

Распределенная архитектура

В настоящее время реализуются системы *параллельного и распределённого имитационного* моделирования.

Параллельное моделирование предполагает, что имитационная модель выполняется на нескольких процессорах.

Выполнение имитационной модели на нескольких компьютерах, объединённых в сеть, - это распределённое имитационное моделирование.

Объединение нескольких имитационных систем в одну распределённую среду вызвано необходимостью связи нескольких тренажёров, имитирующих управление автомобилем, танком, коллективом и т.д.

Например, моделирование транспортной инфраструктуры города следует рассматривать неразрывно с подсистемой электроснабжения и с другими инфраструктурами.

Распределенная архитектура

Возможные варианты распределения вычислений между узлами:

А) Выполнение параллельно с ходом имитационного эксперимента специализированных функций: генерация псевдослучайного числа, управление списком будущих событий, сбор статистических данных. Выигрыш во времени моделирования в этом случае получается небольшим.

Б) При декомпозиции иерархической модели произвести декомпозицию события на подсобытия (sub-event). Эти подсобытия следует выполнять параллельно (иерархическая декомпозиция). Выигрыш во времени при декомпозиции зависит от самой модели.

В) Несколько имитационных прогонов последовательной имитационной модели при эксперименте выполняют на нескольких процессорах (распараллеливание репликаций).

Г) Распределение на уровне объектов (процессов) означает, что часть объектов одной модели выполняется на одном сервере, а часть на других с необходимостью синхронизации модельных часов.

Распределенная архитектура

Алгоритмы синхронизации делятся на консервативные и оптимистические. В оптимистическом алгоритме все модельные процессы идут независимо, но если процесс получает синхро-событие, имеющее временную отметку меньшую, чем уже обработанные события, он выполняет откат и обрабатывает свои события повторно, но теперь в обновленном хронологическом порядке.

Имитационные модели объединяются с помощью дополнительных программных средств (HLA, High Level Architecture).

Компонентами моделирования являются федераты, а объединение федератов называют федерацией.

Федераты одной федерации могут быть разнородными.

Обмен данными между федератами и исполнение федератов в едином модельном времени выполняется с помощью программной оболочки RTI.

Интерфейс HLA представляет разработчику низкоуровневый набор примитивов, который удобен для сопряжения имитационных моделей, реализуемых различными разработчиками.

Распределенная архитектура

Архитектура HLA была разработана по заказу Министерства обороны США для унификации и повторного использования моделей, применяемых в военных целях. Потом эта архитектура была стандартизована институтом IEEE как стандарт 1516, например, версия IEEE 1516-2010.

HLA представляет объектно-ориентированную систему, которая может применяться для построения моделей на различных языках, поддерживающих концепцию ООП (C++, Java и т.п.).

Модель, основанная на архитектуре HLA – федерация – состоит из независимых федератов – основных модельных блоков. Требования HLA ограничивают возможности взаимодействия между федератами только средствами, предоставляемыми HLA RTI (Run-Time Infrastructure). Всё необходимое для работы отдельно взятого федерата в составе любой модели может быть описано и документировано в терминах, определяемых стандартом HLA и не должно зависеть от конкретной реализации RTI. HLA не накладывает никаких ограничений на внутреннюю организацию федератов (составных моделей).

Распределенная архитектура

Название RTI	Разработчик	Версия стандарта HLA	Тип лицензии
ARTIS GAIA	University of Bologna	DMSO 1.3	Открытый код
CERTI	ONERA	DMSO 1.3, IEEE 1516 2000	GPLv2
EODiSP	P&P Software	IEEE 1516 2010	GPL
MAK	MAK Technologies	DMSO 1.3, IEEE 1516 2000, HLA Evolved (2010)	Коммерческая
NCWare	Nextel	IEEE 1516 2000	Коммерческая
Portico	Portico	DMSO 1.3, IEEE 1516 2000	CDDL
pRTI	Pitch Technologies	DMSO 1.3, IEEE 1516 2000, HLA Evolved (2010)	Коммерческая
RTI NG Pro	Raytheon	DMSO 1.3, IEEE 1516 2010	Коммерческая
ИМС СКМ	РусБИТех	IEEE 1516 2010	Коммерческая

Распределенная архитектура

Тренажеры «Вертолет», «Самолет», ...



...



Тренажеры «БМП», «Танк», «ЗРК» ...

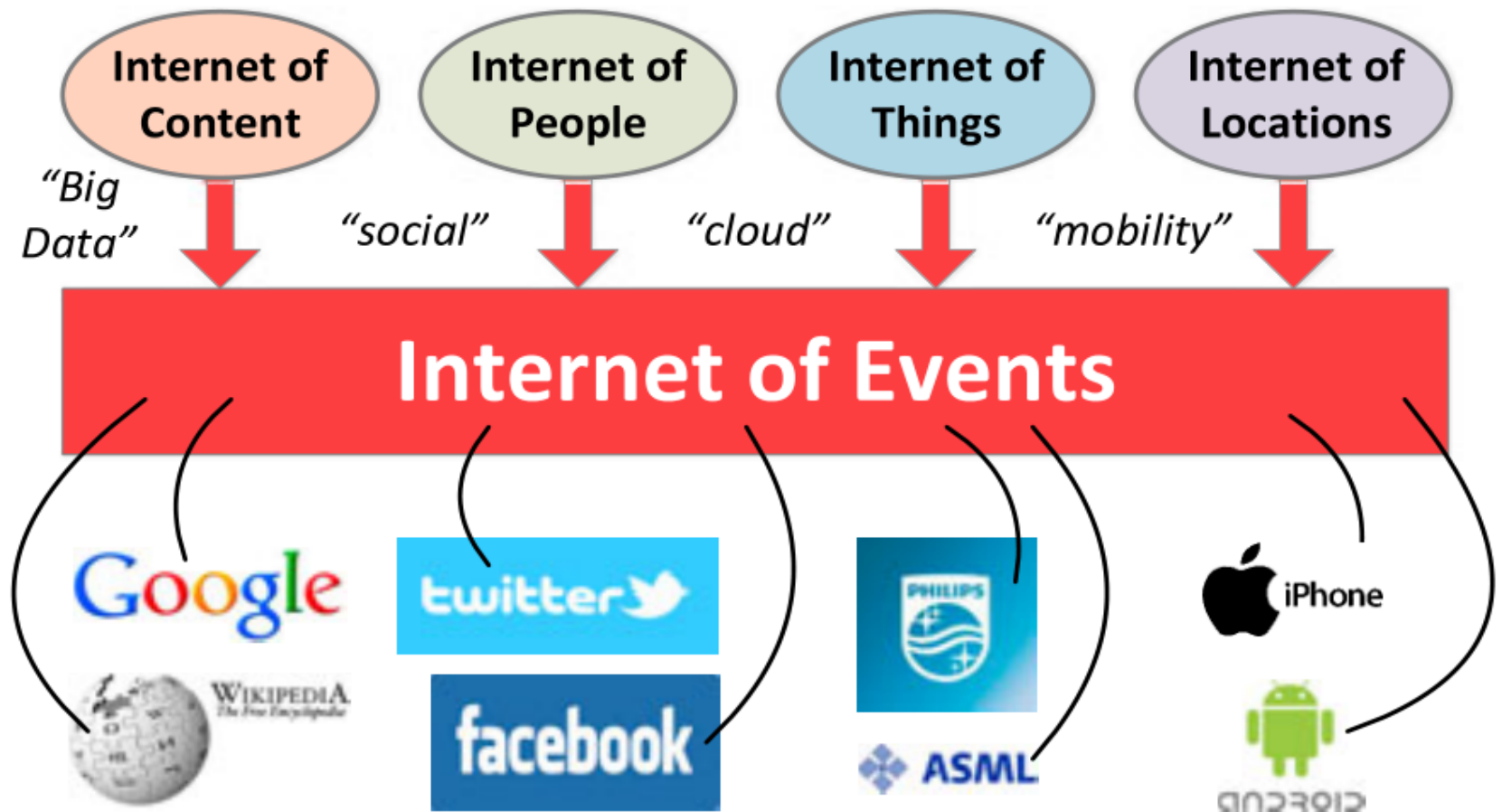
Штаб руководства обучением

Моделирующая система

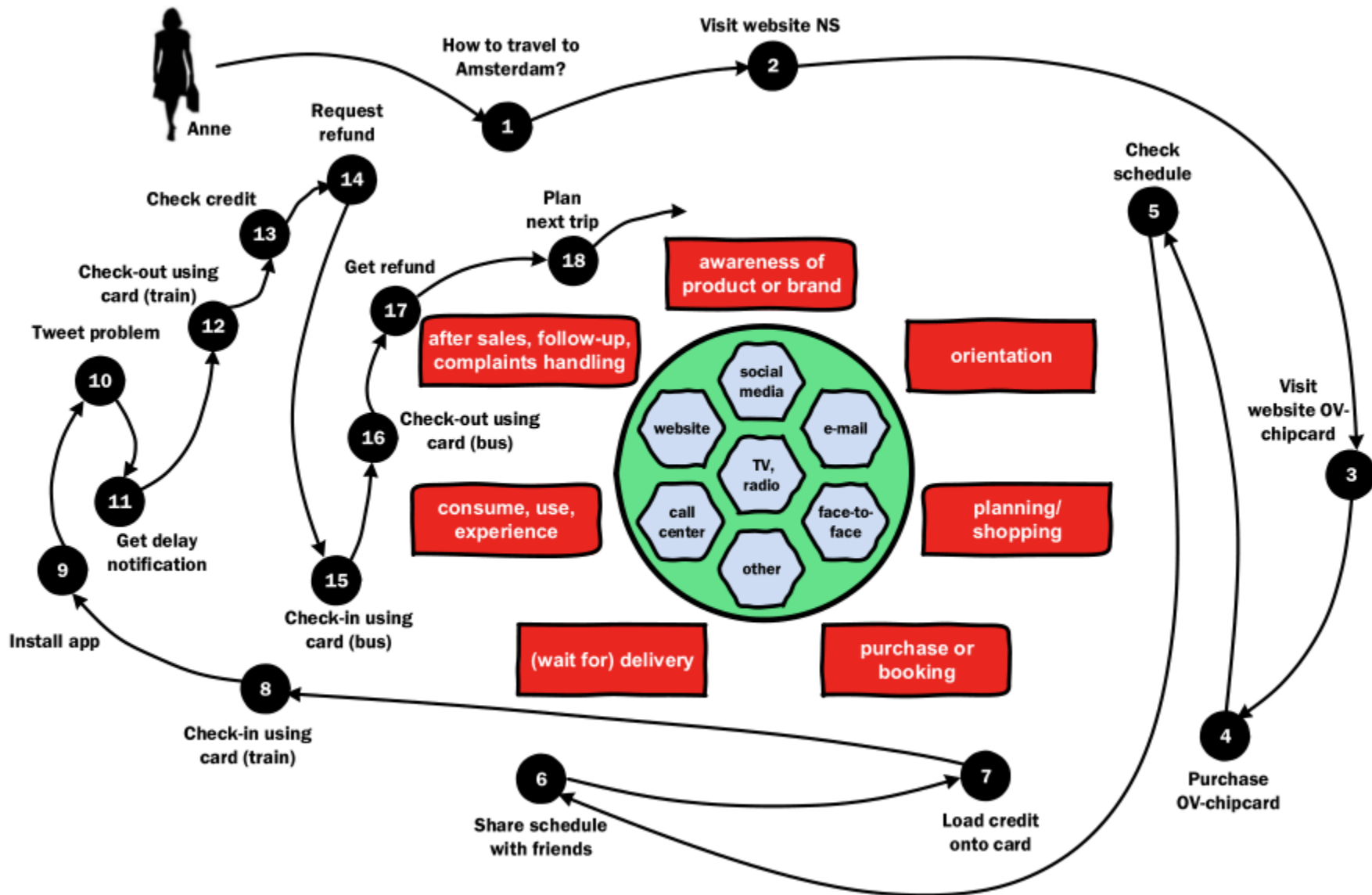


3D-визуализатор

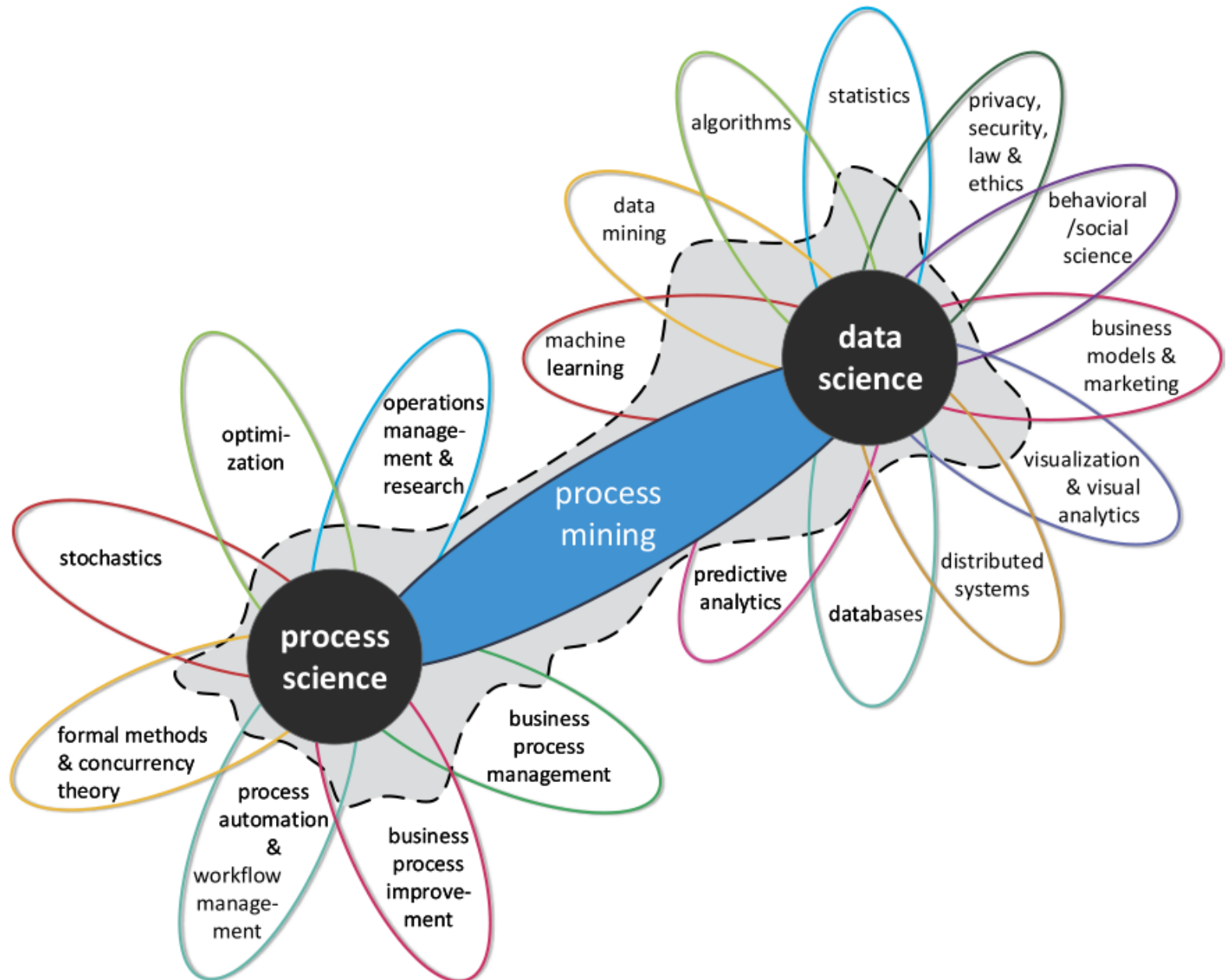
Множество источников данных



Множество событий и процессов



Изучение реальных процессов – process mining



Моделирование процессов

Класс! Остается
это реализовать



Имитационное моделирование – составная часть технологий искусственного интеллекта

Пять ключевых областей



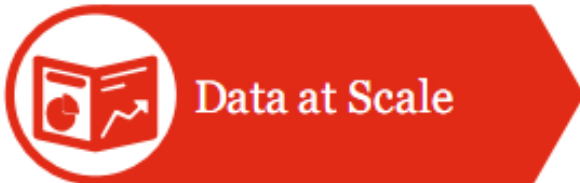
Автоматизация рабочих процессов
Поддержка принятия решений
Обнаружение аномалий



Анализ неструктурированных данных для аудио, изображений и текста
Системы автоматизации и обучения



Извлечение информации, автоматизация рабочего процесса и генерация текстов
Знания из неструктурированных данных
Чат-боты и IPA



Промышленный интернет вещей
Потоковая передача данных в режиме реального времени
Масштабная архитектура обработки данных



Операционный анализ
Поддержка стратегий
Моделирование поведения потребителей
Анализ выхода на новый рынок

Цифровая цепочка поставок, интеллектуальная фабрика и другие промышленные процессы - используют технологию искусственного интеллекта. Имитационные модели этих систем также будут включать компоненты искусственного интеллекта.

ИИ приложения

Маршрутизация продукции

Планы загрузки производства

Принятие решений в режиме реального времени на основе данных датчиков

Диспетчерская в режиме реального времени

ИИ + модели приложения

Компоненты искусственного интеллекта из реальной системы встроены в имитационную модель



Цифровые двойники в реальном времени, системный анализ “что, если”

Компоненты искусственного интеллекта для реальной системы разрабатываются с использованием имитационной модели



Проектирование систем

Методы искусственного интеллекта могут решать сложные проблемы оптимизации/калибровки в крупномасштабных моделях на основе агентов

Крупномасштабные имитационные мультиагентные модели имеют:

- множество параметров для калибровки
- длительное время работы

Традиционные подходы к оптимизации могут потерпеть неудачу, поскольку занимают много времени

Машинное обучение и интеллектуальный выборочный подход могут быть использованы для создания метамоделей, которые значительно ускоряют калибровку:

- мета-модели работают намного быстрее, чем полная имитационная модель
- машинное обучение позволяет фиксировать нелинейное поведение и разрывы в мультиагентных моделях

Распространенным применением моделирования является разработка “оптимальных” правил принятия решений для агентов в сложной системе.

Это проблема, которую можно решить с помощью технологий искусственного интеллекта в сочетании с моделированием.

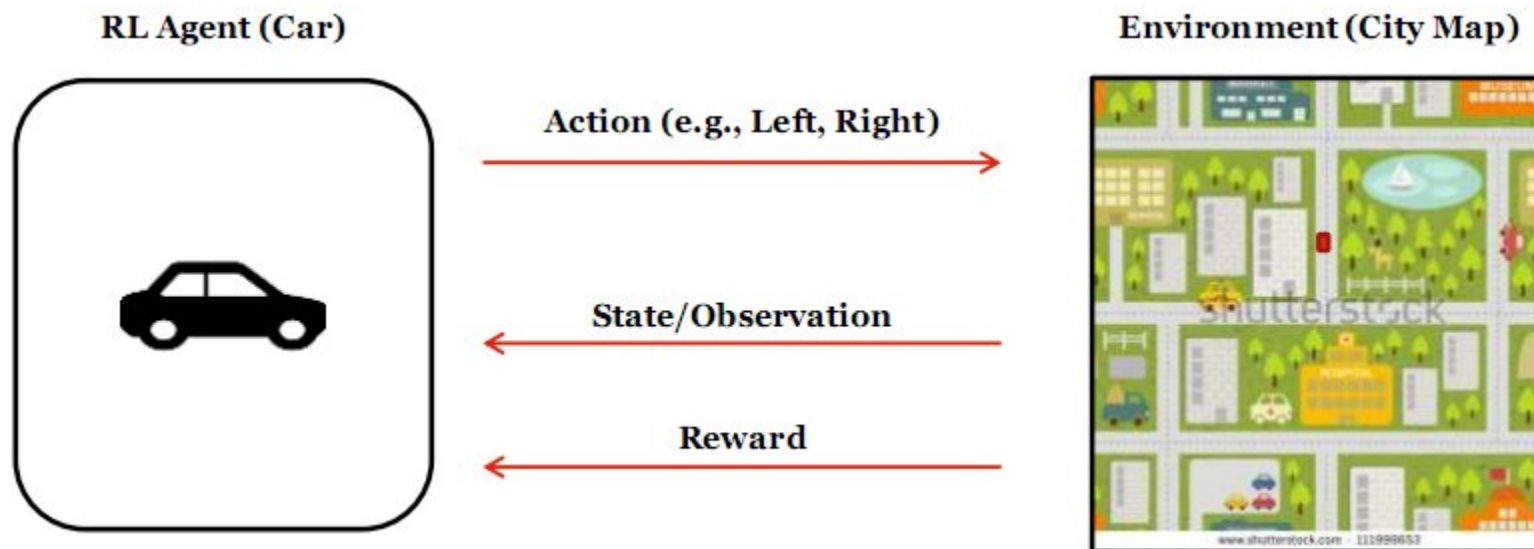
Многие бизнес-проблемы связаны с тем, что несколько лиц, принимающих решения, либо сотрудничают, либо конкурируют за достижение определенной цели.

Например, для управления крупной автомобильной компании, которая внедряет автономный транспорт для перевозки клиентов, используются очень сложные модели AnyLogic, имитирующие транспортные средства, которые при принятии решений следуют закодированным вручную правилам.

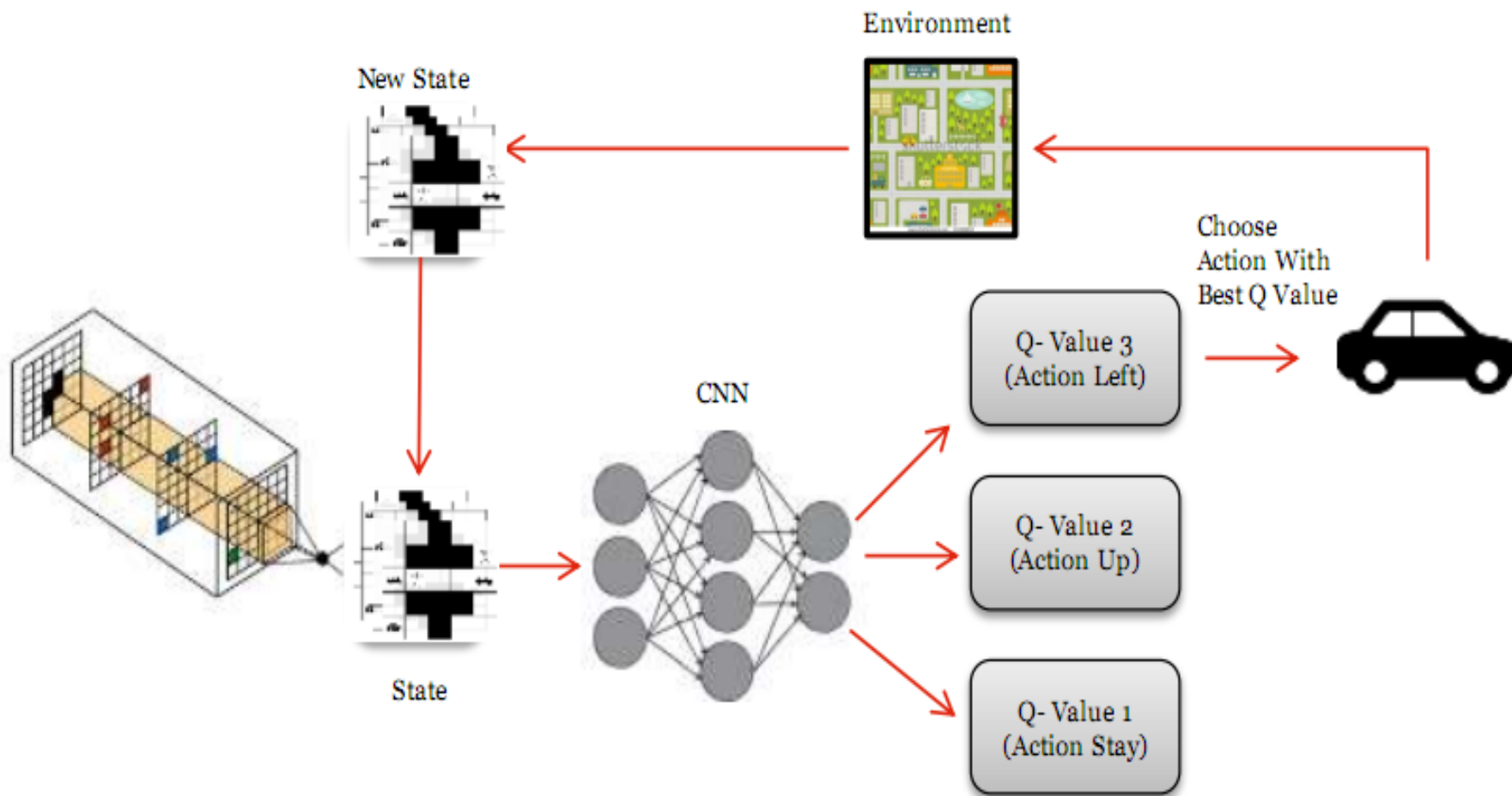
Дополнительно использовано Глубокое обучение с подкреплением (DQN) для обучения транспорта максимизации эффективности автопарка при одновременном удовлетворении спроса клиентов на поездки.

Обучение с подкреплением (Reinforcement Learning) – область машинного обучения, направленная на обучение компьютера политике принятия оптимальных решений с течением времени, используя вознаграждения и штрафы в качестве сигналов.

RL-агент и среда (Environment) – RL состоит из двух сущностей: агента, принимающего решения, и мира (среды), в котором он действует. Агент сообщит о своих действиях этому миру, и в ответ (результат) получит вознаграждение и новое наблюдение (состояние).



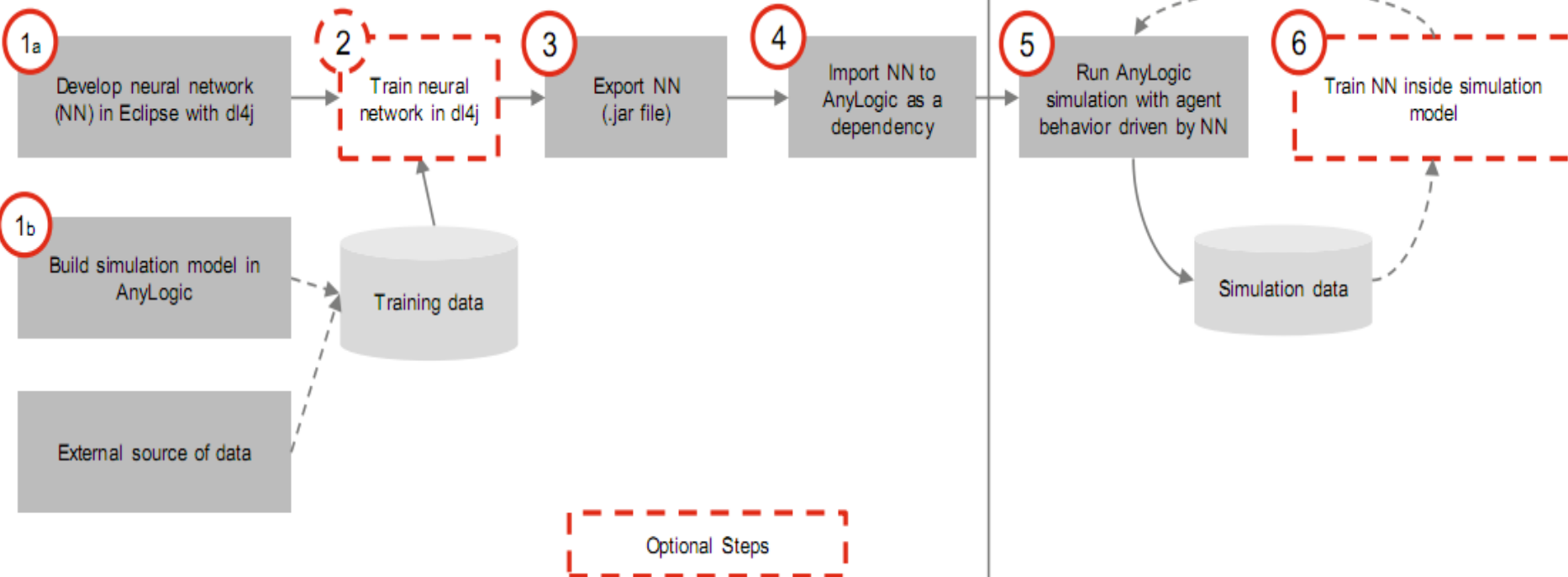
Глубокое обучение с подкреплением (Deep Q Network) используя сверточную нейросеть (CNN), действует как функция Q-значения для определенного состояния и выводит значение Q для каждого действия. В зависимости от политики действий агент выбирает действие, которое дает наибольшее Q-значение, и выполняет его в среде.



Модель глубокого обучения создается во внешней системе и может быть обучена перед интеграцией в AnyLogic

Build

AnyLogic Simulation



Пример на модели сегрегации Шеллинга

Агенты имеют критическую оценку для числа соседей
-единомышленников и размещаются в случайном месте

Моделирование на основе правил

-или-

управляемое нейронной сетью

Агенты оценивают, достаточно ли соседей-единомышленников, каждый раз, когда они переезжают или меняется сосед. Агенты также могут перемещаться случайным образом без всякой причины

Нейронная сеть обучается на "исторических данных о перемещениях", т.е. данных, сгенерированных в результате моделирования на основе правил

Если агентов-единомышленников достаточно, агент остается на своем текущем местоположении

Если агентов-единомышленников недостаточно, агент перемещается в случайное место

Учитывая количество единомышленников и пороговое значение, нейронная сеть делает решение переезжать или не переезжать

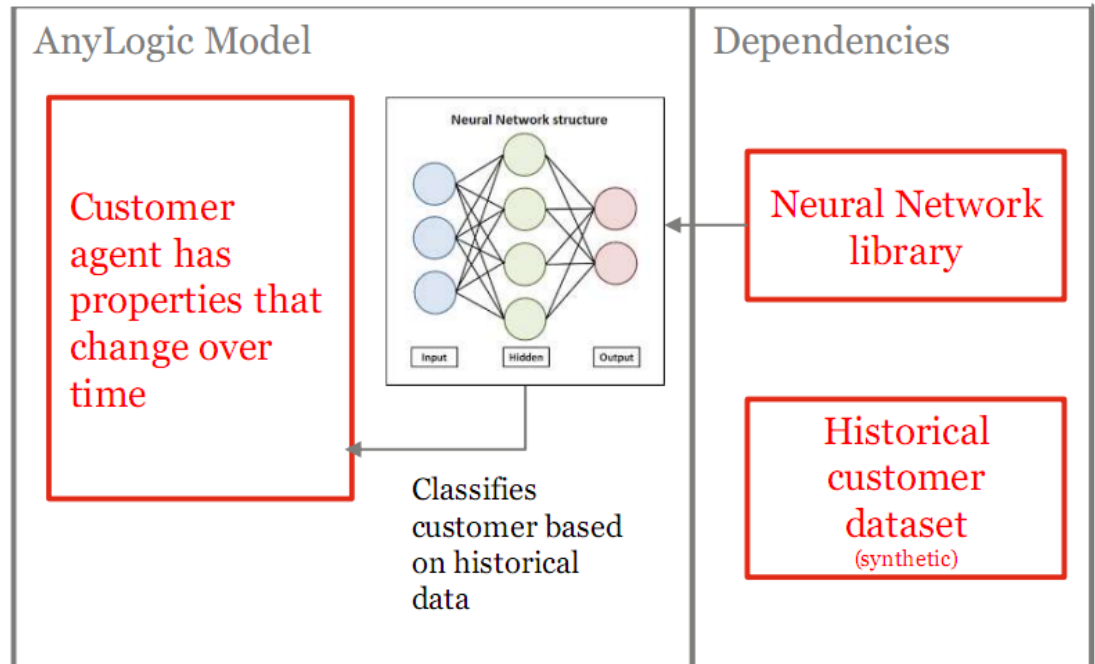
Обе модели сходятся, но в моделировании, управляемом нейронной сетью, результат получается без предварительного знания взаимосвязи между пороговым значением и количеством соседей-единомышленников

Перспективы ИМ и ИИ

- ❖ Всепроникающий ИИ в реальном мире означает, что у нас будет всепроникающий ИИ и в нашем моделируемом мире
- ❖ Искусственный интеллект не только появляется в наших моделях, но и поможет нам их создавать
- ❖ Архитектура AnyLogic позволяет практически интегрировать компоненты искусственного интеллекта непосредственно в моделирование

Модель классификации и приоритезации клиентов

При замене бизнес-правил для определения статуса клиента нейронная сеть обнаруживает тенденции в данных и классифицирует каждого клиента



Применение ИМ с ИИ в бизнесе

В *ElementAI* смоделировали работу продуктового магазина для прогнозирования спроса и пополнения товаров на полках.

1ой целью была генерация данных о поминутном спросе за 5-летний период с учетом колебаний и случайных событий. Важно создать достаточно данных для прогнозирования временных рядов, чтобы алгоритм ИИ обучался с учетом сложности параметров.

2ой целью использования ИИ была помощь менеджменту в определении приоритетов продуктов при заполнении полок, чтобы свести к минимуму убытки.

Такая модель продуктового магазина включает агентов 3 типов с набором стохастических параметров: покупателей, сотрудников, категории продуктов.

Менеджер магазина получает прогноз, что будет распродано в течение следующего часа, т.е. что надо дозаказать со склада ...

- с точностью 60%, если будет ориентироваться на предыдущий час;
- с точностью 80%, если будет использовать это ИИ-решение.

Применение ИМ с ИИ в бизнесе

Разработанное ИМ+ИИ решение состоит из 4 шагов:

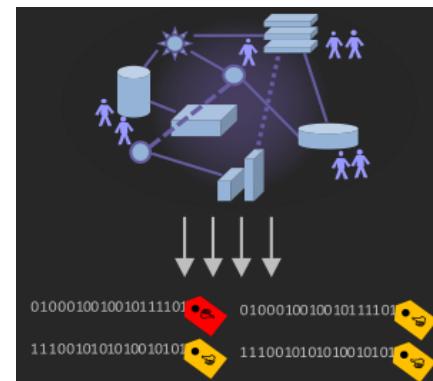
- ✓ запустить имитационную модель на несколько часов модели;
- ✓ поставить модель на паузу и вывести данные о запасах продуктов в текстовый файл;
- ✓ подключить внешний модуль с ИИ;
- ✓ создать список приоритетных товаров для пополнения запасов в текстовом файле;
- ✓ снова запустить модель с обновлёнными параметрами и повторять итерации обучения.

Стратегия	Прибыль (\$)	Относительная прибыль
Приоритет по очереди	16 685.67	-
Приоритет по стоимости продукта	16 783.47	+0.6%
С перспективой на час (прогноз)	17 427.56	+4.4%
Приоритет по потенциальному прогнозируемому доходу	17 517.09	+5%

Применение ИМ в разработке ИИ

А) Синтетические данные

генерация чистых, детальных, помеченных данных, покрывающих все возможные сценарии



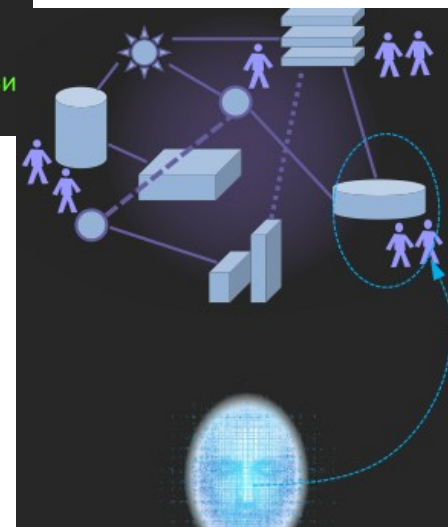
Б) Среда для обучения

дешевая безрисковая среда для глубокого обучения с подкреплением (Deep Reinforcement Learning, DRL)



В) Среда для тестирования

дешевая безрисковая среда для тестирования эффективности ИИ (обученного там же или вовне) путём интегрирования его в модель. ИМ позволяет сравнить ИИ с другими алгоритмами.



Применение ИМ в разработке ИИ

Обученная нейросеть для адаптивного управления может быть заменена на эвристические алгоритмы, которые, *возможно*, покажут лучшую производительность для некоторых задач.

Однако, по мере роста сложности системы (например, дорога со множеством перекрёстков) преимущество ИИ будет все более явным.

Архитектурные варианты решения для *ML* с использованием ИМ

- 1) импортируем инструменты ИИ в среду моделирования
- 2) импортируем имитационную модель в среду разработки ИИ
- 3) имитационная модель выполняется в Облаке как сервис,
ИИ взаимодействует с ней через API

Проблемы и трудности ИИ с ИМ

- ❖ ИМ выполняется медленно (единичный прогон не распараллеливается и не использует GPU)
- ❖ существуют технологии обучения с подкреплением на множестве параллельных прогонов, которые можно организовать в облаке;
- ❖ пространство состояний и действий велико, сходимость обучения ИИ может быть очень медленной;
- ❖ модель отражает только то, что заложено в неё разработчиком (плохая модель -> плохой ИИ);
- ❖ хорошая модель представляет задачу гораздо чище, чем реальные данные!
- ❖ требуется опыт моделирования;
- ❖ много неформальных шагов, “искусства”, интуиции:
выбор состояния / наблюдения, частота циклов обучения,
функция награды, структура нейросети и её гиперпараметры.

*Разработчики ИИ предпочитают бесплатный открытый код,
но для бизнес-моделирования такого софта нет
(или делать модель на C#, Java, Python, C++)*