

Архитектура многоагентных систем

Дисциплина «Проектирование крупных программных систем и баз данных»

Студент: Машеров Д.Е.
Группа: А-13-08

Интеллектуальный агент

- ▶ Интеллектуальный агент — программа, самостоятельно выполняющая задание, указанное пользователем компьютера, в течение длительных промежутков времени.

- ▶ Агенты используются для содействия оператору или сбора информации. Одним из примеров заданий, выполняемых агентами, может служить задача постоянного поиска и сбора необходимой информации в Интернете.

Что не является агентом?

- ▶ **Массовые технологии программирования**
 - ▶ Мастера (Wizards)
 - ▶ Скриптовые языки
 - ▶ Языки программирования
 - ▶ Объекты
 - ▶ Классы

- ▶ **Отдельные методы ИИ**
 - ▶ Системы, основанные только на правилах
 - ▶ Нейросети
 - ▶ Планировщики

Что является агентом?

- ▶ Вирусы
- ▶ Интеллектуальные помощники
- ▶ Поисковые боты
- ▶ Чат-боты
- ▶ NPC (неигровые персонажи) игр
- ▶ Роботы

Многоагентная система

- ▶ Многоагентная система (МАС, англ. Multi-agent system) — это система, образованная несколькими взаимодействующими интеллектуальными агентами.
- ▶ МАС могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы.
- ▶ Примеры: онлайн-торговля, ликвидация чрезвычайных ситуаций, и моделирование социальных структур.

Искусственный интеллект

Причины использования:

- ▶ Распределенный ИИ
 - ▶ Распределенные базы знаний
 - ▶ Распределенный логический вывод и решение задач
- ▶ Интеграция отдельных методов ИИ в агента, способном воспринимать информацию среды и совершать действия
- ▶ Новая концепция интеллекта
 - ▶ Корни интеллекта связаны с культурой и обществом
 - ▶ Общее разумное поведение формируется совместными действиями из большого числа простых полуавтономных индивидуумов

Решает задачи:

- ▶ Микроуровень – «мозг» агента
- ▶ Макроуровень – совместное планирование и решение задач

Разработка ПО

Причины использования:

- ▶ Агенты как абстракция и средство анализа и построения еще более сложных систем
- ▶ Многоагентная архитектура – автономная компонентная архитектура построения интеллектуальных систем

Решает задачи:

- ▶ Дешевая и качественная реализация МАС
- ▶ Методологии
- ▶ Инструменты
- ▶ Платформы

Распределенные системы и сети

Причины использования:

- ▶ Новый уровень управления программной и аппаратной инфраструктурой

Решает задачи:

- ▶ Архитектуры производительных агентных платформ
- ▶ Координация действий (например, в Grid-системах)

Характеристики многоагентной системы

- ▶ Агенты, входящие в состав многоагентной системы зачастую разные
- ▶ Окружающая среда с точки зрения агентов – динамическая
- ▶ Коллективная информация, доступная агентам – распределенная
- ▶ Управление в многоагентной системе – распределенное
- ▶ Мир для каждого агента – частично наблюдаемый, знание - распределенное

Преимущества многоагентных систем

- ▶ **Увеличение производительности и эффективность** – из-за асинхронного и параллельного исполнения
- ▶ **Устойчивость к сбоям и надежность** – вся система продолжает работать если один из ее компонентов выходит из строя
- ▶ **Масштабируемость и гибкость** – легко добавлять новых агентов в систему
- ▶ **Стоимость** - цена одного агента мала по сравнению со всей системой
- ▶ **Повторное использование** – модульное программное обеспечение легче разрабатывать и поддерживать

Формальное определение MAS

▶ **MAS = (A, E, R, ORG),**

A – множество агентов;

E = {e} – среда, в которой находится данная MAS;

R – множество взаимодействий между агентами (например, коммуникативные акты, аукционы и т.п.);

ORG – множество базовых организационных структур, соответствующих конкретным функциям (ролям) агентов и установившимся отношениям между ними.

Агентные среды

- ▶ Агент должен находиться в среде
- ▶ Действия агента выполняются в среде
- ▶ Объект восприятия агента является частью среды
- ▶ Среда имеет множество параметров среды, влияющих на архитектуру агента
- ▶ Параметры определяются с точки зрения агента

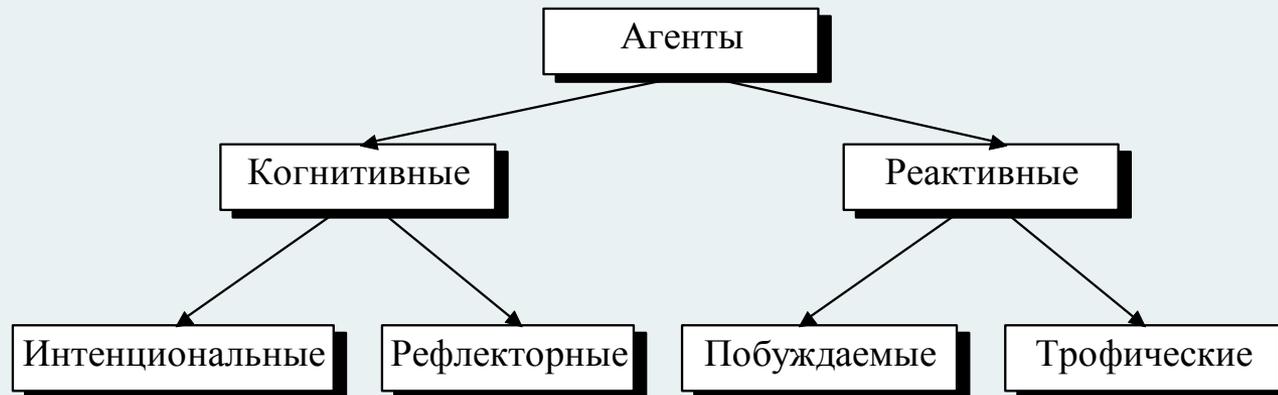


Свойства среды

- ▶ Детерминированная/случайная
- ▶ Эпизодическая/последовательная
- ▶ Дискретная/непрерывная
- ▶ Статичная/динамичная
- ▶ Одноагентная/многоагентная



Классификация агентов



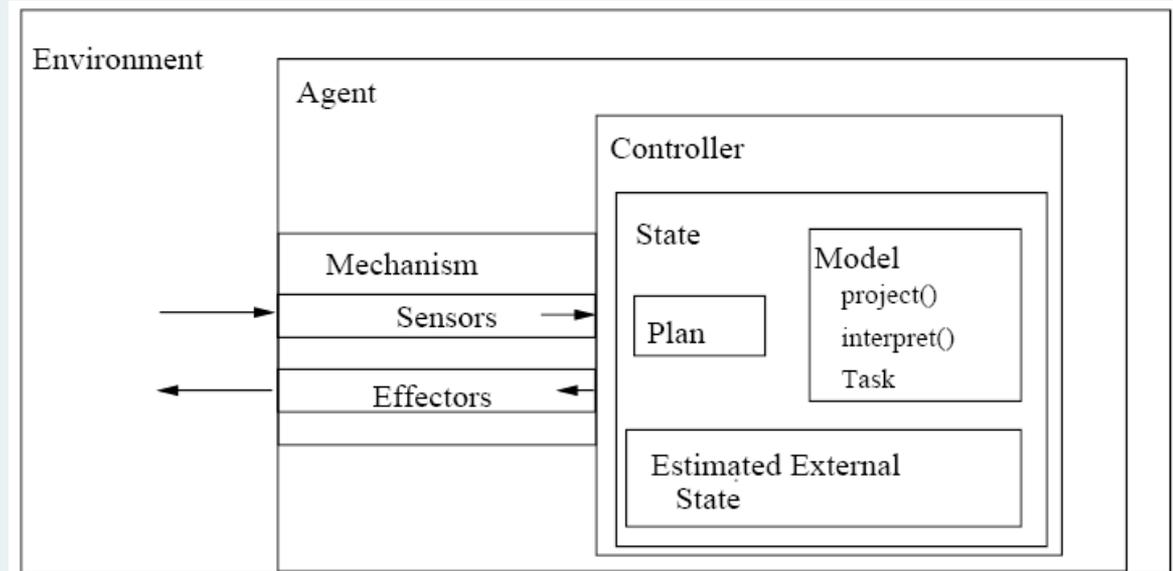
Архитектуры агентов

Делиберативная архитектура

- ▶ Используется формальное представление картины окружающей среды в символической форме
- ▶ Решения принимаются на основе формальных рассуждений

Преимущества и недостатки делиберативной архитектуры

- Удобство символического представления знаний
- Существуют языки манипуляции формальными представлениями (Prolog, JESS)



- ▶ Доказательство теорем в логике предикатов 1-ого порядка может продолжаться неограниченно долго
- ▶ Создание точной и полной модели мира, процессов и механизмов рассуждения представляют определенные трудности

Реактивная архитектура

- ▶ Реактивные агенты не имеют какой-либо символической внутренней модели мира
- ▶ Реактивные агенты работают по правилам типа *ситуация-действие*, выбирая из них наиболее подходящие действия к конкретной ситуации



Преимущества реактивной архитектуры

- ▶ Позволяет наилучшим образом использовать множество образцов поведения для реакции агента на определенные стимулы для конкретной предметной области
- ▶ Простота
- ▶ Устойчивость к сбоям
- ▶ Экономичность (в смысле вычислительного времени)
- ▶ Вычислительная разрешимость

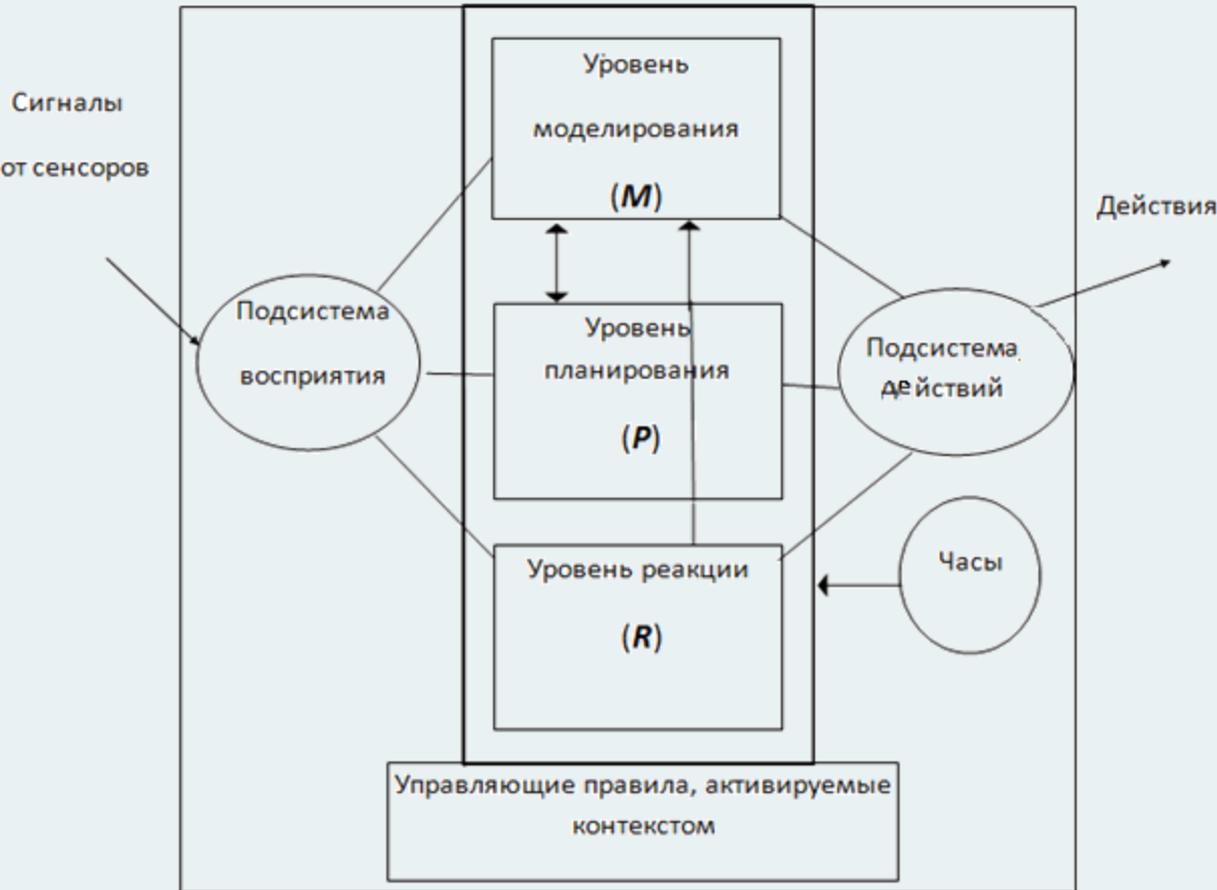
Недостатки реактивной архитектуры

- ▶ Необходимость полного ситуационного определения всех возможных активностей агентов
- ▶ Агенты без модели среды должны иметь достаточно информации от локальной среды
- ▶ Сложность создания обучающихся агентов
- ▶ Сложность в проектировании агентов с многочисленными вариантами поведения (динамическим взаимодействием)

Архитектуры агентов

Архитектура Touring Machines

(Многоуровневая архитектура для автономного агента)



Три уровня, каждый из которых соответствует различным типам способностей агента.

-уровень реакции на события **R** поддерживает способность агента быстро реагировать на события, выдаваемые вышележащим уровнем, даже если они ранее не планировались;

-уровень планирования **P** генерирует, исполняет и динамически реконструирует частичные планы, например, для выбора маршрута подвижного робота;

-уровень предсказания, или моделирования **M** моделирует поведение сущностей внешней среды и самого агента, что может использоваться для объяснения наблюдаемого поведения и предсказания возможного их поведения в будущем.

Архитектуры агентов

IDS-архитектура (Inhabited Dynamic Systems)



-Первый уровень - это уровень процессов, на котором периодически выполняются с заданной частотой некоторые вычисления, а также осуществляется управление процессами восприятия и исполнения.

-Второй уровень(уровнем ответной реакции), вычисляет ответную реакцию на асинхронные события, которые либо воспринимаются уровнем процессов, либо им генерируются.

-Уровень анализа выполняет символические рассуждения, такие, как предсказание, планирование и перепланирование.

Взаимодействие агентов

- ▶ Для решения глобальной задачи агенты должны уметь *сотрудничать, координировать действия и вести переговоры*

- ▶ Взаимодействие означает установление двусторонних динамических отношений между агентами. При этом оно является одновременно источником и продуктом некоторой организации.

- ▶ Основные ситуации взаимодействия агентов в МАС можно охарактеризовать с помощью четырех основных критериев:
 - 1) совместимость целей или намерений агентов;
 - 2) отношение агентов к ресурсам;
 - 3) опыт агентов; связанный с некоторой проблемной областью;
 - 4) обязательства агентов друг перед другом.

Типы взаимодействия агентов

ТИП СИТУАЦИИ ВЗАИМОДЕЙСТВИЯ АГЕНТОВ	Совместимость целей	Потребность в чужом опыте	Наличие распределенных ресурсов
Координируемое сотрудничество	+	+	+
Простое сотрудничество	+	+	-
Непродуктивное сотрудничество	+	-	+
Безразличие (независимость)	+	-	-
Коллективное соперничество за ресурсы	-	+	+
Чистое коллективное (командное) соперничество	-	+	-
Индивидуальное соперничество за ресурсы	-	-	+
Чистое индивидуальное соперничество	-	-	-

Проблемы и трудности построения многоагентных систем

- ▶ Как произвести декомпозицию задачи и выделить подзадачи для агентов?
- ▶ Как реализовать децентрализованное управление?
- ▶ Как реализовать эффективные алгоритмы планирования и обучения?
- ▶ Как представить знание? Как заставить агентов адекватно реагировать на действия, планы и знания других агентов?

Проблемы и трудности построения многоагентных систем

- ▶ Как сделать возможным коммуникации между агентами? Какие языки и протоколы использовать?
- ▶ Как сделать возможным переговоры между агентами и разрешение конфликтов?
- ▶ Как заставить агентов формировать организационные структуры, такие как команды или коалиции?
- ▶ Как быть уверенным в стабильности и когерентности системы?

Заключение

В презентации рассмотрены:

- ▶ понятия интеллектуального агента и многоагентных систем(МАС)
- ▶ Некоторые архитектуры агентов
- ▶ Достоинства и трудности построения МАС