.

Настоящая глава посвящена рассмотрению ключевых моделей и подходов к принятию решений, которые применяются при моделировании поведения агентов. Мы рассмотрим, как концепция рационального агента соотносится с реальными ограничениями, с которыми сталкиваются как люди, так и искусственные агенты; обсудим методы работы с множественными критериями, деревья принятия решений, теоретико-игровые подходы и модели формирования общественного мнения.

8.2. Рациональный агент и максимизация ожидаемой полезности

Классическая теория принятия решений строится на концепции **рационального агента** — агента, который всегда выбирает действие, максимизирующее ожидаемую полезность (Expected Utility, EU). Математически критерий рационального выбора записывается следующим образом:

$$\operatorname{argmax}_a EU(a|s) = \operatorname{argmax}_a \sum p(s'|a, s) \cdot U(s')$$

Здесь $p(s'|a, s)$ — вероятность перехода в состояние s' при выполнении действия a в состоянии s , а $U(s')$ — функция полезности, характеризующая предпочтительность результирующего состояния для

агента. Иными словами, рациональный агент взвешивает все возможные исходы своих действий и выбирает то действие, которое в среднем приведёт к наиболее предпочтительному результату.

Концепция рационального агента исходит из ряда жёстких предпосылок: существования единственной чётко определённой цели; ясности и однозначности задачи; полной известности всех альтернатив и последствий; стабильности предпочтений; отсутствия временных и вычислительных ограничений. В реальных системах эти предпосылки, как правило, не выполняются.

На практике агент — будь то человек в экономической системе, робот в физической среде или программный агент в цифровом пространстве — действует в условиях неполной информации, ограниченного времени и вычислительных ресурсов, множественных и порой конкурирующих целей. Именно это несоответствие между идеалом полной рациональности и реальными возможностями агентов стало главным мотивом для разработки альтернативных моделей принятия решений.

8.3. Ограниченная рациональность: «ножницы поведения»

Понятие **ограниченной рациональности** (bounded rationality) было введено Гербертом Саймоном (Herbert Simon) в 1950-х годах. Саймон предложил образный термин «ножницы поведения» (behavioral scissors), описывающий дихотомию факторов, реально определяющих выбор агента (рисунок 8.1).



Рисунок 8.1 — «Ножницы поведения» Г. Саймона: внешний контекст (Context) и когнитивные способности (Cognition) как два лезвия, совместно определяющие поведение (Источник: [1])

Одно лезвие ножниц — это **внешний контекст**: ограничения, накладываемые средой на процесс принятия решений. К ним относятся: ограниченность доступной информации и знаний; временные ограничения (агент должен принять решение в отведённый срок); физические ограничения (не все действия технически доступны); неопределённость и риски. Другое лезвие — это **внутренние (когнитивные) ограничения** самого агента: многокритериальность целей, ограниченные вычислительные ресурсы и объём памяти, а также когнитивные искажения (систематические отклонения в суждениях и умозаключениях).

Важно подчеркнуть, что когнитивные искажения — не исключительная привилегия биологических агентов. Исследования показывают, что современные языковые модели и чат-боты также демонстрируют систематические ошибки, аналогичные человеческим когнитивным искажениям, вплоть до возможности их ассоциирования с психиатрическими диагнозами. Это говорит о том, что искажения могут быть имманентным свойством любого агента с ограниченными вычислительными возможностями, а не лишь следствием биологической архитектуры.

Концепция ограниченной рациональности породила целый спектр моделей, стремящихся описать реальное поведение агентов более точно, чем идеал полной рациональности.

Удовлетворённость (satisficing). Саймон предложил, что агенты не ищут оптимальное решение, а останавливаются на первом решении, превышающем некоторый пороговый уровень приемлемости. Это вычислительно более экономично, хотя выбор самого порога представляет отдельную нетривиальную задачу.

Эвристики и когнитивные искажения (Kahneman & Tversky). Агенты используют упрощённые правила-«срезки пути», позволяющие быстро принять решение ценой потенциальной потери оптимальности. Классические примеры: эвристика доступности (availability heuristic), эвристика репрезентативности, эффект якоря.

Стоимость рассуждений (Stigler, Akerlof, Yellen). Процесс рассуждений сам по себе требует ресурсов — времени, памяти, вычислительной мощности. Модели учитывают этот факт, балансируя между точностью и стоимостью принятия решения.

Аффективные рассуждения (Slovic, Damasio). Эмоциональные состояния и интуитивные оценки («чувство») участвуют в принятии решений наравне с аналитическими рассуждениями.

Глубина рассуждений (Stahl & Wilson, 1994; Camerer). Агенты рассуждают рекурсивно: «я считаю, что ты считаешь, что я считаю...», но эта цепочка имеет конечную глубину, характерную для конкретного агента.

Модели на базе автоматов (Rubinstein). Поведение агента описывается конечным автоматом; ограничения задаются числом состояний автомата.

Обучение (Skinner, Sargent, Roth, Camerer). Агент адаптирует своё поведение на основе опыта — через адаптивное обучение, обучение с подкреплением, байесовское обновление убеждений.

Принципиальным достоинством перечисленных подходов является возможность их отображения на формализм обучения с подкреплением через соответствующие модификации: представления состояния (включение истории, наблюдений и ограничений памяти), модели системы (учёт неопределённости), а также функции вознаграждения (моделирование как внешней, так и внутренней мотивации агента).

8.4. Многокритериальные оценки

Большинство реальных задач принятия решений требует одновременного учёта множества критериев, зачастую конкурирующих между собой. Покупка жилья, выбор инвестиционного портфеля, проектирование инженерного сооружения — всё это задачи, где ни один критерий не является достаточным сам по себе. В агентном моделировании необходимость работать с многокритериальными оценками возникает всякий раз, когда функция вознаграждения не является скалярной, а представляет собой вектор.

Скаляризация. Наиболее простой подход — **скаляризация**: свести все критерии к единому числу с помощью взвешенной суммы. Формально, если агент оценивается по n критериям $c_1, c_2, \dots, c_n$, то суммарная оценка вычисляется как $V = \sum_i w_i \cdot c_i$, где w_i — вес i -го критерия. Для обеспечения сопоставимости критерии предварительно нормализуют. Метод прост и хорошо интегрируется в существующие алгоритмы обучения с подкреплением. Его слабое место — необходимость явно задать веса и предположение о линейном, монотонном характере предпочтений, которое выполняется далеко не всегда.

Метод анализа иерархий (Analytic Hierarchy Process, АНП), разработанный Томасом Саати (Thomas Saaty) в 1970-х годах, предлагает системный подход к структурированию сложных задач.

Метод строит иерархию из цели верхнего уровня, промежуточных критериев и конечных альтернатив (рисунок 8.2). Весовые коэффициенты каждого уровня определяются через попарные сравнения — отношения предпочтения между элементами одного уровня иерархии.



Рисунок 8.2 — Структура иерархии АНР на примере выбора квартиры: цель → критерии (цена, размер, комнаты, близость, категория) → альтернативы (Источник: [15])

Матрица попарных сравнений (рисунок 8.3) задаётся экспертом или аналитиком: значение в ячейке (i, j) показывает, насколько критерий i важнее критерия j . Диагональные элементы равны 1, матрица антисимметрична: $a(j,i) = 1/a(i,j)$. Нормированный вектор весов извлекается как главный собственный вектор этой матрицы.

	цена	размер	комнаты	близость	категория
цена	1	3	1	1/2	5
размер	1/3	1	1/4	1/7	2
комнаты	1	4	1	1	6
близость	2	7	1	1	8
категория	1/5	1/2	1/6	1/8	1

Рисунок 8.3 — Матрица попарных сравнений критериев для задачи выбора квартиры (Источник: [15])

Мощность метода демонстрирует реальный пример управления дамбой Санкт-Петербурга (рис. 8.4): к иерархии критериев (безопасность, надёжность, экономическая эффективность) добавляется слой экспертных оценок с различными коэффициентами доверия. Итоговое ранжирование планов управления получается взвешенным суммированием по всем уровням иерархии — это позволяет учесть как объективные характеристики, так и субъективные мнения нескольких экспертов [4].

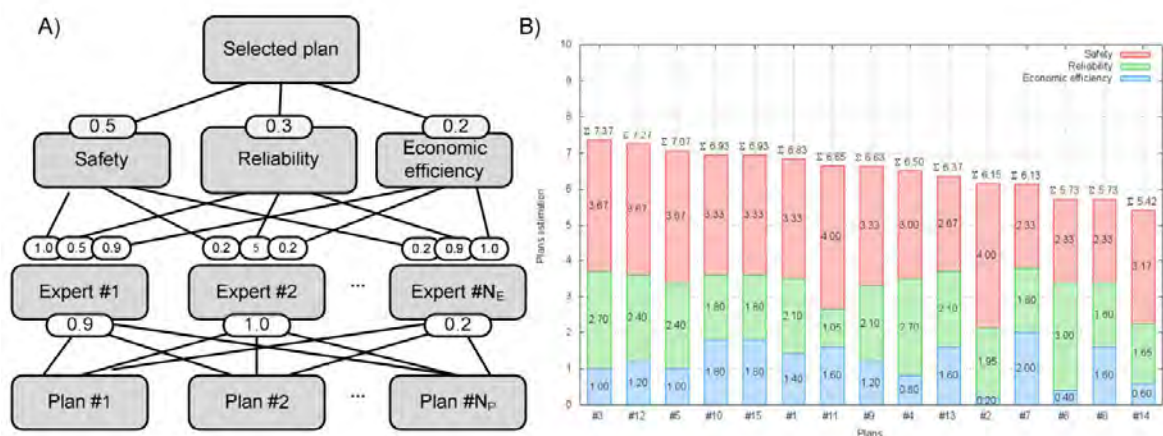


Рисунок 8.4 — Применение АНР для ранжирования планов управления дамбой Санкт-Петербурга: слева — иерархическая структура с экспертными весами; справа — итоговые оценки 15 планов по трём критериям (Источник: [4])

Фронт Парето и многокритериальное обучение с подкреплением.

Альтернативный подход к многокритериальной оптимизации основан на концепции **Парето-оптимальности**. Решение x называется *доминирующим* по отношению к y , если x не хуже y по всем критериям и строго лучше хотя бы по одному. Множество всех недоминируемых решений образует **фронт Парето (Pareto front)** — границу компромиссов между критериями. Решения, не лежащие на фронте Парето, заведомо хуже хотя бы одного недоминируемого решения и могут быть отброшены без потери качества.

В многоцелевом обучении с подкреплением (**Multi-Objective MDP, MOMDP**) фронт Парето используется как инструмент для организации фазы исследования (exploration): алгоритм сосредотачивает усилия на недоминируемых стратегиях, исключая заведомо субоптимальные варианты. Такой подход применяется, например, в алгоритмах семейства Mixture Policies и аналогичных методах. Концепция «Парето-эффективности» — классическое понятие микроэкономики, введённое итальянским экономистом Вильфредо Парето в конце XIX века, — таким образом, обретает новую жизнь в контексте обучения агентов.

При работе с многокритериальными задачами исследователи также обращаются к **субоптимальным и квазиоптимальным решениям**: в некоторых прикладных задачах нахождение точного фронта Парето вычислительно недостижимо, и приходится ограничиваться приближёнными решениями с гарантированной погрешностью.

8.5. Деревья принятия решений

Дерево принятия решений (**decision tree**) — это фундаментальная структура для представления процесса последовательного выбора. Дерево состоит из трёх типов узлов: **узлов решения** (decision nodes, обозначаются прямоугольниками D), в которых агент выбирает одно из нескольких действий; **вероятностных узлов** (chance nodes, круги C), в которых происходит случайное событие с заданными вероятностями; и **терминальных узлов** (листьев), в которых фиксируется итоговая полезность.

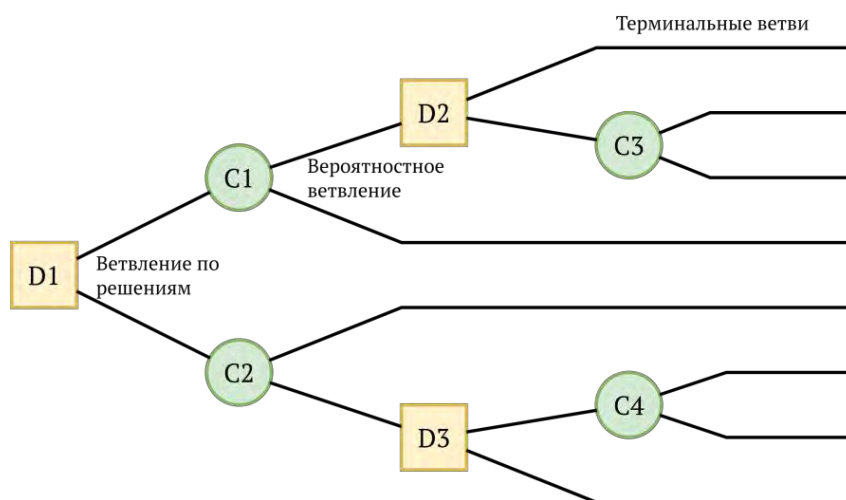


Рисунок 8.5 — Структура дерева принятия решений: квадраты D — узлы решения агента, круги C — вероятностные узлы среды

Деревья принятия решений служат трём основным целям. Во-первых, это **предсказание и оценка**: обходя дерево сверху вниз, можно оценить ожидаемую полезность каждого варианта действий, в том числе рассчитать ценность дополнительной информации. Во-вторых, это **анализ и описание**: дерево наглядно представляет структуру задачи и возможные исходы. В-третьих, это **предписание**: метод обратного распространения (backward induction) позволяет рассчитать оптимальную стратегию.

Самый простой вариант — **детерминированное дерево**, в котором нет вероятностных узлов, а все ветвления определяются значениями признаков. Классический пример — решение о прогулке в зависимости от погоды (рисунок 8.6): если облачно — гулять; если солнечно — проверить влажность; если влажность высокая — не гулять, нормальная — гулять; если дождь — проверить ветер и т.д.

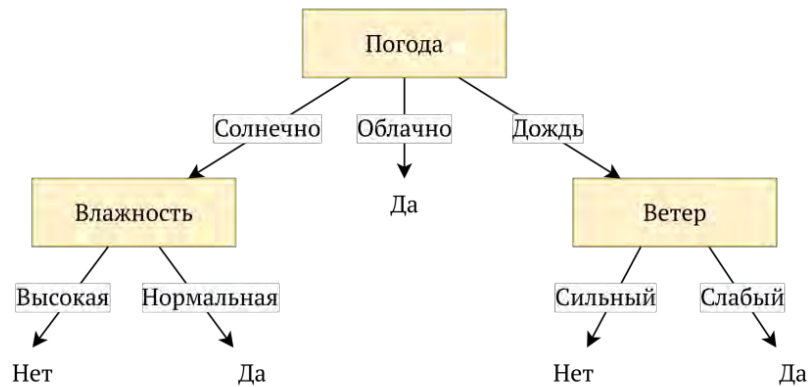


Рисунок 8.6 — Детерминированное дерево принятия решений: решение о прогулке в зависимости от погодных условий

В задачах агентного моделирования интереснее **стохастические деревья**, включающие как узлы решений агента, так и вероятностные узлы, отражающие случайную реакцию среды. В теоретико-игровой постановке дерево может содержать узлы решений нескольких агентов. Тогда, применяя метод обратной индукции и предполагая, что каждый агент действует рационально и предполагает рациональность оппонента, можно вычислить равновесные стратегии и соответствующие ожидаемые выплаты для каждого игрока. Такое дерево (рисунок 8.7) демонстрирует покерообразную ситуацию: первый игрок разыгрывает карту, второй игрок, видя ставку, принимает решение уравнивать или спасовать.

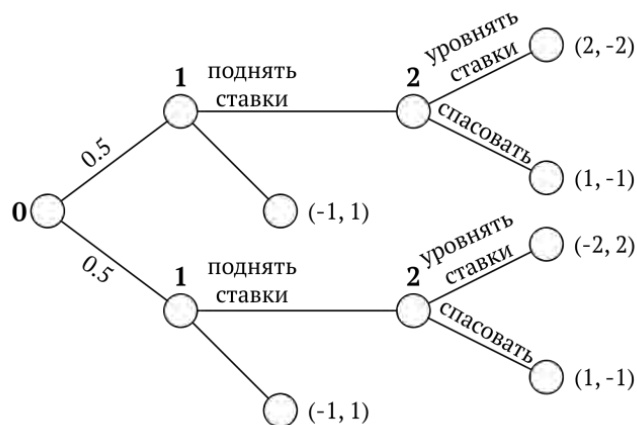


Рисунок 8.7 — Дерево решений с несколькими агентами: узлы 1 и 2 — узлы решений первого и второго игрока; 0 — начальный вероятностный узел раздачи карты; в листьях — выплаты игрокам

8.6. Ранжирование, голосование и парадокс Кондорсе

В ряде задач агент не имеет количественной функции полезности и может лишь *упорядочивать* альтернативы по своим предпочтениям.

Такие «более слабые» формы оценки включают: отношения предпочтения (А предпочтительнее В); ранжированные списки (упорядоченный перечень альтернатив); бинарный выбор (указание единственной предпочтительной альтернативы). Проблема состоит в том, что при агрегации индивидуальных предпочтений нескольких агентов в коллективное мнение возникают принципиальные трудности.

Парадокс Кондорсе. Маркиз де Кондорсе ещё в XVIII веке описал ситуацию, вошедшую в историю как **парадокс Кондорсе**: при мажоритарном голосовании трёх и более агентов по трём и более альтернативам коллективные предпочтения могут оказаться нетранзитивными, то есть образовывать цикл.

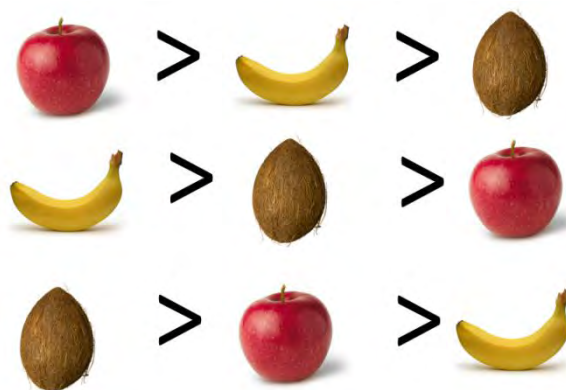


Рисунок 8.8 — Иллюстрация парадокса Кондорсе: три агента с разными ранжировками яблок (А), бананов (В) и кокосов (С). Агент 1: $A > B > C$; Агент 2: $B > C > A$; Агент 3: $C > B > A$. Мажоритарное голосование даёт цикл $A > B > C > A$

Рассмотрим три агента с предпочтениями: первый ранжирует альтернативы как «яблоки > бананы > кокосы»; второй — «бананы > кокосы > яблоки»; третий — «кокосы > яблоки > бананы». При попарном голосовании большинство предпочитает яблоки бананам (агенты 1 и 3), большинство предпочитает бананы кокосам (агенты 1 и 2), но большинство предпочитает кокосы яблокам (агенты 2 и 3). Возникает цикл: $A > B > C > A$ — и никакого «победителя» не существует.

Теорема Эрроу и проблема манипулируемости. Парадокс Кондорсе является частным проявлением более общего результата — **теоремы Эрроу о невозможности** (Arrow's impossibility theorem) [6]. Кеннет Эрроу доказал, что при наличии трёх и более альтернатив и двух и более избирателей не существует процедуры агрегации индивидуальных предпочтений, одновременно удовлетворяющей нескольким разумным требованиям: единогласию, независимости от посторонних альтернатив и отсутствию диктатора. Иными словами,

любая реальная система голосования вынуждена жертвовать хотя бы одним из этих свойств.

На практике это проявляется в возможности **манипулирования голосованием**: результат может зависеть от порядка рассмотрения вопросов, от структуры избирательных округов (рис. 8.9), от количества туров и т.д. Классический пример манипуляции через нарезку округов (gerrymandering) показан на рис. 8.9: при одинаковом общем соотношении предпочтений (27 «жёлтых» против 0 «розовых» в одном из сценариев) правильная нарезка групп может обеспечить победу меньшинству.

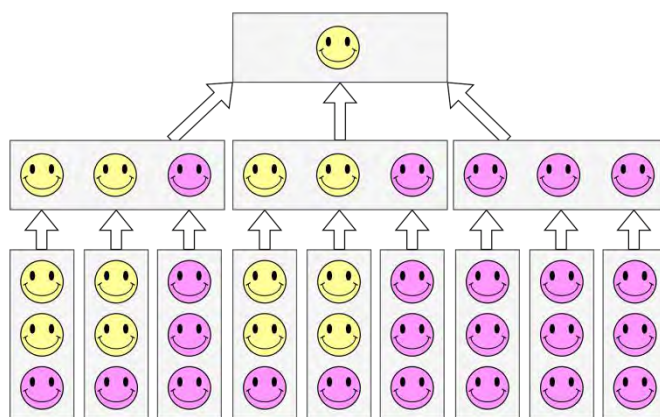


Рисунок 8.9 — Манипулирование при групповом голосовании (gerrymandering): одни и те же 27 агентов с разными предпочтениями (жёлтые и розовые) могут привести к разным результатам при разной нарезке групп

Исследования в области теории социального выбора направлены на разработку процедур голосования, наиболее устойчивых к манипуляциям, — хотя, согласно теореме Эрроу, идеальной процедуры не существует. Среди практических подходов: метод Борда (Borda count), система одобрительного голосования (approval voting), ранжированное голосование (instant-runoff voting), попарное голосование (Condorcet methods). Каждая из них имеет свои компромиссы между простотой, честностью и устойчивостью к стратегическому поведению.

8.7. Теоретико-игровые подходы

Теория игр (**game theory**) предоставляет формальный язык для описания и анализа взаимодействия рациональных агентов, принимающих решения с учётом решений друг друга. В агентном моделировании теоретико-игровые подходы позволяют как описывать поведение агентов, так и проверять (валидировать) стратегии, выработанные методами обучения с подкреплением.

Игры классифицируются по множеству признаков. По характеру взаимодействия различают **кооперативные игры** (агенты могут образовывать коалиции и перераспределять выигрыш) и **некооперативные игры** (каждый агент действует независимо). По соотношению выигрышей выделяют **игры с нулевой суммой**, где один игрок выигрывает ровно столько, сколько проигрывает другой, и **игры с ненулевой суммой**, где суммарный выигрыш может как расти, так и падать. По порядку ходов различают **последовательные** и **параллельные (одновременные)** игры. По доступности информации — игры с **полной** и **неполной информацией**. По типу стратегий — **игры с чистыми стратегиями** (агент однозначно выбирает одно действие) и **смешанными стратегиями** (агент рандомизирует выбор с заданными вероятностями).

Важно, что игроками могут выступать не только пары агентов, но и агент с внешней средой. Такая постановка задачи «агент–среда» напрямую соответствует формализму MDP, что создаёт органическую связь между теорией игр и обучением с подкреплением. Через теоретико-игровую призму можно интерпретировать и социальный характер взаимодействия: **взаимовыгодное поведение** (mutualism/reciprocity) — оба игрока в выигрыше; **эгоизм** (selfishness) — целевой игрок в выигрыше, оппонент в проигрыше; **альтруизм** (altruism) — целевой игрок в проигрыше, оппонент в выигрыше; и даже **агрессию / «злость»** (spite) — оба игрока в проигрыше.

Центральным понятием некооперативной теории игр является **равновесие Нэша** (Nash equilibrium), названное в честь Джона Нэша, — профиль стратегий, при котором ни один игрок не может увеличить свой выигрыш, в одностороннем порядке изменив свою стратегию. Это устойчивое состояние системы, «из которого никому не выгодно уходить».

Пример: дилемма заключённого. Наиболее известный пример, иллюстрирующий парадоксальный характер равновесия Нэша, — дилемма заключённого. Двух подозреваемых (назовём их Игрок 1 и Игрок 2) задерживают по отдельности. Каждому предлагают сделку: сотрудничать со следствием (сдать партнёра) или хранить молчание. Матрица выплат выглядит следующим образом (рисунок 8.10): если оба молчат — каждый получает 1 год; если оба сотрудничают — каждый получает 5 лет; если один молчит, а другой сотрудничает — молчащий получает 10 лет, а сотрудничающий выходит на свободу (0 лет).

Игрок 1 ↓ Игрок 2 →	Хранить молчание	Сдать товарища
Хранить молчание	выигрыш выигрыш	большой проигрыш большой выигрыш
Сдать товарища	большой выигрыш большой проигрыш	проигрыш проигрыш

Рисунок 8.10 — Матрица выплат в дилемме заключённого

Анализ доминирующих стратегий показывает, что для каждого игрока, каким бы ни было решение партнёра, «сдать» всегда лучше, чем «молчать». Следовательно, единственное равновесие Нэша — оба сотрудничают со следствием и получают по 5 лет. Однако взаимное молчание дало бы обоим лишь по 1 году — это очевидно более выгодный исход. Так индивидуальная рациональность приводит к коллективно худшему результату.

Дилемма заключённого находит многочисленные применения в моделировании: от описания гонки вооружений и экологических проблем до исследования эволюции кооперации в биологических системах. В многошаговой версии (итерированная дилемма заключённого) агенты могут выработать кооперативное поведение, используя стратегии типа «Tit-for-Tat» (ответное сотрудничество).

Пример: Игра «Сороконожка» и проверка рациональности.

Другой классический пример — игра «Сороконожка» (Centipede Game), предложенная Розенталем в 1981 году. Два игрока поочерёдно делают ходы. На каждом шаге текущий игрок может либо «взять» (завершить игру, забрав большую часть накопленного выигрыша) либо «продолжить» (передать ход, позволив выигрышу вырасти). Возможные исходы показаны на рисунок 8.11, где пара чисел в каждой терминальной вершине — выплаты для игроков 1 и 2.

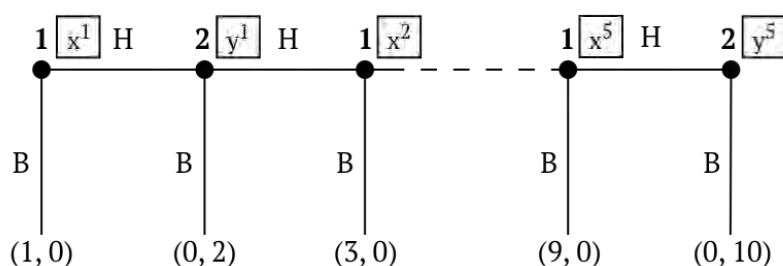


Рисунок 8.11 — Схема игры «Сороконожка»: игроки 1 и 2 по очереди выбирают «взять» (В) или «продолжить» (Н); числа в листьях — выплаты (игрок 1, игрок 2)

Применяя обратную индукцию (backward induction), нетрудно показать, что рациональному игроку следует «взять» на самом первом шаге: ведь в конце игры рациональный Игрок 2 обязательно заберёт выигрыш, значит, Игрок 1 должен взять на предпоследнем шаге, и так далее вплоть до начала. Теоретически оптимальная стратегия — завершить игру в узле (1; 0), получив суммарный выигрыш 1 единица вместо потенциальных 19 единиц.

Однако эксперименты с участием людей неизменно показывают иное: реальные игроки проходят несколько шагов, прежде чем «забрать». Это объясняется неуверенностью в рациональности партнёра: «возможно, второй игрок будет продолжать, и тогда мне выгоднее подождать». Игра «Сороконожка» служит прекрасной экспериментальной проверкой степени рациональности агентов и иллюстрирует, что в реальных системах глубина стратегических рассуждений конечна и определяется психологическими и когнитивными факторами.

Теория игр и обучение с подкреплением: взаимосвязь. Теория игр и обучение с подкреплением дополняют друг друга. Обучение с подкреплением позволяет агентам вырабатывать стратегии в сложных средах без явного задания модели, однако требует обучения и зависит от гиперпараметров. Теоретико-игровые решения оптимальны по определению, но требуют полной спецификации задачи и работают только в небольших пространствах состояний.

На практике полезна следующая схема: модель обучения с подкреплением обучается на реалистичной постановке задачи; затем выработанная стратегия верифицируется на упрощённом теоретико-игровом примере, где оптимальное решение известно аналитически. Если стратегия RL-агента совпадает с теоретико-игровым решением — это свидетельство корректности обучения; расхождение указывает на проблемы с функцией вознаграждения, алгоритмом или гиперпараметрами.

8.8. Дизайн механизмов

Теория игр описывает поведение агентов при заданных правилах игры. **Дизайн механизмов** (mechanism design) ставит обратную задачу: *как сконструировать правила игры так, чтобы рациональные агенты, действуя в собственных интересах, приходили к социально оптимальному результату?* Дизайн механизмов иногда называют «теорией игр наоборот»: вместо анализа результатов при фиксированных правилах мы синтезируем правила под желаемые результаты.

Формально механизм описывается следующими компонентами: множество сообщений игроков M ; правила распределения $x_i(M)$, определяющие, кто и что получает; правила выплат $p_i(M)$. Предполагая, что каждый игрок имеет истинную частную ценность v_i , его полезность определяется как $u_i = v_i \cdot x_i(M) - p_i(M)$. Целью дизайнера является максимизация некоторой **функции социального благосостояния** (social welfare function), при том что агенты действуют стратегически.

Аукцион Викри и механизмы второй цены. Классическим примером успешного дизайна механизма является аукцион Викри (Vickrey auction, или аукцион второй цены), разработанный Уильямом Викри в 1961 году [6], за что в 1996 году ему была присуждена Нобелевская премия по экономике. В этом аукционе все участники одновременно подают закрытые заявки; побеждает участник с наивысшей заявкой, однако платит он не свою заявленную цену, а *вторую по величине* заявку.

Этот механизм обладает замечательным свойством: для каждого участника доминирующей стратегией является указание своей *истинной* ценности объекта. Если подать меньше — можно проиграть аукцион, хотя объект стоит для вас дороже. Если подать больше — можно выиграть, но заплатить больше, чем объект для вас стоит. Механизм обеспечивает **стимульную совместимость** (incentive compatibility): агентам не нужно скрывать истинные предпочтения.

Модель Хотеллинга и дизайн рынков. Дизайн механизмов распространяется на более широкую область — **дизайн рынков**. Поучительный пример — **модель линейного города (модель Хотеллинга)**. Представим одномерный город (отрезок длиной L), в котором два продавца A и B могут свободно выбирать своё местоположение. Покупатели равномерно распределены по городу и идут к ближайшему продавцу. Цена для покупателя включает транспортные издержки — стоимость пути до продавца (рисунок 8.12).

Теоретико-игровой анализ показывает: равновесие Нэша в данной игре — положение обоих продавцов в центре города (точка $L/2$). Из любого другого положения хотя бы одному продавцу выгодно сдвинуться к центру, увеличив свой рыночный охват. Именно поэтому на практике мы наблюдаем кластеризацию конкурентов: несколько заправок на одном перекрёстке, несколько похожих политических партий в центре электорального спектра, несколько крупных торговых центров в одном районе. Это явление называют *«принципом минимальной дифференциации Хотеллинга»*.



Рисунок 8.12 — Модель Хотеллинга: два продавца А и Б делят рынок линейного города; граница раздела рынка определяется расстоянием от покупателя до каждого продавца (Источник: [9])

Инструменты дизайна механизмов позволяют исследовать, как введение регуляторных ограничений (минимальное расстояние между заправками, зонирование территорий) изменяет равновесное поведение агентов и приближает систему к социально желаемому состоянию. Аналогичные подходы используются в задачах распределения радиочастот, организации рынков электроэнергии, дизайна систем распределения органов для трансплантации и других областях.

С формальной точки зрения задача дизайна механизмов напоминает обучение с подкреплением, однако с иной ролью стратегии и функции ценности: если в стандартном RL мы фиксируем среду (функцию ценности) и оптимизируем стратегию агента, то в дизайне механизмов мы принимаем стратегию агентов как данность (они рационально оптимизируют свою полезность) и оптимизируем правила игры — функцию ценностей, которую формирует механизм.

8.9. Модели общественного мнения

Отдельное и очень активно развивающееся направление — моделирование формирования и динамики **общественного мнения**. Типичная постановка задачи предполагает сеть агентов с заданной топологией связей; каждый агент хранит некоторое «мнение» (дискретный или непрерывный параметр) и обновляет его в соответствии с мнениями соседей. Вопрос: какие коллективные состояния (консенсус, поляризация, кластеры) возникают из локальных правил взаимодействия?

Основные модели.

Модель Изинга (Ising model) — классическая модель статистической физики, описывающая взаимодействие «спинов» на

решётке (см. также раздел 5.8). В контексте моделирования мнений спин агента (+1 или -1) интерпретируется как бинарное мнение (например, за/против). Агент обновляет мнение с вероятностью, зависящей от мнений соседей и «температуры» системы (параметра, отвечающего за случайность выбора). При низкой температуре система приходит к консенсусу, при высокой — к хаотическому состоянию.

Пороговые модели (threshold models) описывают агентов, меняющих мнение, когда доля соседей с иным мнением превышает индивидуальный порог θ . Классический пример — модель «standing ovation» Миллера и Пейджа: аудитория встаёт аплодировать, если достаточная доля соседей уже встала. Такие модели воспроизводят каскадные эффекты и «эпидемии» изменения мнений.

Модели ограниченной уверенности (bounded confidence models, например, модель Дефуант или НК-модель) предполагают, что агенты обновляют мнение лишь при взаимодействии с теми соседями, чьё мнение отличается от их собственного не более чем на некоторый порог ε . Если разница мнений превышает ε , взаимодействия не происходит. Этот механизм приводит к возникновению кластеров-«сообществ» с похожими взглядами.

Модели голосования реализуют в агентной сети один из алгоритмов агрегации мнений, рассмотренных в разделе 8.6: мажоритарное голосование среди k ближайших соседей, случайный выбор одного соседа и принятие его мнения (voter model) и т.д.

Эффекты меньшинства, эхо-камеры и влияние СМИ. Даже относительно простые модели воспроизводят ряд наблюдаемых социальных феноменов. Поучительный пример — модель конформистов и «хипстеров» (рисуно 8.12). Конформисты принимают мнение большинства среди своих непосредственных соседей ($t-1$); «хипстеры», напротив, ориентируются на то, чтобы отличаться от большинства в широкой сети ($t-\tau$), причём с некоторой задержкой τ . Исследование [10] показало, что даже небольшое меньшинство «хипстеров» способно изменить коллективный исход: при правильных условиях они создают «антиистеблишментное большинство», иллюстрируя парадоксальную динамику, невидимую из простого анализа предпочтений.

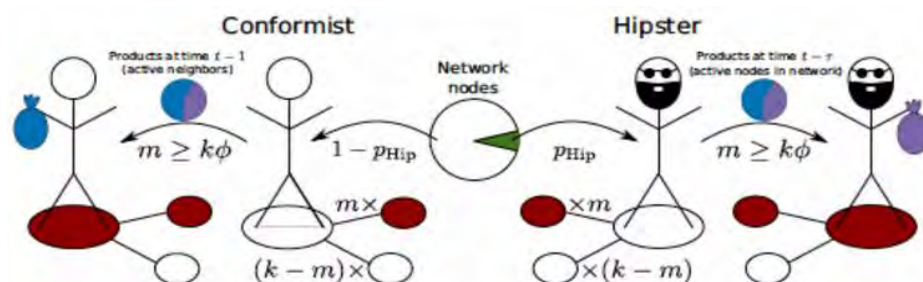


Рисунок 8.12 — Схема модели конформистов и «хипстеров»: конформисты следуют мнению активных соседей ($t-1$), «хипстеры» стремятся отличаться от большинства в сети ($t-\tau$). Параметр p_{Hip} — доля «хипстеров» (Источник: [10])

Другой наблюдаемый феномен — **эхо-камеры** (echo chambers) и **информационные пузыри**. В условиях избирательного взаимодействия (bounded confidence) агенты, чьё мнение расходится сверх порога, перестают общаться; их мнения «схлопываются» внутри кластеров, создавая у каждого агента иллюзию, что именно его взгляды разделяет большинство. На примере Twitter/X можно наблюдать, как одна и та же платформа используется совершенно по-разному несколькими слабо пересекающимися сообществами: политическим, научным, развлекательным.

Существенную роль играют **источники внешней информации** — СМИ, влиятельные блогеры (influencers), алгоритмы рекомендаций. В работе [11] демонстрируется, как в модели ограниченной уверенности сочетание числа медиа-аккаунтов и количества их подписчиков влияет на коэффициент «заражения» мнением R в сети: при росте как числа аккаунтов, так и охвата каждого из них эффект медиа экспоненциально усиливается.

8.10. Заключение

В данной главе мы рассмотрели основные подходы к моделированию процессов принятия решений агентами. Принятие решений — от простейших бинарных выборов до стратегического взаимодействия в многоагентных системах — является центральным элементом агентного поведения и требует разнообразного инструментария.

Классическая парадигма рационального агента, максимизирующего ожидаемую полезность, обеспечивает строгий математический фундамент, однако делает предпосылки, не выполняющиеся в большинстве реальных систем. Концепция ограниченной рациональности Саймона даёт концептуальную основу

для понимания реального поведения агентов через «ножницы» внешнего контекста и внутренних когнитивных ограничений.

Многокритериальные задачи решаются через скаляризацию, метод анализа иерархий, фронт Парето или субоптимальные эвристики. Деревья принятия решений обеспечивают наглядное и вычислительно эффективное представление процесса последовательного выбора. Теория игр предоставляет формальный аппарат для анализа многоагентных взаимодействий, позволяя находить равновесные состояния и выявлять парадоксы рациональности (дилемма заключённого, игра «Сороконожка»). Дизайн механизмов переворачивает задачу: вместо анализа поведения при данных правилах мы конструируем правила, ведущие к желаемому поведению.

Модели общественного мнения демонстрируют, что сложные коллективные феномены — поляризация, информационные пузыри, лавинообразное распространение взглядов — воспроизводятся из простых локальных правил взаимодействия агентов в сети. Это открывает возможности как для понимания реальных социальных процессов, так и для проектирования управляющих вмешательств.

Связь описанных подходов с обучением с подкреплением, рассмотренным в предыдущей главе, очевидна: RL выступает как универсальный инструмент выработки стратегий принятия решений, тогда как теоретико-игровые методы и аналитические решения служат для верификации и интерпретации получаемых стратегий. В следующей главе мы переходим к рассмотрению того, как эти инструменты применяются к моделированию поведения агентов реального мира.

Рекомендуемая литература и источники

1. Simon H.A. A Behavioral Model of Rational Choice / H.A. Simon // *Quarterly Journal of Economics*. — 1955. — Vol. 69, No. 1. — P. 99–118. — DOI: 10.2307/1884852. — URL: <https://doi.org/10.2307/1884852>
2. Kahneman D. Prospect Theory: An Analysis of Decision Under Risk / D. Kahneman, A. Tversky // *Econometrica*. — 1979. — Vol. 47, No. 2. — P. 263–292. — URL: <https://www.jstor.org/stable/1914185>
3. Saaty T.L. The Analytic Hierarchy Process / T.L. Saaty. — New York : McGraw-Hill, 1980. — 287 p.
4. Kovalchuk S.V. Towards the Management of the St. Petersburg Flood Prevention Facility / S.V. Kovalchuk [et al.] // *Procedia Computer Science*. — 2013. — Vol. 18. — P. 2247–2256. — DOI: 10.1016/j.procs.2013.05.392. — URL: <https://doi.org/10.1016/j.procs.2013.05.392>
5. Nash J. Non-Cooperative Games / J. Nash // *The Annals of Mathematics*. — 1951. — Vol. 54, No. 2. — P. 286–295. — URL: <https://www.jstor.org/stable/1969529>

6. Vickrey W. Counterspeculation, Auctions, and Competitive Sealed Tenders / W. Vickrey // Journal of Finance. — 1961. — Vol. 16, No. 1. — P. 8–37. — DOI: 10.1111/j.1540-6261.1961.tb02789.x. — URL: <https://doi.org/10.1111/j.1540-6261.1961.tb02789.x>
7. Arrow K.J. Social Choice and Individual Values / K.J. Arrow. — New York : Wiley, 1951. — 99 p.
8. Rosenthal R.W. Games of Perfect Information, Predatory Pricing and the Chain-Store Paradox / R.W. Rosenthal // Journal of Economic Theory. — 1981. — Vol. 25, No. 1. — P. 92–100. — DOI: 10.1016/0022-0531(81)90018-1. — URL: [https://doi.org/10.1016/0022-0531\(81\)90018-1](https://doi.org/10.1016/0022-0531(81)90018-1)
9. Hotelling H. Stability in Competition / H. Hotelling // Economic Journal. — 1929. — Vol. 39, No. 153. — P. 41–57. — DOI: 10.2307/2224214. — URL: <https://doi.org/10.2307/2224214>
10. Touboul J. The Hipster Effect: When Anticonformists All Look the Same / J. Touboul // Physical Review Research. — 2019. — Vol. 2, No. 2. — Art. 023041. — DOI: 10.1103/PhysRevResearch.2.023041. — URL: <https://doi.org/10.1103/PhysRevResearch.2.023041>
11. Pilditch T.D. Market Dynamics: When Does Being Rational Lead to Irrational Behavior? / T.D. Pilditch [et al.] // Financial Innovation. — 2020. — DOI: 10.1186/s40854-020-00211-3. — URL: <https://doi.org/10.1186/s40854-020-00211-3>
12. Deffuant G. Mixing Beliefs Among Interacting Agents / G. Deffuant [et al.] // Advances in Complex Systems. — 2000. — Vol. 3, No. 1–4. — P. 87–98. — DOI: 10.1142/S0219525900000078. — URL: <https://doi.org/10.1142/S0219525900000078>
13. Bowling M. Multiagent Learning Using a Variable Learning Rate / M. Bowling, M. Veloso // Artificial Intelligence. — 2002. — Vol. 136, No. 2. — P. 215–250. — DOI: 10.1016/S0004-3702(01)00138-7. — URL: [https://doi.org/10.1016/S0004-3702\(01\)00138-7](https://doi.org/10.1016/S0004-3702(01)00138-7)
14. Leibo J.Z. Multi-Agent Reinforcement Learning in Sequential Social Dilemmas / J.Z. Leibo [et al.] // Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017). — São Paulo, 2017. — P. 464–473.
15. Альпаков В.В. Индивидуальный рациональный выбор : методические указания по выполнению лабораторных работ / В.В. Альпаков. — 2017. — 23 с.

Лекция 9. Моделирование поведения агентов реального мира

9.1. Философия и принципы моделирования

Любое научное или инженерное исследование, направленное на понимание сложных систем, неизбежно приходит к необходимости строить их модели. Переход от реальной системы к её формальному представлению и обратно — возвращение смысла полученных результатов к реальности — является ключевым методологическим шагом, с которого начинается и которым заканчивается любой цикл моделирования. В этом контексте особую роль играет знаменитый афоризм британского статистика Джорджа Бокса: "All models are wrong, but some are useful" («Все модели ошибочны, но некоторые полезны»). Этот принцип отражает фундаментальную диалектику моделирования: каждая модель по своей природе есть упрощение реальности, а значит, она неизбежно что-то искажает или опускает. Вместе с тем именно это упрощение и делает модель инструментом, доступным для анализа.

Зачем вообще строить модели? В большинстве случаев с реальными сложными системами — социальными, экологическими, экономическими, медицинскими — невозможно напрямую экспериментировать. Мы не можем «разобрать» город, чтобы понять, как распространяется инфекция, или остановить биржу, чтобы исследовать механизм финансового пузыря. Мы не всегда способны декомпозировать систему на независимые части, выработать оптимальные управляющие решения или безопасно внедрить новые элементы без риска необратимых последствий. В этих условиях моделирование становится единственным методом, позволяющим проводить виртуальные эксперименты, оптимизировать параметры и тестировать сценарии до их реализации в действительности. Модель, таким образом, выступает как обходной путь от задачи к решению — через мир абстракций.

Как показано на рисунок 9.1, в «мире моделей» над горизонтальной границей происходят аналитические расчёты и имитационные эксперименты, тогда как в «реальном мире» формулируется проблема и применяется найденное решение. Переход вверх соответствует процессу абстрагирования и построения модели, переход вниз — интерпретации результатов и их воплощению в реальной практике. Именно эти два перехода являются методологически наиболее трудными и концептуально значимыми:

ошибки, допущенные здесь, не исправляются никакой вычислительной точностью.

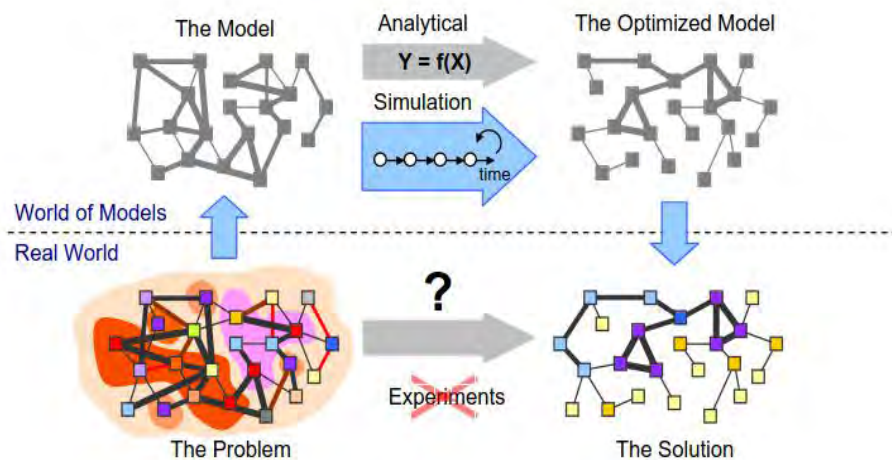


Рисунок 9.1 — Отношение «мира моделей» и «реального мира»: модель позволяет перейти от проблемы к решению там, где прямой натурный эксперимент невозможен (Источник: [8])

Существуют различные концептуальные фреймворки, описывающие процесс моделирования. Одним из наиболее авторитетных является фреймворк Бернарда Цайглера (Bernard Zeigler), предложенный в рамках теории **DEVs (Discrete Event System Specification)** [6]. В рамках этого подхода выделяются три ключевых сущности: проблемная сущность (Problem Entity / System), концептуальная модель (Conceptual Model) и компьютеризованная модель (Computerized Model). Их взаимодействие задаёт полный цикл моделирования, включающий прямые (построение модели) и обратные (верификация и валидация) шаги.

На рисунке 9.2 показана классическая треугольная схема, в которой **реальная система**, **концептуальная модель** и **компьютерная модель** связаны между собой тремя процессами. Переход от концептуальной модели к компьютерной — это программирование и реализация (Computer Programming and Implementation). Связь компьютерной модели с реальным миром проверяется операциональной валидацией (Operational Validation), а соответствие концептуальной модели реальности — валидацией концептуальной модели (Conceptual Model Validation). Нижняя дуга — верификация компьютерной модели (Computerized Model Verification) — проверяет корректность реализации концептуальной модели в программном коде. Важно подчеркнуть двунаправленность всех этих

процессов: построение модели и её проверка являются итеративными, взаимоусиливающими шагами.

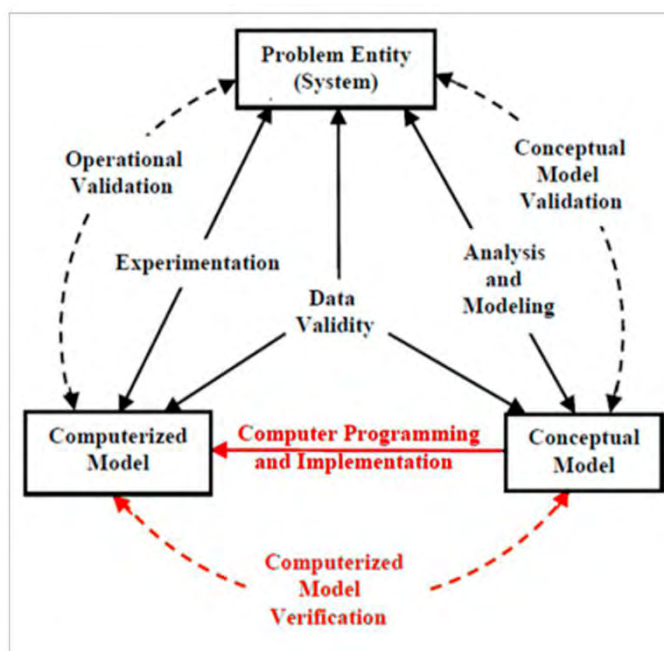


Рисунок 9.2 — Фреймворк верификации и валидации: взаимосвязь реального мира, концептуальной и компьютерной модели (Источник: [1])

Говоря о принципах моделирования в целом, следует выделить четыре ключевых аспекта: идентификацию, калибровку и усвоение данных как механизмы связи модели с реальностью; верификацию и валидацию как контроль качества; интерпретируемость как необходимое условие практической применимости; иерархичность самого процесса моделирования — от постановки задачи до экспериментов с готовой моделью.

9.2. Иерархия моделей и компоненты

Процесс перехода от реального объекта к его агентной модели является многоступенчатым и иерархическим. На первом этапе формулируется **проблема** и определяются **требования**, которым должна удовлетворять модель. Затем на основе требований строится **концептуальная модель**, включающая описание агентов, среды и правил их взаимодействия. Следующий шаг — **спецификация** имитационной модели, то есть формализованное описание, готовое к программной реализации. Наконец, осуществляется собственно **реализация** — создание программного кода и его запуск в конкретной вычислительной среде.

Принципиально важно, что эта иерархия не является строго последовательной — она итеративна. Результаты экспериментов с моделью могут потребовать пересмотра концептуальной модели или даже переформулировки самой проблемы. Кроме того, каждый уровень иерархии несёт в себе собственные артефакты и абстракции: описания классов агентов, схемы взаимодействий, архитектурные решения программной реализации. Именно на этих переходах между уровнями возникает большинство методологических проблем — и именно здесь верификация и валидация приобретают конкретный операциональный смысл.

Таким образом, разработка агентной модели реального мира включает минимум два различных класса моделей. Первый — **концептуальная модель** (conceptual model) — это описание системы на языке предметной области, не привязанное к конкретной программной реализации. Второй — **имитационная (компьютерная) модель** (simulation / computerized model) — это программный артефакт, непосредственно запускаемый на ЭВМ. Связь между ними, обеспечиваемая через верификацию и валидацию, является ключевой для обеспечения качества всего моделирования.

9.3. Типология агентных моделей: от абстрактных к эмпирическим

Агентные модели образуют широкий спектр по степени своей связи с реальностью. На одном конце этого спектра находятся **абстрактные модели** (Abstract ABMs), на другом — полностью **управляемые данными модели** (Data-Driven ABMs). Между ними располагаются **эмпирические агентные модели** (Empirical ABMs), которые, в свою очередь, подразделяются на несколько подклассов (рисунок 9.3).

Абстрактные модели (Abstract ABMs) мотивированы наблюдаемыми феноменами, однако не привязаны ни к конкретным наборам данных, ни к конкретным сценариям. Классическим примером служит модель сегрегации Шеллинга, в которой агенты двух «цветов» располагаются на прямоугольной сетке и перемещаются, если доля «чужих» соседей превышает некоторый порог. Несмотря на крайнюю простоту правил, модель демонстрирует выраженную эмерджентную сегрегацию — паттерн, наблюдаемый в реальных городах. Тем не менее, непосредственный перенос такой модели в конкретный городской контекст является самостоятельной и нетривиальной задачей.

Абстрактные модели хорошо пригодны для фундаментальных исследований — изучения механизмов коллективного поведения, условий возникновения эмерджентных паттернов, демонстрации концептуальных принципов.

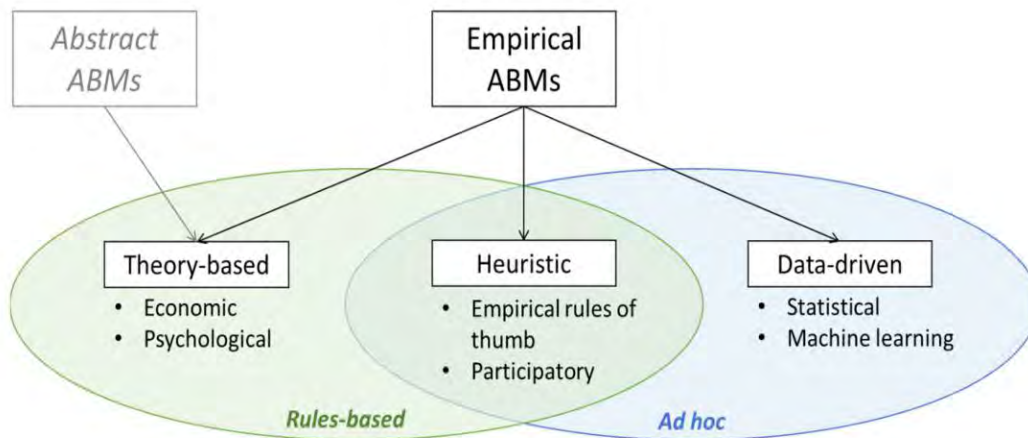


Рисунок 9.3 — Типология агентных моделей: абстрактные и эмпирические АВМ с их подклассами (Источник: [5])

Эмпирические агентные модели (Empirical ABMs) занимают промежуточное положение и делятся на три подкласса. Теоретически обоснованные модели (Theory-based) строятся на базе экономических, психологических или социологических теорий — например, теории ожидаемой полезности, теории перспектив Канемана и Тверски или теорий социального влияния. Эвристические модели (Heuristic) используют упрощённые эмпирические правила поведения, нередко получаемые через *participatory modeling* — непосредственное вовлечение экспертов или стейкхолдеров в процесс разработки модели. Они занимают пограничное положение между «чисто теоретическими» и «полностью data-driven» подходами, позволяя работать в условиях ограниченных наблюдений и высокой сложности сценария. Data-Driven ABMs строятся на основе реальных данных, что позволяет максимально точно воспроизводить поведение конкретной системы; для идентификации и калибровки таких моделей широко используются статистические методы и машинное обучение.

Выбор конкретного типа модели определяется целями исследования. Если задача состоит в изучении фундаментальных механизмов, абстрактные модели могут быть предпочтительнее. Если же необходимо дать прикладное решение с прямым выходом на управление реальной системой, то data-driven подход становится необходимым. При этом следует помнить, что цели исследования по

своей природе лежат вне самой агентной модели: модель — средство, а не конечный результат.

9.4. Имитационное моделирование: парадигмы и уровни абстракции

Современное имитационное моделирование опирается на несколько взаимодополняющих парадигм, различающихся уровнем абстракции, единицей описания и областью применения. В практике наиболее широко применяются четыре подхода, наглядно представленные на рисунок 9.4.

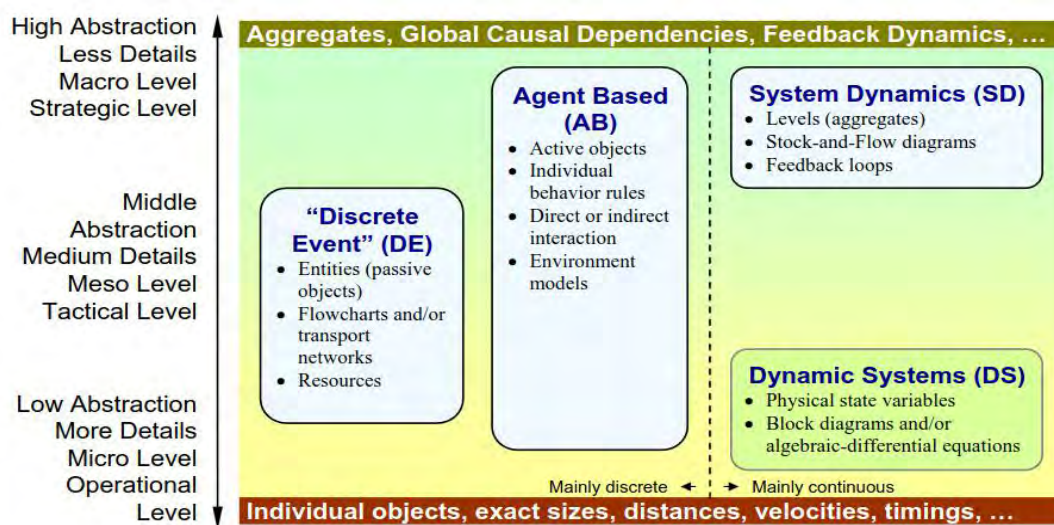


Рисунок 9.4 — Уровни абстракции и парадигмы имитационного моделирования: DE, AB, SD, DS (Источник: [8])

Дискретно-событийное моделирование (Discrete Event Simulation, DE) концентрируется на описании потоков работ (workflow) и обработки событий во времени. Мир здесь представлен пассивными объектами (entities), перемещающимися через цепочки ресурсов; типичными приложениями служат моделирование очередей, производственных линий и логистических цепочек. Детализация соответствует тактическому уровню.

Системная динамика (System Dynamics, SD) работает с агрегированными переменными — «уровнями» (stocks) и «потоками» (flows), — описывая систему через причинно-следственные петли обратной связи (feedback loops). Этот подход идеально подходит для стратегического уровня — анализа долгосрочных тенденций, политик, макроэкономических закономерностей. Он принципиально игнорирует индивидуальность отдельных элементов, оперируя только совокупными характеристиками.

Агентное моделирование (Agent-Based Modeling, AB) описывает систему как совокупность автономных активных объектов-агентов, взаимодействующих друг с другом и со средой по индивидуальным правилам поведения. Уникальной особенностью агентного подхода является то, что он охватывает все уровни абстракции на рис. 9.6 — от операционного (микро) до стратегического (макро), органично вбирая в себя элементы обоих смежных подходов. Именно поэтому агентное моделирование нередко выступает интегрирующим ядром гибридных имитационных систем.

Динамические системы (Dynamic Systems, DS) используют алгебраические и дифференциальные уравнения для описания непрерывных процессов на уровне физических переменных состояния. Этот подход ближе к численному моделированию и традиционно применяется в физике, механике и электротехнике.

Принципиально важной характеристикой современных имитационных платформ (таких как AnyLogic, Repast Symphony, NetLogo и других) является поддержка гибридных моделей, совмещающих несколько парадигм. Например, агенты могут действовать в среде, чья динамика описывается системной динамикой, а внутриагентная логика может включать дискретно-событийные автоматы переходов. Такая мультипарадигмальность существенно расширяет выразительную силу модели, позволяя адекватно воспроизводить системы, одновременно демонстрирующие индивидуальное поведение (агентный уровень) и агрегированные закономерности (системодинамический уровень).

9.5. Работа с данными в контексте агентного моделирования

Данные, на основе которых строятся и калибруются агентные модели, отличаются значительной разнородностью как по своей природе, так и по способу получения. Это принципиальное отличие реальных прикладных моделей от абстрактных, где данные могут генерироваться произвольно.

По характеру наблюдений данные делятся на **прямые** (direct) — непосредственно измеряемые характеристики — и **косвенные** (indirect) — вычисляемые или выводимые из других данных. По временному режиму выделяют **оперативные** данные (operative), требующие немедленной реакции системы, и **исторические** (historical), используемые для анализа ретроспективы и построения прогнозов. По семантике данные подразделяются на **состояния** (states) — дискретные или непрерывные характеристики системы в конкретный момент времени; **события** (events) — факты, связанные с конкретным

моментом времени; и **характеристики** (attributes) — описательные параметры агентов и среды. Важно понимать, что даже непрерывный по своей природе процесс в большинстве случаев наблюдается дискретно — в периодические или непериодические моменты времени.

Кроме того, все реальные данные в той или иной мере **неидеальны**. Они могут быть **неточными** (содержать измерительный шум), неполными (часть измерений пропущена), **субъективными** (основаны на экспертных суждениях) или **противоречивыми** (разные источники дают несовместимые значения). Отдельную проблему представляет **неопределённость** (uncertainty) — невозможность точно знать истинное состояние системы на основе имеющихся наблюдений.

Для непосредственной работы с данными в рамках агентного моделирования принято различать три связанных процесса: **идентификацию модели** — определение её структуры и параметров на основе данных; **калибровку** — настройку параметров существующей модели так, чтобы минимизировать расхождение с наблюдаемыми данными (обычно по историческим данным); и **усвоение данных** — оперативную коррекцию состояния модели в реальном времени (см. следующий раздел).

9.6. Усвоение данных и концепция DDDAS

Усвоение данных (Data Assimilation, DA) — это метод оперативной коррекции состояния модели на основе поступающих наблюдений. В отличие от калибровки, которая является однократной настройкой параметров по историческим данным, усвоение — это непрерывный динамический процесс (рисунок 9.5). Его основная идея состоит в следующем: в любой момент времени у нас есть прогноз модели (который содержит некоторую неопределённость) и наблюдение реальной системы (которое тоже содержит ошибку). Усвоение данных позволяет найти оптимальную оценку истинного состояния системы, учитывающую оба источника информации с весами, обратно пропорциональными их ошибкам.

Концептуально этот подход давно используется в гидрометеорологии, где прогноз численных моделей атмосферы регулярно корректируется по данным наблюдений метеостанций и спутников. Однако применение усвоения данных к агентным моделям представляет значительные трудности, поскольку классические методы, прежде всего фильтр Калмана (Kalman Filter), требуют линейности переходных уравнений состояния и гауссовости шумов — условий,

которые в агентных моделях практически никогда не выполняются в силу их дискретной и эвристической природы.

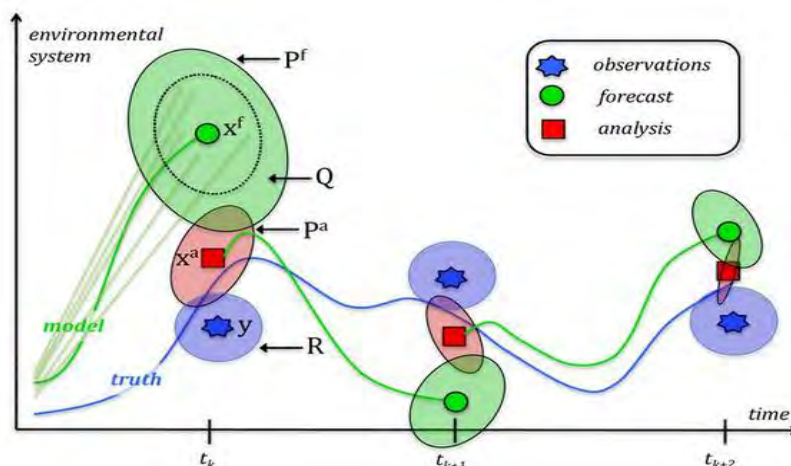


Рисунок 9.5 — Принцип усвоения данных: корректировка ансамблевого прогноза модели на основе зашумлённых наблюдений (Источник: [7])

Тем не менее существуют методы усвоения, применимые к агентным моделям. Фильтр частиц (Particle Filter, Sequential Monte Carlo), не требующий линейности и нормальности распределений, показывает хорошие результаты как на синтетических, так и на реальных задачах [9]. Его суть заключается в представлении неопределённости состояния ансамблем «частиц» — гипотетических состояний системы, — каждая из которых обновляется в соответствии с динамической моделью, а весовые коэффициенты пересчитываются в зависимости от правдоподобия наблюдения. В итоге фильтр «притягивает» ансамбль к наиболее вероятному состоянию, не доверяя слепо ни модели, ни измерению (рисунок 9.6а).

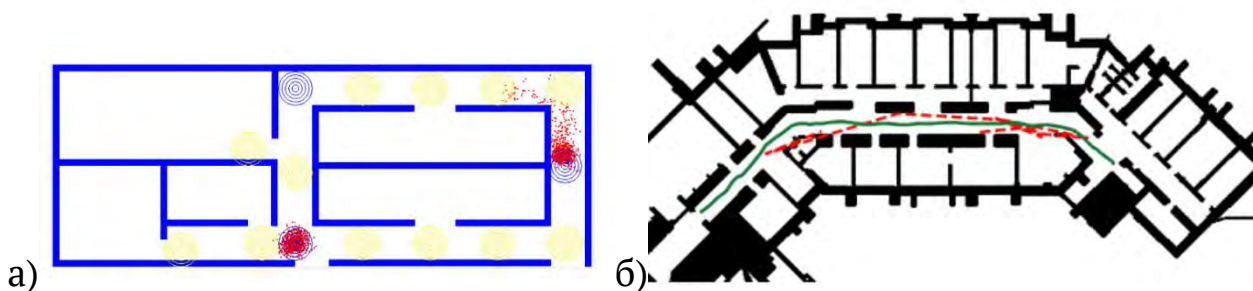


Рисунок 9.6 — Примеры решения задачи усвоения данных в агентных моделях: а) синтетический лабиринт (Источник: [9]); б) позиционирование медицинского персонала в «умном» госпитале (Источник: [10])

Показательным примером служит задача позиционирования медицинского персонала в «умном» госпитале (рисунок 9.6б). Датчики, расположенные в помещениях, и браслеты сотрудников позволяют

отслеживать перемещения агентов. Однако в условиях закрытого пространства с многочисленными стенами прямые методы позиционирования дают значительные ошибки: агенты на карте могут «телепортироваться» через стены или внезапно смещаться на несколько метров. Применение фильтра частиц в сочетании с агентной моделью, в которой стены непроницаемы, позволяет устранить физически невозможные переходы — траектории движения становятся плавными и естественными. Более того, усвоение может работать не только на уровне позиционирования, но и на уровне выявления плана деятельности агента: наблюдая за перемещениями, можно оценить, какому поведенческому сценарию следует агент в данный момент (например, «обходит палаты» или «ожидает у сестринского поста»).

Dynamic Data-Driven Application Systems (DDAS) представляет собой более общую архитектурную парадигму, в которой моделирование и управление физической системой объединены в два взаимосвязанных цикла обратной связи. Термин был введён Фредериком Дарема (Frederica Darema) в начале 2000-х годов и стал основой для целого научного сообщества (рисунок 9.4).

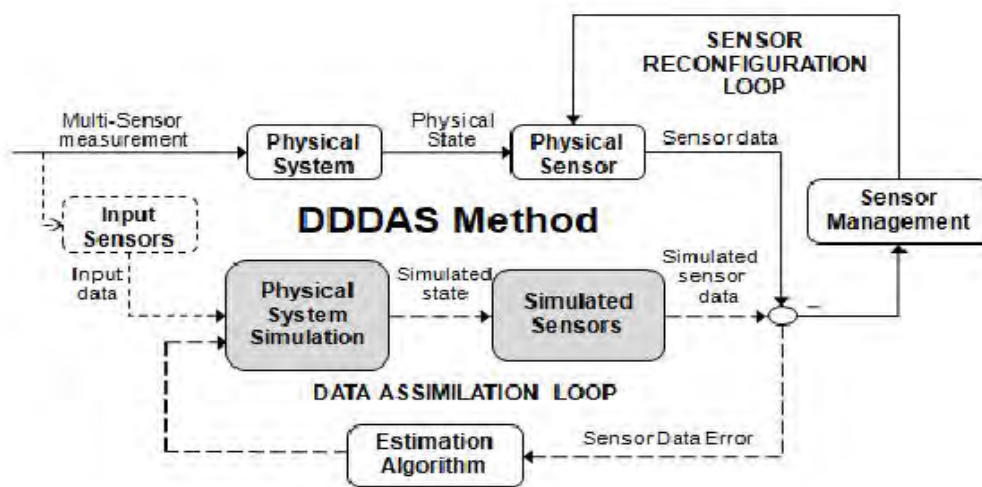


Рисунок 9.7 — Архитектура DDAS: два цикла обратной связи — управление физической системой и улучшение модели (Источник: [4])

Первый цикл, называемый циклом управления (Sensor Reconfiguration Loop), направлен на управление реальной физической системой: данные с датчиков передаются в модель, на их основе формируется управляющее воздействие, которое применяется к реальной системе через актуаторы. Второй цикл — цикл усвоения данных (Data Assimilation Loop) — служит для непрерывного улучшения самой модели: поступающие оперативные данные используются для коррекции состояния модели (усвоение), перекалибровки её

параметров и, при необходимости, перезапуска. Симбиоз этих двух циклов позволяет создавать высокоадаптивные системы, работающие в режиме реального времени.

DDAS находит применение в широчайшем круге задач: управление автономными роботами (где одновременно управляются реальные роботы и моделируются их виртуальные двойники), мониторинг и прогнозирование лесных пожаров, управление энергетическими сетями, тактическое управление в военных операциях, предсказание распространения загрязнений в атмосфере и водных объектах. Ключевой особенностью DDAS является то, что оба цикла могут разрабатываться и тестироваться относительно независимо: цикл управления — на основе имитационных экспериментов, цикл усвоения — на исторических данных. Такое разделение существенно снижает сложность разработки.

9.7. Верификация и валидация агентных моделей

Надёжность агентной модели определяется строгостью двух взаимосвязанных, но принципиально различных процессов: верификации и валидации. Эти понятия нередко смешиваются, однако их чёткое разграничение критически важно для методологической корректности любого исследования (рисунок 9.8).

Верификация (Verification) — это процесс проверки того, что компьютерная (имитационная) модель корректно воплощает задуманную концептуальную модель. Верификация отвечает на вопрос: «Правильно ли мы построили модель?» — то есть соответствует ли программный код тому, что было задумано. По своей природе верификация близка к верификации программного обеспечения и использует аналогичный инструментарий: пошаговую отладку (debugging), ревью кода, анализ логов выполнения, модульное тестирование и визуализацию промежуточных состояний. Важно, однако, помнить, что верификация модели и верификация программы — связанные, но не тождественные понятия: верификация модели всегда ориентирована на концептуальное соответствие предметной постановке задачи.

Валидация (Validation) — это проверка того, что модель адекватно описывает целевую реальную систему. Валидация отвечает на вопрос: «Правильную ли модель мы построили?» В классификации Сарджента [1] выделяются несколько уровней и типов валидации.

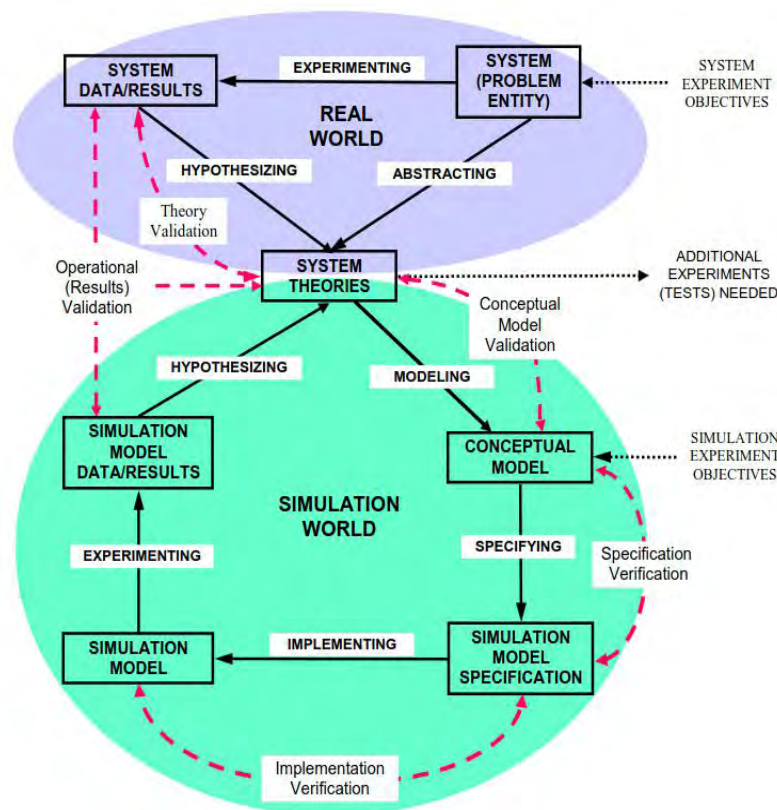


Рисунок 9.8 — Взаимодействие реального мира и мира моделирования в контексте верификации и валидации (Источник: [1])

Construct validity (валидность конструкта / структурная валидность) оценивает, насколько структура модели — взаимосвязи между агентами, правила поведения, топология среды — соответствует реальным механизмам, происходящим в исследуемой системе. Это, по существу, валидность концептуальной модели: правильно ли мы поняли систему ещё до того, как написали код?

Content validity (содержательная валидность) оценивает полноту модели: охватывает ли она все существенные варианты поведения системы и все значимые режимы её работы? Высокая содержательная валидность означает, что модель «покрывает» пространство возможных сценариев достаточно широко, чтобы делать обоснованные выводы.

Face validity (операциональная валидность) — наиболее распространённый тип проверки. Он состоит в сопоставлении выходных данных модели с реальным поведением системы. Это сопоставление может быть как количественным (сравнение временных рядов, статистических характеристик), так и качественным (экспертная оценка правдоподобия динамики). Примечательно, что face validity применима даже к абстрактным моделям: если мы наблюдаем в модели эмерджентную сегрегацию, соответствующую мотивирующим нас

идеям о реальном явлении, это уже является определённой операциональной валидацией.

Criterion validity (критериальная валидность) — формализованное сравнение результатов модели с реальными данными по конкретным количественным критериям: показателям точности, полноты, F-мерам или иным метрикам. Этот тип валидации ближе всего к тому, что в машинном обучении называют оценкой качества модели на тестовой выборке.

Помимо этих четырёх типов, рассматриваются и дополнительные виды валидации. **Predictive validity** (предсказательная валидность) оценивает способность модели воспроизводить ненаблюдаемые состояния — необязательно в будущем, но нередко и в настоящем. Например, при моделировании состояния пациента мы можем «предсказывать» его параметры между моментами измерения. **Internal validity** (внутренняя валидность) проверяет корректность модели в рамках ограниченного контролируемого эксперимента, тогда как **external validity** (внешняя валидность) оценивает, насколько выводы, полученные в модели, обобщаются на реальную «дику» систему. Наконец, **concurrent validity** (конкурентная валидность) предполагает сравнение разрабатываемой модели с уже существующими альтернативными моделями той же системы.

9.8. Прозрачность и интерпретируемость моделей

С ростом сложности агентных моделей всё более острой становится проблема их **интерпретируемости** (interpretability) и **прозрачности** (transparency). Под интерпретируемостью понимается способность исследователя, эксперта или лица, принимающего решения, понять, почему модель ведёт себя так, а не иначе, и каким именно механизмам обязаны наблюдаемые результаты.

Интуитивно кажется, что более детальная и точная модель должна быть одновременно и более ценной, и более понятной. Однако на практике между детализацией модели и её интерпретируемостью существует, как правило, обратная зависимость. Слишком простая модель может быть понятна, но давать неверные предсказания. Слишком сложная модель — нейросетевые функции поведения агентов, огромное пространство состояний, запутанные взаимодействия — может давать точные результаты, но превращаться в «чёрный ящик», поведение которого никто не может объяснить. Наконец, само по себе эмерджентное поведение, ради наблюдения которого зачастую и строится агентная модель, является источником трудно

интерпретируемых феноменов: мы видим паттерн, но не всегда можем объяснить, из каких взаимодействий он вырастает.

Полезность и точность модели поначалу возрастают с её усложнением, однако слишком сложная модель становится как дорогой в реализации, так и непрозрачной. Интерпретируемость, в свою очередь, нередко сначала растёт (когда мы добавляем нужные детали, связывающие модель с понятными механизмами), а затем падает (когда модель становится слишком сложной для аналитического осмысления). Нахождение оптимального баланса зависит от конкретного сценария и от того, кто является конечным пользователем модели — исследователь, эксперт или лицо, принимающее решения.

Существует ещё один важный аспект: валидируемость модели может быть возможна лишь в определённом диапазоне сложности. Если мы слишком упростим модель, она перестанет воспроизводить наблюдаемые в реальной системе паттерны и станет невалидной — пусть и формально интерпретируемой. Таким образом, существует нижняя граница сложности, продиктованная требованиями валидации, и верхняя граница, продиктованная требованиями интерпретируемости. Между ними и должна находиться практически пригодная модель.

Для улучшения интерпретируемости и обеспечения прозрачности агентных моделей применяется ряд специализированных инструментов. **Визуализация** — один из наиболее мощных: поскольку человек воспринимает значительно больше информации через визуальный канал, анимированные карты перемещения агентов, графики плотности, тепловые карты и временные диаграммы позволяют интуитивно схватить сложные динамические паттерны, которые в числовом виде остались бы незамеченными. Большинство современных платформ агентного моделирования имеют встроенные механизмы визуализации.

Анализ чувствительности (Sensitivity Analysis) — инструмент, без которого любые выводы из модели остаются недостаточно обоснованными. Запустить модель один раз и сделать вывод из единственного прогона — методологически некорректно: случайные флуктуации или конкретный набор начальных условий могут полностью определить наблюдаемое поведение. Систематическая вариация параметров (в том числе с учётом их реальной неопределённости, связанной с ошибками измерений) позволяет: выявить наиболее влиятельные параметры; оценить устойчивость выводов к ошибкам в данных; исследовать пространство возможных поведений системы.

Управление неопределённостью (Uncertainty Management) предполагает явное включение неопределённости в цикл моделирования. Вместо одного «точечного» прогноза строится распределение возможных результатов, соответствующее диапазону неопределённости в данных и параметрах модели. Это позволяет честно сообщать не только о предсказании, но и о его надёжности.

Особого внимания заслуживает редукционизм в агентном моделировании. Упрощая модель ради интерпретируемости, необходимо следить за тем, чтобы не потерять воспроизводимость ключевых феноменов. В частности, эмерджентное поведение нередко наблюдается только при достаточно большом числе агентов: при слишком малом числе агентов статистически значимые коллективные паттерны попросту не успевают сформироваться. Таким образом, масштаб системы является самостоятельным параметром качества агентной модели, который нельзя произвольно редуцировать.

9.9. Заключение

Моделирование поведения агентов реального мира — это комплексная методологическая задача, требующая согласованного решения на нескольких уровнях. Во-первых, необходимо грамотно выстроить концептуальный фреймворк: чётко разграничить концептуальную и имитационную модели, определить место данного исследования на шкале от «абстрактных» до «data-driven» агентных моделей и обеспечить двунаправленную связь между моделью и реальной системой.

Во-вторых, ключевым ресурсом для построения реалистичных моделей являются данные. Разнородность и неидеальность реальных данных требуют использования широкого арсенала методов: от process mining и скрытых марковских моделей до машинного обучения и фильтрации частиц. Концепции усвоения данных и DDDAS открывают возможность для создания адаптивных систем, непрерывно обновляющих свои модели в режиме реального времени.

В-третьих, необходим строгий контроль качества модели. Верификация и валидация — в их разнообразных формах (construct, content, face, criterion, predictive, internal/external, concurrent) — должны пронизывать весь жизненный цикл модели, а не быть разовым финальным этапом.

Наконец, интерпретируемость и прозрачность являются не факультативными характеристиками, а неотъемлемым атрибутом научной и практической ценности модели. Поиск баланса между

сложностью, точностью и интерпретируемостью, с использованием инструментов визуализации, анализа чувствительности и управления неопределённостью, представляет собой центральную методологическую задачу при создании любой агентной модели реального мира.

Рекомендуемая литература и источники

1. Sargent R.G. Verification and Validation of Simulation Models / R.G. Sargent // Proceedings of the 2010 Winter Simulation Conference. — Baltimore, 2010. — P. 166–183. — DOI: 10.1109/WSC.2010.5679166. — URL: <https://doi.org/10.1109/WSC.2010.5679166>
2. Smajgl A. Behaviour and Space in Agent-Based Modelling: Poverty Traps and Land Use in a Rural Indonesian Setting / A. Smajgl, E. Bohensky // Environmental Modelling & Software. — 2013. — Vol. 45. — P. 8–15. — DOI: 10.1016/j.envsoft.2012.08.021. — URL: <https://doi.org/10.1016/j.envsoft.2012.08.021>
3. Robinson S. Conceptual Modelling for Simulation / S. Robinson. — London : Palgrave Macmillan, 2008. — 283 p. — ISBN 978-0-230-53602-7.
4. Darema F. Dynamic Data-Driven Applications Systems: A New Paradigm for Application Simulations and Measurements / F. Darema // Proceedings of the International Conference on Computational Science (ICCS 2004). — Kraków : Springer, 2004. — P. 662–669. — DOI: 10.1007/978-3-540-24688-6_86. — URL: https://doi.org/10.1007/978-3-540-24688-6_86
5. Bijak J. Towards Empirical Agent-Based Modelling of Social Dynamics / J. Bijak [et al.] // Land. — 2023. — Vol. 12, No. 4. — Art. 756. — DOI: 10.3390/land12040756. — URL: <https://doi.org/10.3390/land12040756>
6. Zeigler B.P. Theory of Modeling and Simulation / B.P. Zeigler, H. Praehofer, T.G. Kim. — 2nd ed. — San Diego : Academic Press, 2000. — 510 p. — ISBN 978-0-12-778455-7.
7. Zia K. Particle Filter Based Localization and Tracking in Agent-Based Simulations / K. Zia, A. Ferscha // Simulation Modelling Practice and Theory. — 2015. — Vol. 55. — P. 100–119. — DOI: 10.1016/j.simpat.2015.05.001. — URL: <http://dx.doi.org/10.1016/j.simpat.2015.05.001>
8. Borshchev A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools / A. Borshchev, A. Filippov // Proceedings of the 22nd International Conference of the System Dynamics Society. — 2004. — URL: <https://www.anylogic.com/resources/articles/from-system-dynamics-and-discrete-event-to-practical-agent-based-modeling-reasons-techniques-tools/>
9. Wang M. Data Assimilation in Agent Based Simulation of Smart Environments Using Particle Filters / M. Wang, X. Hu // Simulation Modelling Practice and Theory. — 2015. — Vol. 56. — P. 36–54. — DOI: 10.1016/j.simpat.2015.05.001. — URL: <https://doi.org/10.1016/j.simpat.2015.05.001>
10. Ahmad L. Particle Filter Data Assimilation for Reconstructing Latent Hospital Care Pathways under Partial Observability / L. Ahmad, S.V. Kovalchuk // Lecture Notes in Computer Science. — Cham : Springer, 2026.

Лекция 10. Когнитивные состояния: модели и архитектуры

В настоящей лекции рассматриваются подходы к формальному описанию когнитивных состояний интеллектуальных агентов: от оценки уровней интеллекта и таксономии когнитивных архитектур до конкретных моделей — BDI, Theory of Mind, клонирования поведения и обратного обучения с подкреплением. Особое внимание уделяется связи между рациональными и иррациональными паттернами поведения реальных людей и их формальным представлением в агентных системах.

10.1. Искусственный интеллект общего назначения и проблема оценки интеллекта

Моделирование поведения агентов, способных адекватно воспроизводить поведение человека, требует перехода от простых реактивных правил к полноценному описанию сложных когнитивных процессов. Такой агент должен не только реагировать на стимулы среды, но и демонстрировать способности к планированию, обучению, коммуникации и решению нестандартных задач. Этот запрос прямо формулируется в области **искусственного интеллекта общего назначения** (*Artificial General Intelligence, AGI*) — создания систем, способных решать широкий класс задач так, как это делает человек, без заранее заданной узкой специализации.

Perception	Memory	Attention	Social Interaction	Planning	Motivation
Vision	Working	Visual	Communication	Tactical	Subgoal creation
Smell	Episodic	Auditory	Appropriateness	Strategic	Affect-based
Touch	Implicit	Social	Social Inference	Physical	Deferred Gratification
Taste	Semantic	Behavioral	Cooperation	Social	Altruism
Audition	Procedural	Communication	Competition	Emotion	Modeling Self / Other
Cross-modal			Relationships		
Proprioceptive	Reasoning	Gestural	Learning	Perceived	Self-Awareness
	Induction	Verbal		Expressed	Other-Awareness
Actuation	Deduction	Musical	Imitation	Control	Relationships
Physical Skills	Abduction	Pictorial	Reinforcement	Understanding	Self-Control
Tool Use	Physical	Diagrammatical	Dialogical	Sympathy	Theory of Mind
Navigation	Causal	Language Acquisition	Media-Oriented	Empathy	Sympathy
Proprioceptive	Associational	Cross-Modal	Experimentation		Empathy
Building/Creation			Quantitative		
Physical Construction with Objects			Counting Observed Entities		
Formation of Novel Concepts			Grounded Small Number Arithmetic		
Verbal Invention			Comparison of Quantitative Properties of Observed Entities		
Social Organization			Measurement Using Simple Tools		

Рисунок 10.1 — Компетенции AGI (Источник: [1])

Концепция AGI восходит к самым ранним работам в области кибернетики и информатики, однако активное её развитие приходится на период после 2000-х годов, когда стремительный рост вычислительных мощностей сделал её достижение реалистичной долгосрочной целью. Сегодня крупнейшие исследовательские лаборатории — OpenAI, DeepMind, Meta AI Research и другие — формулируют создание AGI как одну из своих основных задач.

Свойства AGI. По общепринятым представлениям [1], система, претендующая на статус AGI, должна обладать рядом ключевых способностей: принимать решения в условиях неопределённости и использовать стратегии решения задач; формировать и применять знания, включая обобщённую модель реальности; планировать последовательности действий для достижения целей; непрерывно обучаться на основе опыта; общаться на естественном языке; наконец, интегрировать все эти способности для решения комплексных задач произвольной природы.

Тест Тьюринга и другие подходы к оценке интеллекта. Одной из ключевых проблем AGI является вопрос о том, как измерить уровень интеллекта системы. Наиболее известным инструментом остаётся **тест Тьюринга**, предложенный Аланом Тьюрингом в 1950 году: человек-оценщик вступает в диалог с двумя собеседниками — человеком и машиной — и пытается определить, кто из них кто. Если машина систематически вводит оценщика в заблуждение, она считается прошедшей тест. Однако этот подход давно подвергается критике: он оценивает лишь способность имитировать, а не наличие подлинного понимания.

Более современным и многомерным является «**тест Возняка**», неформально сформулированный Стивом Возняком: он поверит в AGI лишь тогда, когда система сможет войти в незнакомый дом и самостоятельно приготовить чашку кофе. Это задание кажется бытовым, однако требует построения полной цепочки рассуждений: понять, что кофе варят с помощью определённых устройств, которые обычно находятся на кухне, локализовать кухню в незнакомой планировке, составить план действий и реализовать его. Взрослый человек справляется с этим легко, опираясь на неявные знания о мире — именно это и недоступно современным системам AI.

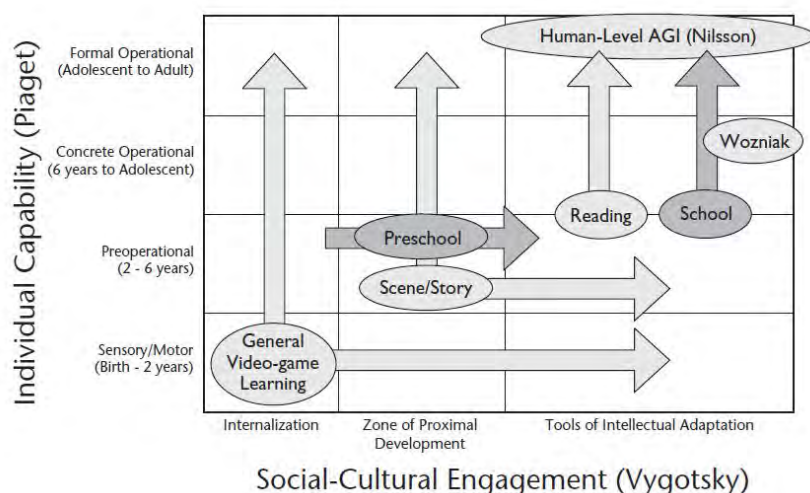


Рисунок 10.2 — CompetenceScale для оценки интеллекта агента
(Источник: [1])

Систематизацию тестов и критериев интеллекта предложила **шкала CompetenceScale** [1]: она раскладывает возможности агента по двум осям — сложности индивидуальных способностей и развитости социокультурной адаптации — и соотносит их с этапами когнитивного развития человека от младенчества до зрелости (рисунок 10.2).

Эффект Даннинга-Крюгера применительно к современным языковым моделям. Появление больших языковых моделей (Large Language Models, LLM), прежде всего ChatGPT, породило волну оптимистичных оценок достижимости AGI. Тем не менее необходимо критически оценивать поверхностное правдоподобие ответов таких систем. Характерный пример: на запрос о реализации классификатора с помощью TabTransformer в TensorFlow модель генерирует развёрнутый ответ с примером кода и ссылкой на статью — однако при детальной проверке оказывается, что упомянутая статья не существует, URL ведёт в никуда, а описанный API-вызов не поддерживается ни одной из существующих версий библиотеки. Данная ситуация хорошо иллюстрирует **эффект Даннинга-Крюгера** применительно к ИИ: система, лишённая подлинного понимания, генерирует ответ, выглядящий убедительно для неподготовленного пользователя, — и именно такой пользователь с наибольшей вероятностью примет его за корректный. Проблема состоит в том, что языковые модели оптимизируются на лингвистическое правдоподобие ответа, а не на его семантическую корректность. Похожесть на правильный ответ не является правильностью, и это принципиальное ограничение необходимо учитывать при любом применении LLM в задачах, где трудно осуществить независимую проверку результата.

10.2. Уровни когнитивных способностей агентов

Попытка систематически ранжировать когнитивные способности агентов независимо от конкретных тестов привела к появлению нескольких классификационных шкал.

Таксономия BICA Society. Организация BICA Society (Biologically Inspired Cognitive Architectures) предложила иерархическую таксономию когнитивных архитектур, упорядочивающую их по возрастанию функциональной сложности. Таксономия выделяет пять ключевых уровней. Эта иерархия не случайна: она в точности соответствует этапам когнитивного развития живых существ — от рефлекторного поведения новорождённого до самоанализа и метарассуждений взрослого человека.

Шкала когнитивных способностей ConScale. Независимый подход предложен в рамках проекта ConsScale: шкала ранжирует агентов по суммарной сложности когнитивных способностей и включает одиннадцать уровней — от неуправляемого поведения (уровень 0) до сверхсознания (уровень 11). Примечательно, что в этой шкале человек помещён на уровень 10 («Human-Like»), а уровень выше него («Super-Conscious») по-прежнему остаётся гипотетическим.

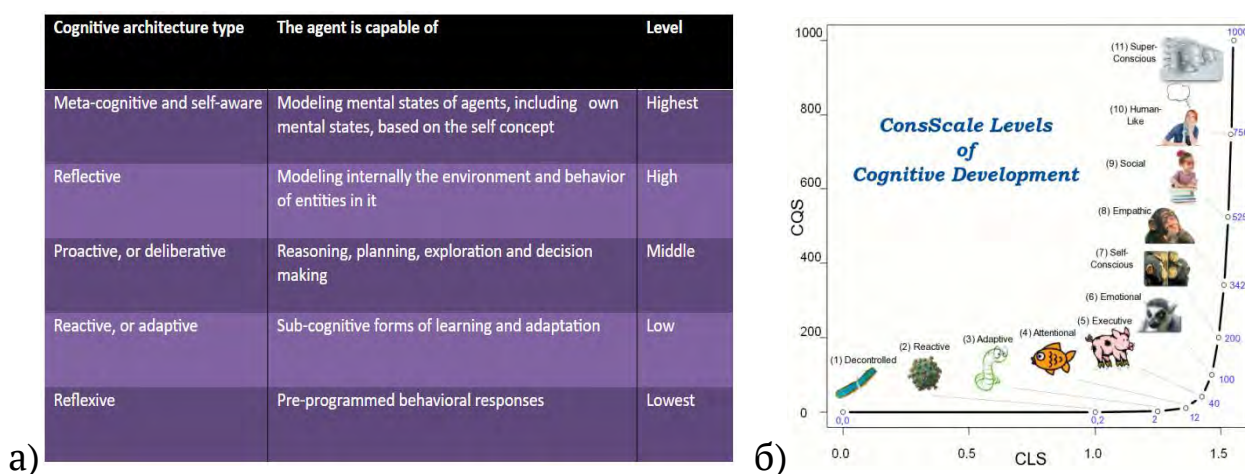


Рисунок 10.3 — Шкалы когнитивных способностей: а) BICA Society (источник: <http://bicasociety.org>); б) ConsScale (Источники: <http://www.conscious-robots.com>)

10.3. Когнитивные архитектуры

Когнитивная архитектура — это цифровая реализация когнитивных процессов в составе интеллектуальных агентов, определяющая универсальные структуры и механизмы познания в абстракции от биологической реализации мозга. В отличие от узкоспециализированных алгоритмов, когнитивные архитектуры

претендуют на воспроизведение *общих* принципов разумного поведения — восприятия, памяти, внимания, обучения, планирования и принятия решений.

Цели, которые ставят перед собой разработчики когнитивных архитектур, включают: реализацию процессов познания и воспроизведение поведения человека; обеспечение надёжного, но при этом адаптивного поведения в непредсказуемых средах; поддержку непрерывного обучения; а также моделирование высокоуровневых когнитивных способностей, включая креативность и мета-рассуждение. Исторически развитие когнитивных архитектур тесно переплетено с историей искусственного интеллекта в целом. Существующие когнитивные архитектуры принято делить на три большие класса [2].

Символьные (Symbolic) архитектуры исходят из постулата о том, что мышление есть манипуляция символами по правилам формальной логики. Они позволяют явно представлять знания, строить доказательства и осуществлять дедуктивный вывод. Расцвет символьных архитектур пришёлся на 1980-е годы — эпоху экспертных систем. Наиболее известные представители — **SOAR** и **ACT-R**. Сильная сторона этих систем — интерпретируемость и предсказуемость поведения; слабая — трудоёмкость формализации знаний и ограниченная способность работать с неопределённостью.

Эмерджентные (Emergent) архитектуры придерживаются противоположной стратегии: интеллектуальное поведение не задаётся явно, а возникает из взаимодействия простых вычислительных элементов — нейронов, агентов нижнего уровня и т. д. Характерный современный пример — **Darwinian Neurodynamics**, сочетающая нейронные сети с эволюционными алгоритмами. Эмерджентные архитектуры лучше справляются с непредсказуемыми ситуациями и потенциально способны к решению творческих задач, однако их поведение значительно труднее интерпретировать.

Гибридные (Hybrid) архитектуры представляют собой наиболее перспективный и распространённый класс. Они объединяют символьную дедукцию с эмерджентными индуктивными механизмами, стремясь воспроизвести все три формы логического вывода, присущие человеку: дедукцию, индукцию и абдукцию. Марвин Минский в книге «The Emotion Machine» (2006) высказывал убеждение, что именно через эмерджентные свойства — а не через прямую нисходящую декомпозицию — в итоге удастся описать человеческую когнитивность.

Требования к AGI-архитектуре. В статье [1] сформулированы детальные требования к когнитивной архитектуре, претендующей на реализацию AGI. Они разделены на две группы.

Требования к архитектуре агента (R0–R11). Архитектура должна работать без перепрограммирования под новые задачи (R0); поддерживать символьные вычисления (R1) и мультимодальное представление знаний (R2); оперировать большими массивами знаний разной степени общности (R3–R5); поддерживать непротиворечивые убеждения и механизмы их пересмотра (R6); обеспечивать иерархическое управление вниманием и приоритетами (R7); обладать мета-когнитивными способностями — рассуждением о собственных возможностях и знаниях (R8); учитывать ограничения ресурсов и времени (R9); поддерживать разнообразные парадигмы обучения (R10) и непрерывное итеративное накопление знаний без разрушения уже усвоенного (R11).

Требования к среде и задачам (C1–C8). Агент должен функционировать в сложной, разнородной (C1) и динамичной, открытой (C2) среде с многомасштабной изменчивостью (C3), взаимодействовать с другими агентами (C4), решать новые разнородные задачи (C5) в условиях частичного доступа к информации (C6), ограниченных вычислительных ресурсов (C7) и на протяжении длительного времени (C8).

Когнитивные архитектуры и решение творческих задач. Одним из наиболее сложных когнитивных рубежей является моделирование **творческого решения задач**. С психологической точки зрения, оно подразумевает генерацию принципиально новых идей, отвечающих требованиям беглости (скорости генерации), оригинальности, гибкости, восприимчивости к проблеме и удовлетворённости решением. Практические методологии — такие как ТРИЗ (Теория решения изобретательских задач), Синектика, метод *DO IT* — описывают процесс творческого мышления, но в целом остаются неформальными и высокоабстрактными.

Наглядным примером служит классическая **задача садовника**: посадить четыре дерева так, чтобы расстояние между каждой парой было одинаковым. Большинство людей начинает с двумерного представления сада и пытается расположить деревья в углах квадрата или ромба — и немедленно сталкивается с трудностью: диагонали квадрата длиннее его сторон. Решение требует *озарения* — осознания, что задача не ограничивает нас двумерным пространством, и деревья можно разместить в вершинах правильного тетраэдра (например, три

на плоской поверхности, одно — на холме). Это решение иллюстрирует механизм когнитивного ограничения: человек редуцирует задачу до знакомой модели, и именно эта редукция блокирует верный ответ. С точки зрения формализации в когнитивных архитектурах, подобные задачи особенно интересны: если архитектура не опирается на явное кодирование предположений об ограничениях задачи, у неё принципиально больше шансов найти нетривиальное решение через исследование пространства состояний.

Символьная архитектура **Carion** воспроизводит процесс «вынашивания» проблемы: она чередует явную (логическую) и неявную (ассоциативную) фазы обработки, что позволяет имитировать момент инсайта. Эмерджентная архитектура **Darwinian Neurodynamics** применяет эволюционные алгоритмы к популяциям нейросетей, порождая разнообразные варианты решений. Пример из области игр — **AlphaGo**: система демонстрировала ходы, расцениваемые профессиональными игроками как «интуитивные» и нестандартные — продукт обучения с подкреплением, а не символьного программирования.

Вместе с тем необходимо сохранять критическое отношение к утверждениям о «творчестве» современных языковых моделей: их генерация во многом комбинаторна и опирается на паттерны, усвоенные при обучении на данных. Можно предположить, что «творчество» в этом случае рождается не в самой модели, а в восприятии наблюдателя, обнаруживающего неожиданные сочетания.

10.4. Взаимодействие человека и агента ИИ

Исторически агенты искусственного и естественного интеллекта противопоставлялись в рамках тестов (как в тесте Тьюринга). Однако всё более перспективной парадигмой становится не противопоставление, а **взаимодополнение** — идея *гибридного* или *дополненного* интеллекта (Augmented Intelligence), при которой человек и ИИ работают совместно, усиливая возможности друг друга.

В этой парадигме различают два принципиально разных режима взаимодействия. В режиме **Human-in-the-Loop** человек остаётся непосредственным участником процесса: он вводит правила, размечает данные, валидирует результаты и при необходимости объясняет своё решение. Итогом является расширение возможностей оператора. В режиме **Cognitive Computing** система функционирует автономнее: она контролирует процесс обучения самостоятельно, берётся за сложноформализуемые задачи и требует лишь периодической

валидации. Такой подход открывает доступ к решению задач, требующих высокого уровня когнитивных способностей.

Принципиальная проблема в обоих случаях — как организовать взаимодействие человека и ИИ так, чтобы система в целом работала лучше каждого из компонентов по отдельности. Современные подходы обычно строятся по схеме наблюдателя: ИИ наблюдает за действиями человека и адаптируется к ним. Однако, значительно более продуктивным направлением представляется организация *коэволюции* агентов — совместного выстраивания ими нового способа взаимодействия с предметной областью.

10.5. Клонирование поведения и обратное обучение с подкреплением

Клонирование поведения (*Behavioral Cloning, BC*; также называемое *apprenticeship learning, learning by demonstration, learning by imitation*) — класс методов, при которых агент обучается вести себя так же, как наблюдаемый эксперт-человек, путём извлечения стратегии поведения непосредственно из записей его действий.

Разница в терминологии отражает, как правило, не столько различие в механизме, сколько различие в сценарии применения: управление техническими системами, обучение роботов, взаимодействие человека с компьютером, обучение игровых агентов и т. д. Принципиальная логика остаётся единой: наблюдаем пары $\langle \text{состояние среды, действие эксперта} \rangle$, на их основе строим контроллер агента.

Прямое клонирование (*direct BC*) реализует прямое отображение «состояние $\rightarrow$ действие»: агент при каждом наблюдаемом состоянии среды выдаёт то же действие, что и эксперт. Это наиболее простой подход, ограниченный, однако, тем, что контроллер агента и контроллер человека должны быть эквивалентны. Когда это не так — например, робот управляет автомобилем принципиально иным физическим способом, чем человек, — прямое отображение неприменимо.

Непрямое клонирование (*indirect BC*) предполагает декомпозицию задачи: задача или цель разбивается на подзадачи, каждая из которых управляется отдельным контроллером меньшего масштаба. Этот подход расширяет применимость методологии, но усложняет как процесс обучения, так и последующее управление.

Алгоритмически реализация клонирования может опираться на правила, деревья принятия решений, регрессионные модели (включая локально-взвешенную регрессию) и методы машинного обучения. Несмотря на разнообразие инструментов, все они в конечном счёте решают одну и ту же задачу: найти наилучшую стратегию поведения (*policy*), отвечающую наблюдаемой истории действий эксперта. Это в точности соответствует задаче обучения с подкреплением — с той разницей, что мы учимся не на собственном опыте агента, а на чужой истории.

Обратное обучение с подкреплением. В классическом обучении с подкреплением (*Reinforcement Learning, RL*) агент знает функцию вознаграждения и ищет оптимальную стратегию (*policy*). **Обратное обучение с подкреплением** (*Inverse Reinforcement Learning, IRL*) решает обратную задачу: наблюдая за поведением эксперта, восстановить функцию вознаграждения $R(s)$, которую он, по всей видимости, максимизирует.

Восстановление функции вознаграждения представляет самостоятельный практический интерес: она может быть использована для дальнейшего обучения агентов, для анализа того, как эксперты принимают решения в сложных ситуациях, и для построения более интерпретируемых моделей поведения.

Задача IRL сопряжена с рядом фундаментальных трудностей. Во-первых, существует **неоднозначность функции вознаграждения**: одно и то же наблюдаемое поведение может порождаться бесконечно многими различными функциями R . Во-вторых, возникают статистическая и вычислительная сложности, напрямую зависящие от размерности пространства состояний. В-третьих, большинство методов предполагает, что стратегия π эксперта нам известна, тогда как на практике мы наблюдаем лишь её проявления.

Первые систематические подходы к IRL предложили Ng и Russell [4], формализовав задачу как линейную программу. Abbeel и Ng (2004) развили эту линию в *apprenticeship learning*. Сегодня наиболее популярным является метод **Maximum Entropy IRL** [5]: вместо того чтобы выбирать единственную функцию вознаграждения, он выбирает распределение поведений с максимальной энтропией, не противоречащее наблюдаемым данным. Это делает решение устойчивым и хорошо обобщающимся. Современные глубокие реализации IRL, такие как **DeepIRL** с максимизацией энтропии, используют нейронные сети для аппроксимации функции вознаграждения непосредственно из признаков состояния: нейросеть

«переваривает» карту признаков среды, граф переходов между состояниями и выдаёт набор компонент вознаграждения R(s).

10.6. Модель BDI и Theory of Mind

Если задача IRL — восстановить функцию вознаграждения агента из наблюдений, то для полноценного моделирования поведения человека необходимо учитывать и структуру его внутренней «ментальной среды». Классическим инструментом здесь служит модель **BDI** (*Beliefs, Desires, Intentions*), предложенная на основе теории практического рассуждения философа Майкла Братмана и формализованная в рамках ИИ Михаэлем Дживесом и Джаспером Нидером в конце 1980-х годов.

Модель включает три компонента. **Beliefs (убеждения)** — субъективные представления агента об окружающей среде, которые могут не совпадать с объективным состоянием мира. **Desires (желания)** — цели агента: результаты, состояния или ситуации, которые он стремится достичь. **Intentions (намерения)** — конкретные планируемые действия, выбранные для продвижения к желаемому состоянию; они отличаются от желаний тем, что агент берёт на себя обязательство их исполнить.

Рассуждение в BDI-агенте организовано в два этапа: сначала из убеждений и желаний генерируются *опции* (Options), затем из опций выбираются намерения (через *фильтр*). Ключевое достоинство BDI — простота понятийного аппарата в сочетании с богатством реализаций. Среди практических реализаций можно выделить системы PRS (*Procedural Reasoning System*), JACK Intelligent Agents, Jason (открытый исходный код), AgentSpeak(L) и многие другие — гибридные системы на основе BDI насчитывают десятки реализаций.

Важна и соответствие BDI формализму обучения с подкреплением: *Beliefs* соответствуют состоянию мира (*state*), *Desires* — функции вознаграждения (*reward*), *Intentions* — стратегии поведения (*policy*). Это позволяет естественным образом интегрировать BDI с методами IRL.

Theory of Mind (ToM), или «теория разума», — концепция, пришедшая из когнитивной психологии и описывающая способность приписывать себе и другим людям ментальные состояния: убеждения, желания, намерения, знания и эмоции. Принципиально, что эти приписываемые состояния могут расходиться с объективной реальностью.

Классическим экспериментальным методом проверки наличия ТоМ является задача Салли и Энн [9]. Участнику показывают

следующую сценку: Салли кладёт мячик в корзинку и уходит; Энн перекладывает мячик из корзинки в коробку; Салли возвращается. Вопрос: *где Салли будет искать мячик?* Дети до 4 лет, как правило, отвечают «в коробке» — там, где мячик находится на самом деле. Дети старше 4 лет понимают, что Салли не видела перекладки и потому будет искать в корзинке, то есть там, где, *по её убеждению*, должен лежать мячик. Именно этот переход — от описания мира «как он есть» к рассуждению о мире «как его видит другой» — знаменует появление ТоМ.

С вычислительной точки зрения, ТоМ можно представить как иерархию убеждений. Агент уровня 0 действует в среде без учёта ментальных состояний других («мы действуем в безличной среде»). Агент уровня 1 учитывает убеждения партнёра («какой тип поведения у нашего партнёра?»). Агент уровня 2 рассуждает о том, как партнёр воспринимает нас («что они думают, что мы думаем о них?»). Теоретически уровень вложенности не ограничен, однако на практике двух уровней, как правило, достаточно для реалистичного моделирования социального поведения.

Важна связь ТоМ с IRL: в то время как стандартный IRL предполагает известность стратегии π и восстанавливает лишь функцию вознаграждения R , IRL, расширенное с использованием ТоМ, одновременно моделирует и π , и R другого агента, по сути, реализуя рассуждение о намерениях наблюдаемого субъекта.

Строгий математический формализм, объединяющий BDI и ТоМ, предоставляет **POMDP** (*Partially Observable Markov Decision Process* — частично наблюдаемый марковский процесс принятия решений). В отличие от полностью наблюдаемого MDP, агент в POMDP не имеет прямого доступа к истинному состоянию среды s , а получает лишь зашумлённые наблюдения o , распределённые согласно функции наблюдения $Z(o | s', a)$.

Ключевым понятием POMDP является **состояние верования** (*belief state*, b) — вероятностное распределение над всеми возможными состояниями среды. Это распределение обновляется байесовски при получении каждого нового наблюдения. Именно *belief state* является «достаточной статистикой» всей истории взаимодействия агента со средой и служит основой для принятия решений: оптимальная политика $\pi(b)$ отображает состояние верования в действие.

Соответствие между POMDP и BDI очевидно: *Beliefs* в BDI формализуются как *belief state*; *Intentions* соответствуют оптимальной политике $\pi(b)$; *Desires* закодированы в функции вознаграждения R .

Иерархический POMDP позволяет формализовать и Theory of Mind: каждый уровень вложенности соответствует отдельному POMDP, моделирующему убеждения другого агента о состоянии мира.

Пример: игра Hanabi. Особенно ярко связка ToM–BDI–IRL проявляется в кооперативных играх с неполной информацией. Японская карточная игра **Hanabi** — идеальный полигон [12]: игроки не видят собственных карт, но видят карты партнёров и должны составить масти по порядку. Подсказки передаются строго по установленным правилам. Успех в игре прямо зависит от способности каждого игрока рассуждать о том, что думает партнёр, опираясь на его ходы и подсказки.

Формальный агент в Hanabi поддерживает *belief state* о своих закрытых картах, рассуждает о действиях партнёра с позиций ToM первого уровня («Боб видит мои карты и карты Кэти; когда он говорит "белый", он наверняка имеет в виду вот эти карты в моей руке»), и, применяя IRL, восстанавливает *reward* партнёра — его неявную стратегию подсказок. Именно эта связка позволяет агентам на основе POMDP/BDI/ToM играть в Hanabi на уровне, сопоставимом с человеческим, не имея доступа к явным внутренним состояниям партнёра.

10.7. Моделирование иррационального поведения

Иррациональность как субъективная рациональность.

Стандартные экономические и агентные модели предполагают рациональность: агент максимизирует некую объективную функцию полезности. Реальное поведение людей систематически отклоняется от этого идеала, что принято называть **КОГНИТИВНЫМИ ИСКАЖЕНИЯМИ** (*cognitive biases*). Однако важно понять: означает ли это, что люди «нерациональны», или что они рациональны, но оптимизируют иную — субъективную и контекстно-зависимую — функцию вознаграждения?

Поведенческий экономист Дэн Ариэли в книге «Predictably Irrational» [10] показывает: большинство когнитивных искажений предсказуемы и воспроизводимы. Это означает, что они не случайны, а системны — то есть поддаются формализации. Следовательно, «иррациональное» поведение реального человека вполне может быть описано в терминах функции вознаграждения с субъективными, контекстно-зависимыми компонентами.

Пример: Эффект приманки (Decoy Effect). Ариэли описывает подписку на журнал *The Economist* с тремя вариантами: онлайн-доступ за 59 \$, печатная версия за 125 \$, печатная + онлайн за 125 \$. Третий

вариант явно доминирует второй, поэтому большинство выбирает именно его — хотя большинству нужна была только онлайн-подписка, стоящая вдвое дешевле. Механизм этого явления — **локальное сопоставление альтернатив**: если среди вариантов выбора присутствует «приманка» (decoy) — вариант А–, чуть хуже варианта А, но лучше варианта В по какому-либо признаку, — люди склонны предпочесть А, даже если изначально они были безразличны между А и В или склонялись к В. В терминах теории выбора, наблюдаемое поведение нарушает принцип независимости нерелевантных альтернатив (IIA), однако описывается как рациональное в расширенной модели оценки, допускающей зависимость предпочтений от контекста.

Пример: Эффект якоря (Anchoring) и импринтинг. В другом эксперименте участники записывали последние две цифры номера своей социальной страховки, после чего оценивали, готовы ли они купить предложенные товары за соответствующую цену. Далее их просили назвать максимальную цену, которую они готовы заплатить. Как показал анализ данных, называемые максимальные цены значимо коррелировали с начальным случайным числом: участники с большим «якорем» называли цены систематически выше. Механизм **импринтинга**, или «эффекта якоря», состоит в том, что первоначально воспринятое значение — даже заведомо случайное и нерелевантное — смещает последующие суждения в свою сторону. Это поведение нерационально с точки зрения объективной функции полезности, но полностью описывается в рамках модели ограниченной рациональности, в которой функция вознаграждения зависит от предыстории наблюдений и доступной рабочей памяти.

Систематизация когнитивных искажений в моделях агентов. Существует целый спектр феноменов иррационального поведения, поддающихся систематическому описанию. Ключевые из них — когнитивная предвзятость (*cognitive bias*), эффект стадности (*herding effect*), иррациональное изобилие (*irrational exuberance*), ошибка невозвратных затрат (*sunk cost fallacy*), склонность к контролю (*lack of control*), предвзятость настоящего (*present bias*) и дискриминация. Для агентного моделирования это означает следующее: если мы хотим, чтобы агент воспроизводил реалистичное поведение человека, нам недостаточно задать объективную функцию полезности. Необходимо включить в модель субъективные компоненты оценки: зависимость от контекста, ограниченность рабочей памяти и влияние предшествующего опыта. Тогда многие «иррациональные» паттерны

превращаются в квазиоптимальные — оптимальные с точки зрения субъективной функции вознаграждения конкретного агента.

10.8. Заключение

Лекция 10 демонстрирует, как широкий спектр методов и концепций — от оценки уровней интеллекта и таксономии когнитивных архитектур до конкретных вычислительных моделей — складывается в целостный инструментарий для описания когнитивных состояний агентов.

Когнитивные архитектуры предоставляют структурированное функциональное описание логики рассуждений агентов, позволяя перейти от биологических метафор к вычислительным моделям. При этом гибридные архитектуры, объединяющие символьные и эмерджентные подходы, сегодня наиболее перспективны для воспроизведения широкого диапазона когнитивных способностей — от логического вывода до творческих решений.

Для вывода стратегии поведения из наблюдений ключевую роль играют методы клонирования поведения (behavioral cloning) и обратного обучения с подкреплением (IRL). Они обеспечивают мощный и принципиально важный инструмент: позволяют автоматически извлекать не только действия, но и неявные цели наблюдаемых агентов.

Модели BDI и Theory of Mind описывают внутренние состояния агентов и их способность рассуждать о рассуждениях других — тем самым открывая путь к социально адекватному поведению в многоагентных и человеко-машинных системах. Строгое математическое воплощение этих концепций обеспечивает POMDP, связывающий частично наблюдаемую среду с байесовским обновлением убеждений и оптимальной политикой.

Наконец, принципиально важно учитывать субъективность и ограниченность реального человеческого познания: «иррациональное» поведение в большинстве случаев является рациональным в рамках индивидуальной, контекстно-зависимой модели вознаграждения. Включение этого понимания в агентные модели критически необходимо для их реалистичности. Открытыми остаются вопросы о глубине вложенности рассуждений в ТоМ, механизмах абдуктивного вывода и о том, каким образом коэволюция людей и ИИ-агентов способна порождать принципиально новые паттерны взаимодействия.

Рекомендуемая литература и источники

1. Adams S. Mapping the Landscape of Human-Level Artificial General Intelligence / S. Adams [et al.] // AI Magazine. — 2012. — Vol. 33, No. 1. — DOI: 10.1609/aimag.v33i1.2322. — URL: <https://doi.org/10.1609/aimag.v33i1.2322>
2. Kotseruba I. 40 Years of Cognitive Architectures: Core Cognitive Abilities and Practical Applications / I. Kotseruba, J.K. Tsotsos // Artificial Intelligence Review. — 2020. — Vol. 53, No. 1. — P. 17–94. — DOI: 10.1007/s10462-018-9646-y. — URL: <https://doi.org/10.1007/s10462-018-9646-y>
3. Bratman M.E. Intention, Plans, and Practical Reason / M.E. Bratman. — Cambridge : Harvard University Press, 1987. — 200 p. — ISBN 978-0-674-45884-1.
4. Ng A.Y. Algorithms for Inverse Reinforcement Learning / A.Y. Ng, S.J. Russell // Proceedings of the 17th International Conference on Machine Learning (ICML 2000). — San Francisco : Morgan Kaufmann, 2000. — P. 663–670.
5. Ziebart B.D. Maximum Entropy Inverse Reinforcement Learning / B.D. Ziebart [et al.] // Proceedings of the 23rd AAAI Conference on Artificial Intelligence. — Chicago : AAAI Press, 2008. — P. 1433–1438.
6. Wulfmeier M. Maximum Entropy Deep Inverse Reinforcement Learning / M. Wulfmeier, P. Ondruska, I. Posner. — arXiv:1507.04888 [cs.LG], 2015. — DOI: 10.48550/arXiv.1507.04888. — URL: <https://arxiv.org/abs/1507.04888>
7. Baker C.L. Action Understanding as Inverse Planning / C.L. Baker [et al.] // Cognition. — 2009. — Vol. 113, No. 3. — P. 329–349. — DOI: 10.1016/j.cognition.2009.07.005. — URL: <https://doi.org/10.1016/j.cognition.2009.07.005>
8. Gmytrasiewicz P. A Framework for Sequential Planning in Multi-Agent Settings / P. Gmytrasiewicz, P. Doshi // Journal of Artificial Intelligence Research. — 2005. — Vol. 24. — P. 49–79. — DOI: 10.1613/jair.1579. — URL: <https://doi.org/10.1613/jair.1579>
9. Baron-Cohen S. Does the Autistic Child Have a «Theory of Mind»? / S. Baron-Cohen, A.M. Leslie, U. Frith // Cognition. — 1985. — Vol. 21, No. 1. — P. 37–46. — DOI: 10.1016/0010-0277(85)90022-8. — URL: [https://doi.org/10.1016/0010-0277\(85\)90022-8](https://doi.org/10.1016/0010-0277(85)90022-8)
10. Ariely D. Predictably Irrational / D. Ariely. — New York : HarperCollins, 2008. — 303 p. — ISBN 978-0-06-135323-9.
11. Minsky M. The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind / M. Minsky. — New York : Simon & Schuster, 2006. — 400 p. — ISBN 978-0-7432-7663-4.
12. Montes N. Combining Theory of Mind and Abduction for Cooperation Under Imperfect Information / N. Montes, N. Osman, C. Sierra // Multi-Agent Systems : EUMAS 2022 / ed. by D. Baumeister, J. Rothe. — Cham : Springer, 2022. — (Lecture Notes in Computer Science ; Vol. 13442). — P. 194–209. — DOI: 10.1007/978-3-031-20614-6_17. — URL: https://doi.org/10.1007/978-3-031-20614-6_17

Лекция 11. Имитационное моделирование: подходы, инструменты, данные

11.1. Терминологические основы: модели и симуляции

Имитационное моделирование занимает центральное место в арсенале методов исследования сложных систем — особенно в тех случаях, когда аналитические решения недостижимы, натурные эксперименты невозможны или крайне дорогостоящи. Прежде чем перейти к детальному рассмотрению подходов и инструментов, необходимо провести принципиальное терминологическое разграничение, хорошо устоявшееся в англоязычной научной литературе, но нередко теряющееся при переводе на русский язык.

В английском языке используются два принципиально разных понятия: **modeling** и **simulation**. *Modeling* (моделирование в узком смысле) — это процесс построения **концептуальной или математической модели**: создание абстрактного, упрощённого представления системы, фокусирующегося на её структуре, компонентах и фундаментальных принципах работы. *Simulation* же (имитационное моделирование, симуляция) — это **проведение экспериментов с моделью**, то есть имитация её поведения во времени при различных сценариях и входных параметрах. В русскоязычном контексте оба термина нередко переводятся одним словом «моделирование», что стирает важное смысловое различие между построением статического описания и динамическим процессом его «проживания» в рамках вычислительного эксперимента.

Таким образом, если *modeling* отвечает на вопрос «Как устроена система?», то *simulation* отвечает на вопрос «Как ведёт себя система со временем при тех или иных условиях?». Агентное моделирование как подход одновременно охватывает оба процесса: оно требует тщательного построения концептуальной модели агентов и среды, а затем проведения имитации их взаимодействий для получения наблюдаемых паттернов поведения.

11.2. Фундаментальный цикл моделирования и симуляции

Любое имитационное моделирование следует определённой методологической последовательности — от реальной системы до содержательных научных выводов. Эта последовательность

описывается рядом концептуальных фреймворков; одним из наиболее влиятельных среди них является фреймворк Цайглера (Zeigler) [1], который структурирует процесс моделирования и симуляции через три взаимосвязанных объекта и отношения между ними.

В центре фреймворка Цайглера находятся три объекта: **исходная система (Source System)** — реальный объект или процесс, который мы изучаем; **модель (Model)** — формализованное описание системы; и **симулятор (Simulator)** — вычислительный инструмент, исполняющий модель. Ключевыми являются два отношения: *отношение моделирования (Modeling Relation)* связывает исходную систему с моделью и отражает степень соответствия модели реальности; *отношение симуляции (Simulation Relation)* — это связь между моделью и симулятором, описывающая корректность вычислительной реализации. Оба отношения являются предметом верификации и валидации.

Более процедурно цикл моделирования описывается следующей схемой: от определения проблемы к концептуальному моделированию, от концептуальной модели к спецификации (платформонезависимая модель симуляции), далее к реализации (платформозависимая модель) и, наконец, к экспериментированию. Эта последовательность является итеративной: результаты экспериментов влекут уточнение концепции и пересмотр спецификации.

Отдельного внимания заслуживает вопрос верификации и валидации — ключевых процессов обеспечения качества имитационной модели. **Верификация (Verification)** отвечает на вопрос «Правильно ли мы построили модель?» и проверяет, соответствует ли компьютерная реализация задуманной концептуальной модели. **Валидация (Validation)** отвечает на вопрос «Правильную ли модель мы построили?» и устанавливает, насколько адекватно модель воспроизводит целевую реальную систему. В схеме Цайглера различают концептуальную валидацию (соответствие концептуальной модели реальной системе), верификацию компьютерной модели (соответствие реализации концепции) и операциональную валидацию (соответствие результатов симуляции поведению реальной системы).

11.3. Дилемма точности и стоимости: «все модели неверны»

Процесс моделирования неразрывно связан с неустранимым противоречием, которое выразил статистик Джордж Бокс в ставшей афоризмом фразе: *"All models are wrong, but some are useful"* — «Все модели неверны, но некоторые полезны». Это высказывание не

является пессимистическим; напротив, оно задаёт прагматическую рамку для работы с моделями.

С одной стороны, исследователь стремится к повышению точности и детализации модели — добавлению новых механизмов, уточнению параметров, расширению временного горизонта. С другой стороны, каждый шаг в сторону большей реалистичности влечёт рост вычислительной сложности, стоимости разработки и требований к данным. Кривые на диаграмме ниже иллюстрируют этот баланс: ценность модели для пользователя растёт с ростом достоверности (model confidence), но и стоимость её создания растёт нелинейно. Оптимум лежит не на 100% достоверности, а в области разумного баланса.

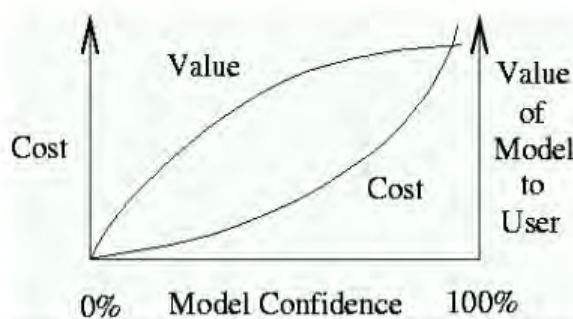


Рисунок 11.1 — Соотношение ценности модели и затрат на её разработку в зависимости от достоверности (Источник: [11])

На практике полная модель «из первых принципов» для реальной системы часто принципиально недостижима из-за ограниченности наблюдений, неточности измерительных приборов, закрытости части информации или принципиальной непредсказуемости отдельных компонент. Поэтому практика имитационного моделирования строится на эвристиках и компромиссах — осознанном упрощении ради достижения конкретной исследовательской цели. Чётко поставленная цель моделирования является главным критерием того, какими аспектами реальности можно пожертвовать.

11.4. Приложения имитационного моделирования

На приведённой на рисунке 11.2 диаграмме показано распределение прикладных задач из различных предметных областей в пространстве трёх парадигм моделирования. Это наглядно демонстрирует, что большинство реальных задач располагаются не строго в одной парадигме, а в промежуточных зонах.

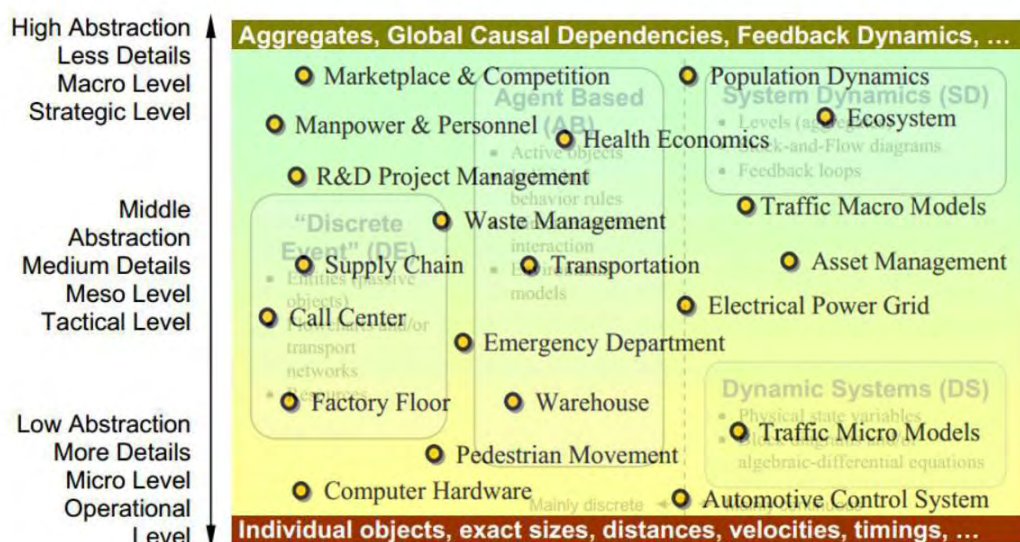


Рисунок 11.2 — Расположение прикладных задач в пространстве парадигм имитационного моделирования (Источник: [12])

Рынки и конкуренция (Marketplace & Competition) и управление персоналом (Manpower & Personnel) тяготеют к макроуровню и применению системной динамики либо агентного моделирования. Задачи управления цепочками поставок (Supply Chain) и логистики находятся на стыке дискретно-событийного и агентного подходов. Моделирование пешеходного движения (Pedestrian Movement) и транспортных потоков располагается на нижнем уровне абстракции и тяготеет к агентным и физическим моделям. Здравоохранение — задачи типа Emergency Department и Health Economics — охватывает широкий диапазон уровней.

Таким образом, одна и та же предметная область может быть смоделирована принципиально разными методами в зависимости от того, какой аспект системы находится в фокусе исследователя.

11.5. Роли имитационного моделирования в аналитическом процессе

Имитационное моделирование может выполнять принципиально разные роли в исследовательском или практическом процессе. Две наиболее распространённые классификации этих ролей описаны ниже.

Аналитический стек. Первая классификация основана на концепции аналитического стека, широко используемой в бизнес-аналитике [2]. Стек включает четыре уровня возрастающей ценности принимаемых решений (см. рис. 11.7):

Наблюдение (Observe) — дескриптивная аналитика (Descriptive Analytics) — описывает, что произошло или происходит.

На этом уровне имитационное моделирование, как правило, не применяется; используются базы данных, запросы и отчётность.

Анализ (Analyze) — диагностическая аналитика (Diagnostic Analytics) — отвечает на вопрос «Почему это произошло?». Симуляция здесь выступает инструментом структурирования информации и восполнения пробелов в данных — совместно с data mining и реальным-временным анализом.

Предсказание (Predict) — предиктивная аналитика (Predictive Analytics) — позволяет получать макрохарактеристики системы в сценариях, которые не наблюдались в реальности. В агентном моделировании типичный пример — предсказание коллективного поведения при изменении внешних условий.

Влияние (Influence) — прескриптивная аналитика (Prescriptive Analytics) — отвечает на вопрос «Как добиться желаемого результата?». Здесь симуляция сочетается с оптимизацией: имитационная модель используется как «чёрный ящик» для поиска оптимальных управляющих воздействий.

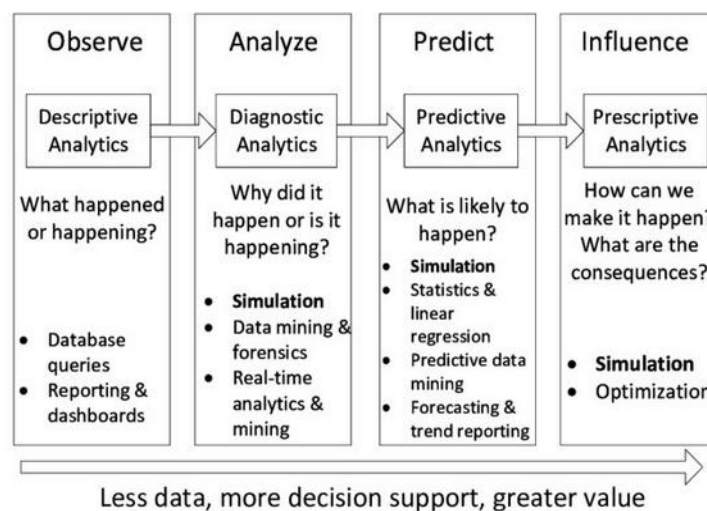


Рисунок 11.3 — Роли симуляции в аналитическом стеке: от наблюдения к влиянию (Источник: [2])

Ось понимания системы. Альтернативный взгляд на роль имитационного моделирования предлагает рассматривать его ценность в зависимости от уровня понимания исследователем целевой системы.

При **низком уровне понимания** системы модель служит **генератором гипотез**: исследователь строит абстрактную модель, которая помогает выявить фундаментальные механизмы и закономерности. Классический пример — модель сегрегации Шеллинга, которая на квадратной решётке демонстрирует, как локальные предпочтения агентов порождают макроскопическую сегрегацию, при

этом не имея прямой привязки к конкретному городу или эпохе. Валидация здесь носит теоретический характер: проверяется логическая состоятельность гипотезы.

При **среднем уровне понимания** модель превращается в «**микроскоп**» — инструмент опосредованного изучения системы. Исследователь уже может отображать отдельные структурные характеристики реальной системы на параметры модели. Та же модель Шеллинга, реализованная на реальной карте города с реальными демографическими данными, перемещается на этот уровень.

При **высоком уровне понимания** модель становится «**калькулятором**»: она способна давать конкретные количественные предсказания поведения реальной системы. На этом уровне верификация и валидация становятся особенно строгими — необходимо не только проверить корректность реализации, но и оценить, насколько реалистичны исходные гипотезы о структуре системы.

11.6. PARTE-фреймворк: архитектура агентной модели

В контексте разработки агентных моделей особую практическую ценность представляет **PARTE-фреймворк** — концептуальная схема, описывающая ключевые компоненты, которые должны быть реализованы в любой агентной имитационной модели. Аббревиатура PARTE расшифровывается следующим образом: **P**roperties (Свойства), **A**ctions (Действия), **R**ules (Правила), **T**ime (Время), **E**nvironment (Среда).

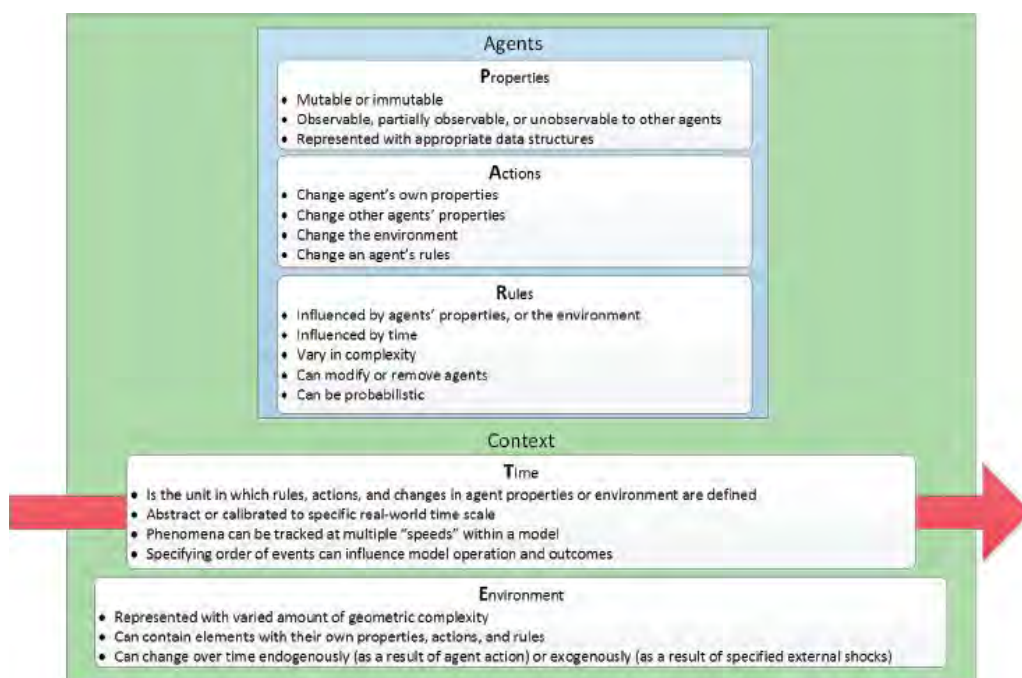


Рисунок 11.4 — PARTE-фреймворк: структура агентной модели
(Источник: [13])

Свойства (Properties) описывают состояние агента. Они могут быть изменяемыми (mutable) или постоянными (immutable), наблюдаемыми другими агентами полностью, частично или вообще ненаблюдаемыми. Для их хранения используются соответствующие структуры данных. В контексте реализации на объектно-ориентированном языке программирования свойства соответствуют полям (атрибутам) класса агента.

Действия (Actions) определяют, что агент может делать. Согласно фреймворку, агент способен: изменять собственные свойства; изменять свойства других агентов (взаимодействие); изменять состояние среды; а также создавать или удалять других агентов (изменять правила их существования). В объектно-ориентированной реализации действия соответствуют методам класса агента.

Правила (Rules) — это логика принятия решений агентом: при каких условиях он совершает то или иное действие. Правила определяются свойствами самого агента и состоянием среды; они могут быть вероятностными, зависеть от времени, варьироваться по сложности. Правила могут также модифицировать или удалять других агентов.

Время (Time) — обязательный компонент любой имитационной модели, поскольку симуляция по определению воспроизводит динамику системы во времени. Время в модели может быть абстрактным (дискретные шаги без привязки к реальному масштабу) или откалиброванным под конкретный реальный масштаб. Принципиально важным является *порядок обработки событий* в рамках одного временного шага, поскольку он может существенно влиять на результаты модели.

Среда (Environment) задаёт пространство, в котором существуют агенты. Она может иметь произвольную геометрическую сложность — от простой решётки до реальной географической карты. Среда может содержать элементы со своими свойствами, действиями и правилами (например, ресурсы, препятствия). Среда может изменяться как вследствие действий агентов (эндогенно), так и под влиянием внешних шоков (экзогенно).

С точки зрения программной реализации, большинство современных фреймворков для агентного моделирования предоставляют базовые абстрактные классы для агента и среды, в которых эти пять компонент заданы в виде абстрактных методов. Исследователь наследует конкретного агента от базового класса и реализует соответствующие методы. Фреймворк берёт на себя

управление временем: на каждом шаге симуляции он вызывает методы агентов, обеспечивает их взаимодействие через среду и продвигает модельное время.

11.7. Лучшие практики разработки имитационных моделей

Существует ряд методических руководств, описывающих процедуру разработки агентных имитационных моделей. Одним из наиболее систематизированных является руководство [3], которое выделяет шесть ключевых шагов (Таблица 11.1). Первый шаг — **определение вопроса или цели (Definition of question or goal)** — является фундаментальным. Нечётко поставленная цель приводит к тому, что модель строится «вообще», а не для решения конкретной задачи. Это, в свою очередь, не позволяет обоснованно принимать решения о допустимых упрощениях. Второй шаг — **определение масштаба и концептуальный дизайн** — требует работы с литературой и экспертным знанием для идентификации ключевых сущностей, взаимосвязей и временного горизонта. Именно здесь задаётся уровень абстракции модели. Третий шаг — **спецификация модели** — переводит концепцию в формализованное описание, пригодное для программной реализации. Здесь принимаются решения о конкретных правилах агентов, параметрах модели и структуре среды. На этом же шаге может быть принято решение о выборе конкретного инструмента или платформы. Четвёртый шаг — **реализация модели** — означает написание программного кода или настройку графического конструктора. Начальные значения параметров должны быть обоснованы на основе реальных данных или экспертных оценок. Пятый шаг (**анализ**) состоит из трёх взаимосвязанных подэтапов. *Тестирование и калибровка* обеспечивают верификацию реализации и соответствие параметров реальным данным. *Дизайн экспериментов и анализ* предполагают систематический перебор условий для проверки гипотез. *Анализ чувствительности* позволяет выявить параметры, наиболее сильно влияющие на результаты, и тем самым оценить устойчивость выводов. Шестой шаг — **синтез и представление результатов** — завершает цикл. Результаты должны быть интерпретированы в терминах исходной исследовательской задачи, представлены наглядно (визуализации, статистические сводки) и задокументированы с достаточной степенью воспроизводимости.

Таблица 11.1 — Ключевые шаги разработки агентной имитационной модели

Ключевой шаг	Описание
1. Определение вопроса или цели (Definition of question or goal)	Тщательно сформулировать цель, которую должна решать модель, или конкретные вопросы, на которые она должна давать ответы.
2. Масштаб модели и концептуальный дизайн (Model scope and conceptual design)	Выявить ключевые концепции, структуры и взаимосвязи из литературы и предварительных исследований. Определить чёткие географические и временные контексты, достаточные для достижения исследовательских целей.
3. Спецификация модели (Model specification)	Спроектировать модель, операционализируя выявленные «ингредиенты» в форме, готовой к реализации.
4. Реализация модели (Model implementation)	Перевести специфицированную модель в вычислительно-исполняемую программу. Определить начальные значения параметров на основе реальных данных и экспертных оценок.
5а. Тестирование и калибровка (Testing and calibration)	Проверить модель на реальных данных и при необходимости итеративно калибровать её дизайн и параметры.
5b. Дизайн экспериментов и анализ (Designing experiments and conducting analysis)	Создать имитационные сценарии для проверки ключевых гипотез исследования и интерпретировать их результаты.
5с. Анализ чувствительности (Sensitivity analysis)	Исследовать пространство параметров для выявления ключевых рычагов воздействия и отобразить, как выбор параметров и допущений влияет на выходные данные модели.
6. Синтез и представление результатов (Synthesis and reporting)	Объединить результаты экспериментов и анализа чувствительности, дать концептуальную интерпретацию. Подготовить статистические анализы и визуализации, чётко документирующие процедуры и выводы.

11.8. Инструменты для агентного моделирования

Ландшафт программных инструментов для агентного и имитационного моделирования чрезвычайно разнообразен. Их можно классифицировать по нескольким критериям: целевое назначение, тип

лицензии, целевая аудитория, язык программирования и масштабируемость.

Инструменты **общего назначения** предназначены для широкого класса задач из любых предметных областей. К ним относятся NetLogo, Mesa, Repast, MASON и AnyLogic. Инструменты же **предметно-ориентированные** разрабатываются с заранее заложенными компонентами конкретной предметной области, что существенно ускоряет построение модели. Так, FlexSim Healthcare содержит готовые модели палат, персонала, потоков пациентов и позволяет построить модель больницы значительно быстрее, чем с нуля на общей платформе. Аналогичные специализированные среды существуют для молекулярной динамики, эпидемиологии и динамики толпы.

По типу лицензии инструменты делятся на **открытые (open-source)**, распространяемые свободно и доступные для модификации, и **проприетарные (proprietary)**, требующие приобретения лицензии. Выбор между ними диктуется не только стоимостью: важно учитывать контекст использования. Так, AnyLogic имеет ограничения на использование в рамках коммерческих решений или грантовых исследований — публикация, поддержанная грантом, может квалифицироваться как коммерческое использование, требующее соответствующей лицензии.

Производительность и масштабируемость. Одним из критических параметров выбора инструмента является производительность при работе с большим числом агентов. Для агентного моделирования, которое по определению обрабатывает каждого агента индивидуально, вычислительная нагрузка растёт, как правило, линейно или хуже с числом агентов и шагов симуляции. Это делает вопрос масштабируемости принципиальным.

Задачи, требующие небольшого числа агентов (десятки, сотни), могут без труда решаться на любом из перечисленных инструментов, включая NetLogo и Mesa на ноутбуке. Задачи с тысячами агентов уже требуют внимания к оптимизации кода. Модели с сотнями тысяч и миллионами агентов — моделирование распространения эпидемий в мегаполисе, молекулярных взаимодействий, динамики больших толп — требуют специализированных высокопроизводительных решений (Distributed MASON, оптимизированные версии Repast) с поддержкой параллельных и распределённых вычислений.

Важно понимать, что переход от прототипа к масштабной модели может потребовать смены инструмента. Поэтому при выборе платформы на этапе постановки задачи следует заранее оценить

планируемый масштаб модели — как по числу агентов, так и по длительности симуляции.

11.9. Заключение

Завершая обзор подходов, инструментов и данных имитационного моделирования, необходимо подчеркнуть, что не существует универсального решения, одинаково хорошо подходящего для всех задач. Выбор инструмента, уровня абстракции и подхода к данным определяется конкретной исследовательской задачей и рядом практических ограничений. При выборе необходимо учитывать следующее. Во-первых, предметную область и специфику задачи: имеются ли предметно-ориентированные инструменты с готовыми компонентами? Каков уровень абстракции, наиболее соответствующий исследовательскому вопросу? Во-вторых, масштаб: сколько агентов и временных шагов потребует модель? Будет ли необходимо масштабирование в будущем? В-третьих, лицензионные условия: для каких целей будет использоваться инструмент — обучение, исследование, коммерческий проект? В-четвёртых, квалификацию команды и доступные вычислительные ресурсы.

Имитационное моделирование, и в особенности агентное моделирование, занимает уникальное место среди методов исследования сложных систем: оно способно соединять микроуровень индивидуального поведения с макроуровнем коллективных паттернов, поддерживать гибридизацию с системной динамикой и дискретно-событийным моделированием, служить одновременно инструментом генерации гипотез и количественного предсказания. Именно это делает его незаменимым компонентом современного исследовательского арсенала в социальных, экономических и биологических науках.

Рекомендуемая литература и источники

1. Zeigler B.P. Theory of Modeling and Simulation / B.P. Zeigler, H. Praehofer, T.G. Kim. — 2nd ed. — San Diego : Academic Press, 2000. — 510 p. — ISBN 978-0-12-778455-7.
2. Holsapple C. A Unified Foundation for Business Analytics / C. Holsapple, A. Lee-Post, R. Pakath // Decision Support Systems. — 2014. — Vol. 64. — P. 130–141. — DOI: 10.1016/j.dss.2014.05.013. - URL: <https://doi.org/10.1016/j.dss.2014.05.013>
3. Mathers A.J. Agent-Based Modeling for Public Health: A Review [Electronic resource] / A.J. Mathers [et al.] // Assessing the Use of Agent-Based Models for Tobacco Regulation / ed. by R. Wallace [et al.]. — Washington : National Academies Press, 2015. — (NBK305917). — URL: <https://www.ncbi.nlm.nih.gov/books/NBK305917/>
4. AnyLogic Company. Simulation Methods Overview [Electronic resource]. — URL: <https://www.anylogic.com/use-of-simulation/>

5. Luke S. MASON: A Multiagent Simulation Environment / S. Luke [et al.] // Simulation. — 2005. — Vol. 81, No. 7. — P. 517–527. — DOI: 10.1177/0037549705058073. - URL: <https://doi.org/10.1177/0037549705058073>
6. North M.J. Complex Adaptive Systems Modeling with Repast Symphony / M.J. North [et al.] // Complex Adaptive Systems Modeling. — 2013. — Vol. 1. — Art. 3. — DOI: 10.1186/2194-3206-1-3. - URL: <https://doi.org/10.1186/2194-3206-1-3>
7. Wilensky U. NetLogo [Electronic resource] / U. Wilensky. — Evanston : Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. — URL: <http://ccl.northwestern.edu/netlogo/>
8. Mesa: Agent-Based Modeling in Python [Electronic resource]. — URL: <https://mesa.readthedocs.io/>
9. Railsback S.F. Agent-Based and Individual-Based Modeling: A Practical Introduction / S.F. Railsback, V. Grimm. — Princeton : Princeton University Press, 2011. — 359 p. — ISBN 978-0-691-13674-5.
10. Law A.M. Simulation Modeling and Analysis / A.M. Law. — 5th ed. — New York : McGraw-Hill, 2015. — ISBN 978-0-07-340132-4.
11. Sargent R.G. Validation and Verification of Simulation Models / R.G. Sargent // Proceedings of the 24th Conference on Winter Simulation (WSC '92). — New York : ACM, 1992. — P. 104–114. — DOI: 10.1145/167293.167311. — URL: <https://doi.org/10.1145/167293.167311>
12. Borshchev A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools / A. Borshchev, A. Filippov // Proceedings of the 22nd International Conference of the System Dynamics Society. — 2004. — URL: <https://www.anylogic.com/resources/articles/from-system-dynamics-and-discrete-event-to-practical-agent-based-modeling-reasons-techniques-tools/>
13. Hammond R.A. Considerations and Best Practices in Agent-Based Modeling to Inform Policy [Electronic resource] / R.A. Hammond // Assessing the Use of Agent-Based Models for Tobacco Regulation / ed. by R. Wallace [et al.]. — Washington : National Academies Press, 2015. — Appendix A. — URL: <https://www.ncbi.nlm.nih.gov/books/NBK305917/>

Лекция 12. Открытые проблемы и перспективы агентного моделирования

Настоящая лекция завершает курс и посвящена систематическому анализу нерешённых задач и перспективных направлений развития агентного моделирования (АМ). Несмотря на впечатляющую историю, охватывающую несколько десятилетий, и широту прикладных областей, агентный подход по-прежнему сталкивается с рядом фундаментальных ограничений как технического, так и концептуального характера. Знание этих ограничений позволяет практикующему исследователю трезво оценивать возможности метода, формулировать реалистичные цели и правильно интерпретировать получаемые результаты.

Ещё в начале курса была представлена шкала применимости агентных моделей: от абстрактных, гипотетических конструкций, нацеленных на выявление общих закономерностей, до конкретных прикладных инструментов, стремящихся воспроизвести реальность с максимальной точностью. На обоих краях этой шкалы подстерегают свои трудности, и к финальной лекции, пройдя весь путь от определений агента до методов имитационного моделирования, имеет смысл ещё раз взглянуть на этот спектр уже с накопленным опытом. Ограничения агентного моделирования принято делить на две большие группы: **технические** и **концептуальные**. Первые связаны с вычислительными и информационными ресурсами, вторые — с самой природой знания, которое мы стремимся получить с помощью модели.

12.1. Технические ограничения

Вычислительная сложность и масштабируемость. Одним из наиболее практически значимых барьеров является **вычислительная сложность** агентных симуляций. Под вычислительной сложностью здесь понимается рост потребности в процессорном времени и оперативной памяти при увеличении числа агентов и усложнении правил их взаимодействия. В идеальном случае для моделирования городской транспортной системы желательно воспроизвести каждого из сотен тысяч водителей и пешеходов в индивидуальных деталях — однако на практике такой подход упирается в технические пределы доступного оборудования. Модель, превосходно работающая при нескольких сотнях агентов, может оказаться нежизнеспособной при масштабировании до миллиона.

Проблема **масштабируемости** (*scalability*) — способности системы сохранять приемлемые характеристики производительности при росте

нагрузки — является отдельным предметом исследования в области высокопроизводительных вычислений. Для АМ это означает необходимость разработки распределённых и параллельных алгоритмов, специализированных архитектур хранения состояния агентов и оптимизированных протоколов межагентного взаимодействия. Такие платформы, как Repast HPC и MASON, частично решают эту задачу, однако не устраняют её принципиально.

Ограниченность и репрезентативность данных. При моделировании реальных социальных, экономических или биологических систем исследователь нередко сталкивается с ситуацией, когда **данные о поведении агентов на микроуровне попросту недоступны**. Если мы моделируем поведение людей — будь то процесс принятия решений или динамика толпы — у нас нет инструмента, который позволил бы наблюдать внутренние когнитивные процессы агента непосредственно. Мы можем спросить человека, почему он поступил так или иначе, но полученный ответ далеко не всегда соответствует реальному механизму принятия решения.

Следствием является то, что многие модели вынуждены опираться на упрощающие допущения и суррогатные данные. Получаемые в итоге параметры модели отражают усреднённое или предполагаемое поведение, а не реальное — что неизбежно сказывается на достоверности выводов. Дополнительным осложнением служит то, что сбор детальных индивидуальных данных (например, о перемещениях конкретных людей или их финансовых транзакциях) нередко вступает в противоречие с требованиями **приватности** и защиты персональных данных.

Идентификация и конструирование модели. Даже при наличии данных процесс **идентификации модели** — подбора её структуры и параметров, наилучшим образом воспроизводящих наблюдаемое поведение системы — является принципиально сложной задачей. Здесь проявляется глубинное противоречие агентного подхода: мы стремимся перейти от гипотетической, абстрактной модели к модели, описывающей конкретную реальность, однако этот переход требует знания, которого у нас часто нет.

Классический пример — модель сегрегации Шеллинга. В её исходном варианте (агенты на квадратной решётке с простым правилом предпочтения соседей) она является абстрактной и демонстрирует общий принцип возникновения сегрегации из микропредпочтений. При попытке адаптировать её к конкретному городу — учесть реальную топологию улиц, демографическую структуру, цены на жильё —

сложность идентификации параметров резко возрастает, и точность воспроизведения реальных данных, как правило, оказывается ограниченной.

Чувствительность к параметрам и неопределённость.

Агентные модели, как правило, демонстрируют высокую **чувствительность к параметрам**: незначительное изменение одного из множества параметров может привести к качественно иному поведению системы. Это порождает риск **ошибок идентификации**, когда модель выдаёт правдоподобные, но ложные результаты, находясь при этом в совершенно другой области параметрического пространства, нежели реальная система.

С этим тесно связана проблема **неопределённости** (*uncertainty*). Неопределённость в АМ имеет двойственный характер: она присутствует как в реальной системе (агенты ведут себя непредсказуемо), так и в самой модели (мы не знаем точных значений параметров). Эти два источника неопределённости накладываются друг на друга, что существенно затрудняет интерпретацию и использование результатов моделирования.

Валидация моделей. Завершающим и, пожалуй, наиболее болезненным техническим вопросом является **валидация** — установление того, в какой мере модель соответствует реальности. Проблема валидации в АМ усугубляется несколькими факторами одновременно: эмерджентным характером результатов (часто трудно заранее предсказать, что именно нужно сравнивать с данными), нехваткой эталонных данных на микроуровне и многопараметричностью моделей. Как обсуждалось в одной из предшествующих лекций, методы валидации включают сравнение с агрегированными статистиками, анализ чувствительности, кросс-валидацию на независимых данных и экспертную оценку — но ни один из них не даёт исчерпывающего ответа на вопрос о достоверности модели.

12.2. Концептуальные ограничения

Принципиальный редукционизм. Агентное моделирование по своей природе является **редукционистским**: оно сводит сложное поведение системы к конечному набору правил, действующих на уровне агентов. Во многих случаях такая редукция допустима и полезна — именно она позволяет получить управляемую и понятную модель. Однако в ситуациях, где поведение агентов обусловлено глубокими когнитивными процессами, неявными социальными нормами,

культурным контекстом или сложной семантикой, редукция к формальным правилам может оказаться принципиально недостаточной.

Здесь уместно вспомнить знаменитое высказывание Джорджа Бокса: «Все модели неверны, но некоторые полезны». Задача исследователя — упростить достаточно, но не более того. Поддержание этого баланса требует не только технической компетентности, но и глубокого понимания предметной области.

Эмерджентное поведение как источник неопределённости.

Способность агентных моделей порождать **эмерджентное поведение** — макропаттерны, не запрограммированные явно, но возникающие из совокупности микровзаимодействий — является одновременно главным достоинством и источником принципиальных затруднений. С одной стороны, именно эта способность позволяет обнаруживать неочевидные системные эффекты. С другой — модель может генерировать артефакты, не имеющие аналогов в реальности, или, напротив, не воспроизводить эффекты, которые мы рассчитываем наблюдать. Отсутствие строгих теоретических предсказаний относительно того, какое именно эмерджентное поведение должна демонстрировать та или иная модель, превращает её анализ в во многом эмпирическое занятие.

Переносимость и применимость моделей. Одним из важных критериев научного знания является его обобщаемость — способность выводить закономерности, применимые не только к конкретному изученному случаю. Агентные модели нередко страдают ограниченной **переносимостью**: модель, точно описывающая процессы в одном конкретном городе или на одном рынке, может оказаться непригодной для другого контекста. Это не означает её бесполезности — даже успешное описание одного сценария представляет практическую ценность — однако ставит под сомнение её научную значимость с точки зрения формирования универсального знания.

12.3. Критичные направления

Помимо перечисленных выше фундаментальных ограничений, в научном сообществе, работающем с агентными моделями, обсуждается ряд дополнительных проблемных областей, каждая из которых заслуживает отдельного рассмотрения.

Гетерогенность агентов и индивидуальные различия.

Большинство агентных моделей, особенно учебного и демонстрационного характера, оперируют **когортными**, или однородными агентами — то есть агентами, объединёнными в группы с

идентичными параметрами поведения. Однако реальные системы — прежде всего социальные — по природе своей **гетерогенны**: каждый человек уникален, и на его поведение одновременно влияют десятки факторов — от физиологических и психологических особенностей до социального положения и текущего эмоционального состояния.

Моделирование гетерогенных агентов требует значительно большего числа параметров, порождает острые проблемы идентификации и резко увеличивает вычислительную нагрузку. Поиск баланса между реалистичностью (учёт индивидуальных характеристик) и управляемой сложностью модели является нетривиальной задачей. Одним из перспективных подходов является использование методов машинного обучения для автоматического извлечения типичных поведенческих профилей агентов из эмпирических данных — что само по себе порождает новые исследовательские вопросы.

Неопределённость и анализ чувствительности. Работа с неопределённостью в АМ включает несколько взаимосвязанных задач. **Анализ чувствительности** (*sensitivity analysis*) позволяет установить, какие параметры модели оказывают наибольшее влияние на результаты, и тем самым определить, где сосредоточить усилия по калибровке. Стандартные методы — однофакторный анализ (ОАТ), метод Морриса, индексы Соболя — позволяют систематически исследовать пространство параметров, однако при большой их размерности становятся вычислительно затратными.

Особую сложность представляет учёт **непредвиденных событий** (*black swans*), кардинально меняющих динамику системы. Агентные модели, откалиброванные на «нормальных» данных, как правило, плохо справляются с воспроизведением редких, но критически важных событий — финансовых кризисов, пандемий, природных катастроф. Это ограничение имеет принципиальный характер, поскольку именно для прогнозирования и управления подобными событиями инструменты АМ были бы наиболее востребованы.

Междисциплинарность: возможность и вызов. Одна из отличительных черт агентного моделирования, одновременно являющейся его сильной стороной и источником трудностей, — **врождённая междисциплинарность**. АМ сложилось на пересечении теории сложных систем, информатики, социальных наук, математики и биологии — и именно поэтому оно с лёгкостью переносится на широкий круг предметных областей: от молекулярной динамики и эпидемиологии до транспортного планирования и финансов.

Ключевым механизмом этой переносимости является возможность заимствовать **эвристики поведения** агентов непосредственно из изучаемой предметной области. Правила поведения клеток в биологическом эксперименте, тактические решения пешеходов в здании, стратегии трейдеров на биржевом рынке — всё это может быть относительно естественным образом транслировано в правила агентов модели. Тем самым агентное моделирование создаёт общий язык между специалистами разных областей.

Вместе с тем эффективное **сотрудничество специалистов разных дисциплин** требует преодоления серьёзных барьеров — терминологических, методологических и организационных. Математик и социолог по-разному понимают такие понятия, как «модель», «валидация» или «рациональность»; экономист и биолог имеют разные традиции представления данных и оценки качества модели. Создание продуктивных мультидисциплинарных коллективов — само по себе исследовательская задача, которой уделяется всё больше внимания в организации науки.

Этические соображения. По мере того как агентные модели всё активнее используются для поддержки принятия ответственных решений — в здравоохранении, финансовом регулировании, городском планировании, управлении критической инфраструктурой — на первый план выходят вопросы этики. Здесь можно выделить несколько взаимосвязанных аспектов.

Приватность данных. Построение реалистичных агентных моделей поведения людей требует детальных индивидуальных данных, сбор и использование которых неизбежно затрагивает права граждан на конфиденциальность. Модели, откалиброванные на данных геолокации, финансовых транзакций или медицинских записей, потенциально несут риски утечки или злоупотребления чувствительной информацией.

Смещение и справедливость (*bias and fairness*). Если данные, на основе которых строится и калибруется модель, отражают существующие социальные неравенства или системные предубеждения, модель может воспроизводить и усиливать эти несправедливости. Это особенно критично в случаях, когда результаты моделирования влияют на распределение ресурсов или возможностей между различными группами населения.

Прозрачность и подотчётность (*transparency and accountability*). Для тех, кто принимает решения на основе результатов модели, важно понимать, как именно эти результаты получены. «Чёрные ящики», выдающие рекомендации без объяснения механизма, — неприемлемый

инструмент для критически важных приложений. Принципы объяснимого искусственного интеллекта (*Explainable AI, XAI*) становятся всё более актуальными и для агентного моделирования.

Отдельной и особенно острой темой является этика автономных агентов. В системах, где агенты принимают решения самостоятельно — будь то беспилотные транспортные средства, торговые роботы или системы поддержки медицинских решений — возникает вопрос об ответственности за последствия этих решений. Кто несёт ответственность за результат: агент, его разработчик, оператор или пользователь? Этот вопрос уже перестал быть сугубо теоретическим: реальные прецеденты, связанные с действиями автономных систем, повлёкшими человеческие жертвы или значительный материальный ущерб, поставили его на повестку дня международного права и этики. Активно развивающаяся область **этики искусственного интеллекта** — с её манифестами, комитетами и регуляторными инициативами — напрямую касается и агентного моделирования в той мере, в какой оно создаёт основу для автономных систем принятия решений.

12.4. Перспективы развития

Интеграция с машинным обучением. Наиболее активно обсуждаемым и, по всей видимости, наиболее перспективным направлением является глубокая интеграция агентного моделирования с методами **машинного обучения** (*machine learning, ML*). На протяжении курса неоднократно рассматривалось **обучение с подкреплением** (*Reinforcement Learning, RL*) — подход, в котором агент обучается оптимальной стратегии поведения через взаимодействие со средой, получая награду или штраф за свои действия. Симбиоз АМ и RL позволяет создавать агентов, которые не просто следуют фиксированным правилам, но адаптируются, обучаются и совершенствуют своё поведение в процессе симуляции.

Это открывает принципиально новые возможности. Во-первых, агенты получают способность к адаптации, сближая модель с реальностью, где агенты тоже обучаются. Во-вторых, методы RL позволяют автоматизировать поиск оптимальных стратегий поведения агентов — то, что ранее требовало явной экспертной спецификации правил. В-третьих, среда многоагентной симуляции становится полигоном для обучения RL-агентов, которые затем могут применяться в реальных задачах управления, планирования и прогнозирования.

Помимо RL, перспективным является использование методов **генеративных моделей** для синтеза реалистичных поведенческих данных агентов, а также применение **нейросетевых аппроксиматоров** для моделирования сложных когнитивных функций агентов — восприятия, памяти, планирования. Последнее направление смыкается с изучавшимися в главе 10 когнитивными архитектурами, такими как ACT-R и SOAR, и открывает путь к принципиально более реалистичным моделям человеческого поведения.

Расширение прикладных областей. История агентного моделирования, как было показано в главе 2, демонстрирует неуклонное расширение прикладных областей. Сегодня АМ активно применяется в эпидемиологии (моделирование распространения инфекционных заболеваний с учётом индивидуального поведения), в экологии (динамика популяций и экосистем), в экономике и финансах (моделирование рыночных механизмов, финансовой нестабильности), в транспортном планировании, урбанистике, социологии, военных приложениях и других областях. Каждое из этих приложений порождает специфические исследовательские вопросы и требует дальнейшего развития методологии.

Особого внимания заслуживает применение АМ для изучения и управления **сложными социотехническими системами** — такими как цифровые платформы, глобальные цепочки поставок, энергетические сети или системы здравоохранения. Эти системы характеризуются тесным переплетением технических и социальных компонентов, нелинейной динамикой и высокой чувствительностью к малым воздействиям — именно теми чертами, для анализа которых агентный подход предоставляет уникальные инструменты.

Методологическое развитие. Помимо прикладных, существуют важные направления методологического развития АМ. Среди них: разработка более строгих и стандартизированных **протоколов валидации** и верификации моделей; создание открытых репозиториев моделей и данных для воспроизведения результатов; развитие **формальных языков описания моделей** (таких как ODD — Overview, Design Concepts, Details), позволяющих однозначно и полно документировать агентные модели; совершенствование методов **анализа чувствительности** применительно к высокоразмерным пространствам параметров.

Активно развивается и теоретическая база: в частности, исследуются математические условия возникновения различных

режимов поведения в агентных системах, связь между микроправилами и макростатистиками, а также связь агентного моделирования с теорией сложных систем, нелинейной динамикой и теорией игр.

12.5. Заключение

Завершая курс, важно ещё раз обозначить то место, которое занимает агентное моделирование в современном научном ландшафте. Несмотря на значительную историю и зрелость как дисциплины, АМ остаётся молодой и динамично развивающейся областью с открытыми фундаментальными вопросами. Это делает её особенно привлекательной для исследователей, готовых работать на переднем крае науки.

Технические ограничения — вычислительная сложность, недостаток данных, трудности идентификации и валидации — реальны, но не фатальны. Они задают направления для технологических инноваций: в области высокопроизводительных вычислений, методов машинного обучения, сенсорики и сбора данных. Концептуальные ограничения — редукционизм, непредсказуемость эмерджентного поведения, ограниченная переносимость — ставят глубокие философские вопросы о природе моделирования и научного знания, не имеющие простых ответов.

Агентное моделирование уникально своей **методологической гибкостью**: через эвристики поведения агентов оно создаёт мост между теоретическими принципами различных дисциплин и конкретными вычислительными экспериментами. Именно эта гибкость обеспечивает ему устойчивое место среди ключевых инструментов анализа сложных систем — наряду с дифференциальными уравнениями, теорией игр и методами машинного обучения.

Интеграция агентного моделирования с современными подходами искусственного интеллекта, в первую очередь с обучением с подкреплением, открывает принципиально новый класс возможностей: создание агентов, способных адаптироваться, обучаться и принимать сложные решения в динамичной среде. Это означает, что будущее АМ — не в постепенном угасании перед лицом новых методов, а в органичном синтезе с ними, сохраняющем главные достоинства агентного подхода: интуитивность, интерпретируемость и способность работать со сложными гетерогенными системами.

Вопросы этики, прозрачности и подотчётности агентных систем будут приобретать всё большее значение по мере их внедрения в критически важные приложения. Ответственный исследователь в

области АМ должен не только владеть техническим инструментарием, но и понимать социальные последствия своей работы — именно к этому пониманию стремился подвести настоящий курс лекций.

Рекомендуемая литература и источники

1. Axelrod R. The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. — Princeton : Princeton University Press, 1997. — 232 p.
2. Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems // Proceedings of the National Academy of Sciences. — 2002. — Vol. 99, suppl. 3. — P. 7280–7287. — DOI: 10.1073/pnas.082080899. - URL: <https://doi.org/10.1073/pnas.082080899>
3. Epstein J.M. Generative Social Science: Studies in Agent-Based Computational Modeling. — Princeton : Princeton University Press, 2006. — 352 p.
4. Macal C.M., North M.J. Tutorial on agent-based modelling and simulation // Journal of Simulation. — 2010. — Vol. 4, no. 3. — P. 151–162. — DOI: 10.1057/jos.2010.3. - URL: <https://doi.org/10.1057/jos.2010.3>
5. Railsback S.F., Grimm V. Agent-Based and Individual-Based Modeling: A Practical Introduction. — 2nd ed. — Princeton : Princeton University Press, 2019. — 376 p.
6. Handbook of Computational Economics. Vol. 2: Agent-Based Computational Economics / ed. by L. Tesfatsion, K.L. Judd. — Amsterdam : North-Holland, 2006. — 904 p.
7. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. — 2nd ed. — Cambridge : MIT Press, 2018. — 548 p. — URL: <http://incompleteideas.net/book/the-book-2nd.html>
8. Floridi L. et al. An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations // Minds and Machines. — 2018. — Vol. 28, no. 4. — P. 689–707. — DOI: 10.1007/s11023-018-9482-5. - URL: <https://doi.org/10.1007/s11023-018-9482-5>

Заключение

Настоящий курс был построен как обзор единого понятия в нескольких принципиально разных контекстах. В зависимости от угла зрения агент предстаёт то как автономная сущность в клеточном автомате, то как рациональный игрок в теоретико-игровой постановке, то как обучающаяся система в среде с вознаграждением, то как когнитивный субъект с убеждениями, желаниями и намерениями. За этим разнообразием стоит единый каркас: агент воспринимает среду, действует в ней, управляет доступными ресурсами и взаимодействует с другими агентами — будь то сородичи, соперники или люди. Именно эти пять понятий — агент, среда, ресурсы, коммуникация, цель — связывают все двенадцать лекций в единое целое и служат навигатором при выборе методов и инструментов для конкретной прикладной задачи.

Методологическим ядром курса остаётся подход «снизу вверх»: сложные глобальные паттерны — пространственная сегрегация, стайное поведение, финансовые пузыри, поляризация мнений — возникают из простых локальных правил взаимодействия, не будучи заложены в них явно. Это делает вычислительный эксперимент незаменимым инструментом там, где аналитическое решение принципиально недостижимо. Сквозной темой курса стала роль человека как специфического агента — одновременно объекта моделирования, источника функции вознаграждения и партнёра в системах поддержки принятия решений. Понимание того, как человек воспринимает информацию, обучается и ошибается, не менее важно для специалиста, чем владение формальным аппаратом обучения с подкреплением или теорией игр.

Одной из ключевых целей курса было выстроить мост между фундаментальными концепциями и конкретными технологиями: за каждым понятием — уравнением Беллмана, равновесием Нэша, фронтом Парето, BDI-моделью, усвоением данных — стоит класс задач, для которых соответствующий инструмент наиболее естественен. Вместе с тем курс честно обозначил пределы подхода. Вычислительная сложность при масштабировании, трудности идентификации и валидации моделей, принципиальная непредсказуемость эмерджентного поведения, открытые вопросы этики автономных систем — всё это реальные ограничения, а не технические детали. Исследователь, понимающий, где модель надёжна, а где она лишь «полезна, хотя и неверна», способен извлекать из неё реальную

ценность, не впадая ни в иллюзию полноты, ни в паралич неопределённости.

Агентное моделирование переживает один из наиболее динамичных периодов своей истории: интеграция с глубоким обучением, появление LLM-агентов, развитие концепции цифровых двойников — всё это не отменяет фундаментальных принципов курса, а наполняет их новым содержанием. Авторы надеются, что освоенные материалы послужат не только введением в конкретные методы и инструменты, но и приглашением к особому стилю мышления: видеть в сложном — взаимодействие простых элементов, в непредсказуемом — закономерности, порождённые правилами, в разнообразии технологий — единую понятийную структуру. Агентный взгляд на мир является, пожалуй, наиболее ценным результатом освоения курса, выходящим за рамки любого конкретного алгоритма или платформы.

Приложение 1. Вопросы для самопроверки и тестирования

1) Что не является обязательным свойством агентов?

- А) Автономность
- Б) Эмерджентность поведения
- В) Перцепция
- Г) Интерактивность

2) Какие элементы обязательно должны присутствовать в рамках агентной модели?

- А) Агенты и процедура их обучения
- Б) Агенты и среда
- В) Статические и динамические характеристики среды
- Г) Большое число агентов

3) Ключевые процессы взаимодействия агента со средой — это...

- А) Рассуждение и прогнозирование
- Б) Перцепция и внутренняя логика агента
- В) Действия и перцепция
- Г) Действия и изменчивость среды

4) Какие основные три типа моделирования выделялись в рамках курса?

- А) Описательное, предсказательное, рекомендательное
- Б) Описательное, предсказательное, прогнозирующее
- В) Индуктивное, дедуктивное, абдуктивное
- Г) Описательное, формально-логическое, концептуальное

5) Какие из перечисленных моделей не являются агентами по определению?

- А) Шеллинга
- Б) Лотки-Вольтерра
- В) Boids
- Г) Брайтенберга

6) Кого можно считать одним из пионеров агентного моделирования?

- А) Томаса Шеллинга
- Б) Иммануила Канта
- В) Джошуа Эпштейна
- Г) Дирка Хелбинга

7) Какие агентные модели применяются для описания динамики толпы?

- А) Модели гидродинамики
- Б) SIR-модели
- В) Модели социальной динамики
- Г) Балансовые модели

8) Применяются ли агентные модели в биомедицине, для чего?

- А) Да, для описания динамики жизненных показателей
- Б) Да, для описания клеточной динамики
- В) Применяются в здравоохранении, но не в биомедицине
- Г) Нет

9) Чем характеризуется текущий этап развития агентного моделирования?

- А) Формированием фундаментальных принципов и подходов
- Б) Формированием области computational social science
- В) Зарождением принципов эмерджентного поведения
- Г) Гибридизацией и интеграцией методов

10) Задача агента по определению соотносится с ...

- А) Перцепцией и действиями
- Б) Результатами и целями
- В) Средой и ее динамикой
- Г) Когнитивными свойствами агента

11) Когнитивные архитектуры делятся на ...

- А) Символьные и эмерджентные
- Б) Прямые и обратные
- В) Гибридные и монолитные
- Г) Гибридные и символьные

12) Агент, для которого действия напрямую определяются перцепцией -

- А) Рассуждающий
- Б) Простой реактивный агент
- В) Агент с памятью
- Г) Интеллектуальный агент

13) Типовые способы обработки нескольких реакций в простом реактивном агенте:

- А) Отмена, параллельное выполнение
- Б) Приоритезация, комбинация, формирование очереди
- В) Формирование очереди, планирование, параллельное выполнение
- Г) Приоритезация, комбинация, параллельное выполнение

14) Тележка Брайтенберга — это пример агента какого типа?

- А) Рассуждающего
- Б) Реактивного с памятью
- В) Простого реактивного
- Г) Интеллектуального

15) Каким образом обрабатываются несколько действий в модели Boids?

- А) Приоретизация
- Б) Агрегация итоговых координат
- В) Агрегация ускорения

Г) Случайный выбор

16) Типовые пути решения проблем “застревания” и циклического поведения в реактивных агентах

- А) Рандомизация и планирование
- Б) Рандомизация и избегание нежелательного поведения
- В) Планирование и избегание нежелательного поведения
- Г) Глубокое обучение

17) Способ избегания нежелательного поведения в реактивных агентах предполагает использование

- А) Когнитивной архитектуры
- Б) Памяти
- В) Комбинации действий
- Г) Перцепции

18) Subsumption architecture основана на ...

- А) Иерархии моделей поведения
- Б) Глубоком обучении с подкреплением
- В) Социальном поведении роботов
- Г) Эмерджентных когнитивных архитектурах

19) Ключевая особенность рассуждающего (deliberative) агента это

- А) Использование памяти
- Б) Сравнение альтернативных решений
- В) Использованием глубокого обучения
- Г) Строгая ориентация на задачи робототехники

20) Архитектура IRMA основана на ...

- А) Модели BDI
- Б) Модели сегрегации
- В) Когнитивной архитектуре
- Г) Усвоении данных

21) Масштабы оценки характеристик многоагентных систем:

- А) Внутренний и системный
- Б) Масштаб агента и масштаб среды
- В) Интер-агентный и интра-агентный
- Г) Микро и макро

22) Типы коммуникации агентов в многоагентных системах:

- А) Массовая, индивидуальная и система с общей памятью
- Б) Вербальная и невербальная
- В) Массовая и маршрутизируемая
- Г) Синтаксическая и семантическая

23) Игра “Жизнь” — это пример

- А) Модели системной динамики
- Б) Модели обучения с подкреплением
- В) Модели клеточного автомата
- Г) Многоагентной модели с непрерывной средой

24) Модель Изинга описывает ...

- А) Распространение инфекции
- Б) Поляризацию мнений
- В) Сегрегацию
- Г) Динамику толпы

25) К свойствам сложных систем не относится:

- А) Большое число элементов
- Б) Многомасштабная изменчивость
- В) Хаотическая природа процессов
- Г) Дальние связи между элементами

26) О том, что модель *boids* является сложной системой, может свидетельствовать:

- А) Реактивный характер агентов
- Б) Возможность взаимодействия с соседними агентами
- В) Три типа взаимодействия агентов
- Г) Эмерджентное поведение

27) К основным классам сложности не относится:

- А) Организованная простота
- Б) Неорганизованная сложность
- В) Организованная сложность
- Г) Неорганизованная простота

28) В модели Шеллинга сегрегация достигает максимальных значений при ... значениях гомофилии.

- А) Минимальных
- Б) Средних
- В) Максимальных
- Г) Нулевых

29) В дополнение к классической постановке агентного моделирования в обучение с подкреплением вводится ...

- А) Вознаграждение
- Б) Среда
- В) Действия
- Г) Наблюдение среды

30) В обучении с подкреплением стратегия определяет отображение состояния в ...

- А) Ценность
- Б) Вознаграждение
- В) Действие
- Г) Наблюдение

31) Поиск стратегии методом динамического программирования сходится при коэффициенте дисконтирования γ

- А) > 1
- Б) $= 1$
- В) < 1
- Г) > 0

32) В отличие от планирования, обучение с подкреплением работает

- А) Только с известной моделью среды
- Б) С изначально неизвестной средой
- В) С дискретным пространством действий
- Г) С улучшением стратегии

33) Оценка “сожаления” в обучении с подкреплением применяется

- А) Для ограничения пространства поиска
- Б) Для дискретизации пространства действий
- В) Для реализации “метода проб и ошибок”
- Г) Для описания когнитивных состояний агента

34) Многоагентное обучение с подкреплением может работать

- А) В режиме кооперации или конкуренции агентов
- Б) Только в режиме кооперации агентов
- В) Только в режиме конкуренции
- Г) Только в рамках заранее определенных команд

35) RLHF это – обучение с подкреплением ...

- А) С гибридными функциями (hybrid function) вознаграждения
- Б) Для человеческого фреймворка (human framework)
- В) С человеческой обратной связью (human feedback)
- Г) С высокой частотой (high-frequency)

36) Парадокс Кондорсе определяет

- А) Противоречивость ранжирования элементов
- Б) Невозможность построения многокритериальных оценок
- В) Ограничение некооперативных игр
- Г) Нарушение сходимости некоторых метода оптимизации

37) Игра с нулевой суммой — синоним

- А) Кооперативной игры

- Б) Некооперативной игры
- В) Антагонистической игры
- Г) Азартной игры

38) Цель дизайна механизмов состоит в оптимизации

- А) Суммарного вознаграждения агентов
- Б) Ожидаемой полезности
- В) Функции социального выбора
- Г) Числа шагов аукциона

39) Какой из механизмов не ориентирован на работу с многокритериальными оценками?

- А) Скаляризация
- Б) Метод анализа иерархий
- В) Анализ Парето-фронта
- Г) Градиентный спуск

40) Эхо-камеры приводят к ...

- А) Иллюзии большинства
- Б) Сегрегации
- В) Повышению эффективности система голосований
- Г) Снижению размерности пространства критериев оценки

41) Верификация проверяет соответствие (“чего” - “чему”)

- А) Концептуальной модели – наблюдаемым данным
- Б) Компьютерной модели – рассматриваемой системе
- В) Компьютерной модели – концептуальной модели
- Г) Концептуальной модели – компьютерной модели

42) Абстрактные агентные модели в основном основаны на ...

- А) Машинном обучении
- Б) Теоретических предположениях
- В) Доступных данных наблюдений
- Г) Предсказании характеристик наблюдаемых объектов реального мира

43) В основные три вида имитационного моделирования не входит ...

- А) агентное моделирование
- Б) дискретно-событийное моделирование
- В) модели системной динамики
- Г) графовые модели

44) Усвоение данных корректирует модель ...

- А) По мере поступления данных
- Б) В рамках анализа истории наблюдений
- В) При начальной настройке модели
- Г) Ни одно из вышеперечисленного

45) Сколько циклов обратной связи рассматривает подход DDDAS?

- А) 1
- Б) 2
- В) Неограниченное множество
- Г) Ни одного

46) Criterion validity оценивает

- А) Качество системы критериев оценки
- Б) Качество моделирования по выбранным критериям оценки
- В) Непротиворечивость критериев оценки
- Г) Соответствие целевому сценарию моделирования

47) Обратное обучение с подкреплением направлено на определение

- А) Поведения агента
- Б) Стратегии
- В) Качества модели среды
- Г) Вознаграждения

48) Модель психики человека (Theory of Mind) описывает

- А) Класс когнитивных архитектур
- Б) Модель неявного вознаграждения
- В) Описание рассуждений другого агента
- Г) Социальное поведение агентов

49) К ролям имитационных моделей не относятся

- А) Генератор
- Б) Предиктор
- В) Медиатор
- Г) Наблюдатель

50) В структуре PARTE framework объединяющую роль играет

- А) Environment
- Б) Rules
- В) Time
- Г) Action

Ключ :

1: Б Б В А Б А В Б Г Б
11: А Б Г В В Б Б А Б А
21: Г А В Б В Г Г Б А В
31: В Б В А В А В В Г А
41: В Б Г А Б Б Г В Г В

Приложение 2. Вопросы для проведения экзамена

1. Понятие агента и среды: общие понятия, определения, характеристики
2. Архитектура и характеристики агентов: реактивные агенты
3. Архитектура и характеристики агентов: рассуждающие агенты
4. Архитектура и характеристики агентов: гибридные архитектуры
5. Перцепция агентов
6. Действия и решения агентов, рациональность
7. Многоагентные системы: свойства среды и агентов
8. Многоагентные системы: особенности реализации
9. Сложные системы в многоагентном моделировании: принципы построения
10. Эмерджентность и коллективное поведение агентов
11. Обучение с подкреплением: базовые определения, модели и алгоритмы
12. Модели принятия решений
13. Модели голосования и коллективного принятия решений
14. Работа с данным в многоагентных системах: усвоение, DDDAS
15. Имитационное моделирование агентов
16. Программные инструменты агентного моделирования: классы, требования, свойства
17. Верификация и валидация агентных моделей
18. Когнитивные архитектуры: базовые модели, виды
19. Когнитивные шкалы, искусственный интеллект общего назначения
20. Обратное обучение с подкреплением и Theory of Mind
21. Иррациональное поведение и когнитивные искажения

Ковальчук Сергей Валерьевич
Береговенко Илья Игоревич

**Моделирование и предсказание поведения агентов:
конспект лекций**
учебное пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе