

Многоагентные системы

Основные концепции, архитектуры и применение

31 октября 2025 г.

1 Введение в многоагентные системы

1.1 Концепция агента

Многоагентная система (МАС) — это система из нескольких взаимодействующих интеллектуальных агентов, распределенных по сети, способных мигрировать по ней в поисках релевантных данных, знаний и процедур и кооперирующихся в процессе выработки решений.

Существует несколько определений агента, предложенных различными исследователями:

Wooldridge and Jennings (1995): “Агент — это инкапсулированная вычислительная система, помещенная в некоторую среду и способная автономно выполнять действия в этой среде для достижения поставленных целей.”

Shaw Green et al. (1997): “Агент — это вычислительная система, автономно действующая от лица других сущностей, выполняющая действия реактивно и/или с определенной целью и в некоторой степени использующая свойства обучаемости, кооперативности и мобильности.”

1.2 Основные свойства агента

В самом общем случае агент обладает следующими свойствами:

1. **Реактивность** — способность агента воспринимать окружающую среду и влиять на нее. Агенты способны воспринимать состояние среды и своевременно реагировать на те изменения, которые в ней происходят.
2. **Целеустремленность (про-активность)** — агент должен действовать в соответствии с заложенными в него целями. Это способность агента брать на себя инициативу, генерировать цели и действовать рационально для их достижения, а не только реагировать на внешние события.
3. **Социальная активность** — агент должен взаимодействовать с другими агентами и/или людьми. Способность функционировать в сообществе с другими агентами, обмениваясь с ними сообщениями с помощью некоторого общепонятного языка коммуникаций.

4. **Автономность** — агент действует без непосредственного вмешательства человека и обладает определенным контролем над своими действиями и внутренним состоянием. Способность ИА функционировать без вмешательства человека и при этом осуществлять самоконтроль над своими действиями и внутренним состоянием.

1.3 Определения интеллектуального агента

Принято различать два определения интеллектуального агента — “слабое” и “сильное”:

1.3.1 Слабое определение

Под интеллектуальным агентом в слабом смысле понимается программно или аппаратно-реализованная система, которая обладает такими свойствами, как:

- автономность
- общественное поведение (sociability)
- реактивность (reactivity)
- про-активность (pro-activity)

1.3.2 Сильное определение

Сильное определение агента подразумевает дополнительно к перечисленным выше свойствам наличие у агента хотя бы некоторого подмножества так называемых “ментальных свойств” (интенциональных понятий):

Знания (knowledge) — постоянная часть знаний агента о себе, среде и других агентах, которая не изменяется в процессе его функционирования.

Убеждения (beliefs) — знания агента о среде, в частности, о других агентах; это те знания, которые могут изменяться во времени и становиться неверными, однако агент может не иметь об этом информации и продолжать оставаться в убеждении, что на них можно основывать свои выводы.

Желания (desires) — состояния и ситуации, достижение которых по разным причинам является для агента желательным, однако они могут быть противоречивыми, и потому агент не ожидает, что все они будут достигнуты.

Намерения (intentions) — то, что агент или обязан сделать в силу своих обязательств по отношению к другим агентам, или то, что вытекает из его желаний (непротиворечивое подмножество желаний, выбранное по тем или иным причинам и совместимое с принятыми на себя обязательствами).

Цели (goals) — конкретное множество конечных и промежуточных состояний, достижение которых агент принял в качестве текущей стратегии поведения.

Обязательства (commitments) — задачи, которые агент берет на себя по просьбе (поручению) других агентов в рамках кооперативных целей или целей отдельных агентов в рамках сотрудничества.

1.4 Дополнительные свойства агентов

Некоторые исследователи считают, что агент должен обладать также рядом других свойств:

Мобильность (mobility) — способность агента мигрировать по сети в поисках необходимой информации для решения своих задач, при кооперативном решении задач совместно или с помощью других агентов.

Благожелательность (benevolence) — готовность агентов помочь друг другу и готовность агента решать именно те задачи, которые ему поручает пользователь, что предполагает отсутствие у агента конфликтующих целей.

Правдивость (veracity) — свойство агента не манипулировать информацией, про которую ему заведомо известно, что она ложна. Предположение, что агент не может намеренно фальсифицировать передаваемую (другим агентам, человеку) информацию.

Рациональность (rationality) — свойство агента действовать так, чтобы достигнуть своих целей, а не избегать их достижения, по крайней мере, в рамках своих знаний и убеждений. Предположение, что агент действует в соответствии со своими целями и не пытается противостоять себе (по крайней мере, насколько это позволяют его убеждения).

2 Теоретические основы многоагентных систем

2.1 Теоретические основы агентов

Существует ряд теорий и концепций, использующихся для описания агентов:

- **Логические системы** — цели и свойства агента описываются при помощи высказываний в различных логических системах.
- **Системы намерений** — внутреннее состояние агента представляется системой мировоззрений (знания, убеждения, желания, намерения и т.п.).
- **Модели коммуникаций** — взаимодействие между агентами происходит посредством специальных действий.

2.2 Формализация ментальных понятий

Первые попытки построить модели ментальных понятий базировались на языках исчисления предикатов первого порядка, однако они не оказались удачными. Например, формула:

$\text{Bel}(\text{Агент_А}, \text{Имеет}(\text{Агент_Б}, X))$

Не является правильно построенной формулой исчисления предикатов первого порядка, поскольку вторым аргументом предиката $\text{Bel}(,)$ является, в свою очередь, предикат.

При выборе формализмов для описания ментальных понятий нужно решать два класса проблем:

- синтаксическую проблему
- семантическую проблему

Для описания синтаксиса существует два варианта:

1. Использование мета-языков (многосортовая логика первого порядка)
2. Использование расширений модальных логик с специальными модальными операторами

С точки зрения семантики языка формализации ментальных понятий применяются:

1. Семантика, представляемая множеством возможных миров
2. Интерпретация символических структур с помощью поставленных им в соответствие функций и структур данных

2.3 Семантика возможных миров

В семантике множества возможных миров ментальные понятия интерпретируются множеством возможных миров и отношением достижимости (доступности) между ними. С каждым возможным миром ассоциируется некоторая теория (множество формул и атомарных предикатов – истинных фактов).

Пример: если игрок в покер получил пиковую даму, то он может сделать выводы о допустимых и недопустимых раскладах карт у других игроков, и множество всех допустимых раскладов карт образует возможный мир игрока А для теории “Игрок А получил пиковую даму”.

3 Приложения многоагентных систем

МАС используются для решения широкого спектра задач:

1. Регулирование трафика — управление транспортными потоками в городах
2. Онлайн-торговля — электронная коммерция и автоматизация торговых операций
3. Моделирование социальных структур — исследование социальных процессов и взаимодействий
4. Групповой ИИ в играх и фильмах — создание реалистичного поведения NPC
5. Устранение чрезвычайных ситуаций (ЧС) — координация служб при авариях и катастрофах
6. Сенсорные сети — распределенный сбор и обработка данных

3.1 Пример: Сенсорная сеть

Сенсорная сеть состоит из автономных сенсоров, постоянно собирающих информацию о температуре/влажности/давлении и пр. Для передачи информации соседние сенсоры должны иметь разные частоты.

Условия задачи:

- Каждый сенсор может выбрать 1 из 3 частот
- Соседние сенсоры не должны иметь одну частоту

Эта задача эквивалентна задаче о раскраске графа в 3 цвета и является примером задачи с ограничениями.

3.2 Задачи с ограничениями

Задача с ограничениями включает:

- Множество переменных (сенсор, вершина)
- Множества возможных значений для каждой переменной (частота, цвет)
- Множество ограничений (пересекающиеся сенсоры, смежные вершины)
- Необходимо назначить каждой переменной значение так, чтобы ограничения были выполнены

Задача оптимизации дополнительно включает:

- Функцию веса для каждого нарушенного ограничения
- Необходимо минимизировать сумму весов нарушенных ограничений

Важно: Задачи с ограничениями в общем случае NP-полные!

3.3 Распределенные задачи с ограничениями

В распределенной задаче оптимизации/с ограничениями переменные распределены между агентами. Обычно каждый агент отвечает за одну переменную.

Подходы к решению:

1. Адаптация классических решений — перебор, нахождение решения во всём пространстве поиска
2. Адаптация алгоритмов локального поиска — нахождение оптимального решения в локальной области пространства поиска
3. Кооперативные подходы — в основном заимствованные из природы/социологии

3.4 Асинхронный перебор

Асинхронный перебор — полный алгоритм для распределенных задач с ограничениями. Характеристики:

- Глобальное упорядочивание агентов
- Нет процедуры останова (но она легко добавляется)
- Хорошо масштабируется

Каждый агент отвечает за одну переменную и предлагает другим агентам принять значение, которое он выбрал.

Расширения:

- AWCS — изменение порядка при конфликтах
- ADOPT, APO — решение задачи оптимизации
- DynAPO — динамическое добавление агентов

3.5 Распределенный локальный поиск

Алгоритмы распределенного локального поиска исследуют возможные изменения состояния:

- Всегда стремятся улучшить состояние (уменьшить количество конфликтов)
- Естественным образом поддерживают динамику (добавление ограничений, агентов)
- Эффективны по времени
- Не полны и требуют настройки параметров

Известные алгоритмы:

- Tabu search
- Simulated annealing
- Iterative Breakout method

3.6 Кооперативные алгоритмы

Популяция — набор индивидуальных агентов, где:

- Агент знает целиком исходную задачу (и может решить сам)
- Агенты координируются для нахождения решения

Известные алгоритмы:

- Эволюционные и генетические алгоритмы
- Particle Swarm Optimization
- Ant Colony Optimization

4 Коллективное поведение агентов

4.1 Модели коллективного поведения

Использование идеи коллективного поведения приводит к проблемам:

- Формирование совместных планов действий
- Учет интересов компаньонов агента
- Синхронизация совместных действий
- Наличие конфликтующих целей
- Конкуренция за совместные ресурсы
- Организация переговоров о совместных действиях
- Распознавание необходимости кооперации
- Выбор подходящего партнера
- Обучение поведению в коллективе
- Декомпозиция задач и разделение обязанностей

4.2 Основные направления исследований

Распределенный искусственный интеллект (РИИ) — ядро составляют исследования взаимодействия и кооперации небольшого числа интеллектуальных агентов. Главная проблема — разработка интеллектуальных групп и организаций, способных решать задачи путем рассуждений, связанных с обработкой символов.

Кооперативное распределенное решение задач (КРРЗ) — сеть слабо связанных между собой решателей, которые совместно работают в целях решения задач, выходящих за рамки индивидуальных возможностей.

Этапы распределенного решения задач:

1. Агент-менеджер проводит декомпозицию исходной проблемы на отдельные задачи
2. Задачи распределяются между агентами-исполнителями
3. Каждый агент-исполнитель решает свою задачу
4. Производится композиция, интеграция частных результатов

Искусственная жизнь — направление связано с трактовкой интеллектуального поведения в контексте выживания, адаптации и самоорганизации в динамичной, враждебной среде. Глобальное интеллектуальное поведение всей системы рассматривается как результат локальных взаимодействий большого числа простых агентов.

Основные положения:

1. МАС есть популяция простых и зависимых друг от друга агентов
2. Каждый агент самостоятельно определяет свои реакции на события

3. Связи между агентами горизонтальные (нет супервизора)
4. Нет точных правил для определения глобального поведения
5. Поведение на коллективном уровне порождается только локальными взаимодействиями

4.3 Модель кооперативного решения проблем (CPS)

Эта модель рассматривает взаимодействие агентов, построенных согласно BDI-архитектуре (Belief-Desire-Intention). В модели ментальные понятия формализуются с помощью операторов временной логики.

Четыре этапа формирования кооперативного решения:

1. **Распознавание** — агент распознает целесообразность кооперативного действия. Например, у агента имеется цель, достичь которую в изоляции он не способен.

Формальное определение потенциала для кооперации: По отношению к цели f агента i имеется потенциал для кооперации тогда и только тогда, когда:

- (1) имеется некоторая группа g , такая, что i верит, что g может совместно достичь f , и
- либо (2) i не может достичь f в изоляции
- либо (3) i верит, что для каждого действия a , которое он мог бы выполнить для достижения цели f , он имеет иную цель, влекущую невыполнение действия a

2. **Формирование группы агентов** — агент, установивший возможность совместного действия, ищет партнеров. При успешном завершении образуется группа агентов, имеющих совместные обязательства для коллективных действий.
3. **Формирование совместного плана** — агенты переговариваются с целью выработать совместный план, который по их убеждению приведет к желаемой цели. Переговоры являются механизмом выработки соглашения. Протокол переговоров есть распределенный алгоритм поиска соглашения.
4. **Совместные действия** — агенты действуют согласно выработанному плану, поддерживая взаимодействие согласно принятым на себя обязательствам.

4.4 Конфликты в многоагентных системах

Вероятность возникновения конфликтов в многоагентной среде является неизбежным следствием децентрализованности таких систем.

Конфликт — ситуация, в которой возникает противоречие вида:

$$p \wedge q \Rightarrow \text{false}$$

где p и q — убеждения агентов.

Виды конфликтов:

1. **В системе убеждений агентов** — могут возникать при получении агентом ложной информации от другого агента или информации, противоречащей убеждениям агента. Уровни поддержания целостности:

- Терминологический
 - Смысловой
 - Временной
2. **Обусловленные неполнотой модели** — неполнота имеющейся у агента модели окружающего мира и моделей других агентов. Связаны с понятием рефлексии агента.
 3. **Конкуренция за ресурсы** — конфликты, связанные с конкуренцией за совместные ресурсы или наличием противоречивости целей.

Разрешение конфликта — снятие логического противоречия за счет отбрасывания одной из альтернатив в соответствии с некоторым критерием, или смены p и q вместе.
Механизмы разрешения конфликтов:

1. Централизованный механизм (при наличии арбитра)
2. На основе правил поведения агентов — например, различные уровни компетентности агентов, при котором агент строит убеждение на основе информации из более компетентного источника
3. Недетерминированный вариант — использование рандомизации или жребия

Пример 1: Модель убеждений с приоритетами

Агенты обмениваются информацией с целью достичь соглашения. Убеждениям ставятся в соответствие приоритеты трех уровней:

- N (necessarily) — ограничения (constraints): Q необходимо истинно
- P (preferably) — предпочтения (preferences): Q предпочтительное убеждение
- O (optionally) — гипотезы (options): Q возможное убеждение
- $U - Q$ есть ложь

Степень доверия: $BD(Q) \in \{N, P, O, U\}$

Пример 2: Уровни компетентности агентов

Если:

- (1) b_1 есть убеждение агента a_1 , имеющего цель в роли r_1 , такое, что b_1 требуется для достижения этой цели, и
- (2) b_2 есть убеждение агента a_2 , имеющего цель в роли r_2 , такое, что b_2 необходимо для достижения этой цели, и
- (3) b_1 конфликтует с b_2 ,

Тогда: b_1 имеет больший уровень доверия, чем b_2 тогда, и только тогда, когда в соответствии с уровнем компетентности $(a_2 r_2) < (a_1 r_1)$.

4.5 Протоколы и языки координации

Протокол взаимодействия агентов определяет схему (распределенный алгоритм), по которой ведутся переговоры между агентами.

Основные схемы переговоров:

1. Теория речевых актов (Speech Act Theory)

Переговоры строятся с использованием небольшого числа примитивов:

- ASK — запросить
- TELL — сообщить
- REJECT — отклонить
- REQUEST — запросить выполнение
- COMMIT — взять обязательство
- NEGOTIATE — вести переговоры

Процесс начинается, когда агент посылает сообщение, содержащее его позицию по некоторому вопросу. Посредством обмена сообщениями агенты обсуждают тему и приходят к общему решению.

2. Протокол контрактных сетей (Contract Net Protocol)

Предназначен для координации в системах распределенного решения проблем. Схема работы:

- Заказчик рассылает объявление о вакансии на выполнение договора
- Потенциальные подрядчики отвечают на объявление
- Заказчик выбирает наиболее подходящего подрядчика
- Заказчик и подрядчик заключают договор (contract)

3. COOL (COOrdination Language)

Язык для управления совместными действиями. Важнейшие конструкции:

- Планы переговоров, определяющие состояния и правила
- Переговоры, определяющие текущее состояние плана

Типы сообщений:

- Информировать
- Запрашивать с ожиданием ответа
- Запрашивать без ожидания ответа

4. UNP (Unified Negotiation Protocol)

Механизм разрешения конфликтов, который позволяет повысить суммарную полезность, достигаемую агентами. Рассматриваются конфликтные ситуации, для разрешения которых используется механизм рандомизации (“бросание монетки”).

5 Архитектура многоагентных систем

5.1 Общие принципы

При выборе архитектуры многоагентной системы необходимо иметь в виду два ее аспекта:

1. Архитектуру взаимодействия агентов в процессе функционирования системы в целом
2. Архитектуру отдельного агента

Архитектура существенно зависит от:

- Концептуальной модели агента
- Принятого формализма и языка спецификаций
- Математической модели кооперации агентов
- Целевого приложения или класса приложений

5.2 Архитектура взаимодействия системы агентов

Выделяют два основных варианта:

5.2.1 Вариант 1: Одноуровневая архитектура

Агенты не образуют иерархии и решают общую задачу полностью в распределенном варианте.

Пример: Система планирования совещаний (встреч)

Особенность задачи: расписание составляется в контексте уже существующих назначений каждого участника. Информация носит личный характер и участники предпочитают ее не раскрывать.

Каждый участник представляется своим агентом (“электронным секретарем”). Стратегия поиска решения использует переговоры для поиска глобально согласованного решения.

Процедура согласования:

1. **Начальный вызов** — Агент-инициатор устанавливает контакт с агентами участников, указывая параметры встречи (дата, время, длительность, тема). Каждый секретарь спрашивает пользователя о заинтересованности и отвечает инициатору, указывая “фактор ограничений”:

- вес = 1, если запланированная встреча может быть передвинута
- вес = 2, если встреча не может быть передвинута

2. **Схема переговоров** — Самый простой случай: все согласившиеся агенты передали фактор ограничений = 0. В противном случае:

- Агент с наибольшим фактором ограничений получает статус MC (Most Constrained) и роль координатора
- Остальные получают статусы MC2, MC3, ..., LC (Least Constrained)

3. **Процесс согласования** — Очередные агенты делают свои ходы. Если агент вынужден поменять решение, делается откат процесса и согласование начинается вновь.
4. **Завершение** — Процесс заканчивается, когда достигнута цель, либо встреча организована в ограниченном составе, либо отменена.

5.2.2 Вариант 2: Иерархическая архитектура

Координация поддерживается специально выделенным агентом мета-уровня.

Место встречи агентов (AMP — Agent Meeting Place) — агент, играющий роль брокера между агентами, запрашивающими ресурсы, и агентами, которые эти ресурсы могут предоставить.

Компоненты AMP:

1. Объекты базовых сервисов — удаленный вызов объектов, упорядочение, дублирование объектов
2. Связные порты — прием и отправка агентов в AMP
3. Компонента установления подлинности — опознание агента, авторизация
4. Консьерж — контроль полномочий агента, наличия запрашиваемого сервиса, помощь в выборе маршрута
5. Поверхностный маршрутизатор — интерфейс между агентами и компонентами AMP
6. Лингвистический журнал — база данных для взаимопонимания агентов и AMP (словари и языки)
7. Глубинный маршрутизатор — ассистирует поверхностному при сложных запросах
8. Менеджер ресурсов — регистрирует агентов и управляет ресурсами AMP
9. Среда исполнения агента — интерпретирует сценарии, обеспечивает доступ к базовым возможностям
10. Система доставки событий — регистрирует события и выполняет поиск агентов для соответствующего типа событий

5.3 Архитектура отдельного агента

Классификация по парадигме:

1. **Архитектура, основанная на знаниях (Deliberative Agent Architecture)**
Содержит символьную модель мира, представленную в явной форме. Принятие решений осуществляется на основе рассуждений логического или псевдо-