

Агентно-ориентированный подход

Материал из Википедии — свободной энциклопедии

Агентно-ориентированный подход (в дальнейшем АОП) к программированию — разновидность представления программ или парадигма программирования, в которой основополагающими концепциями являются понятия агента и его ментальное поведение, зависящее от среды, в которой он находится. Концепция была предложена Шохемом (англ. *Yoav Shoham*) в 1990 г.^[1]. Определение парадигмы, данное автором:^[2]

«*Эту новую парадигму программирования вполне разумно назвать рациональным программированием. Точно так же, как объектно-ориентированное программирование сдвинуло парадигму с написания процедур к созданию объектов, рациональное программирование сдвинуло парадигму с создания информационных объектов к созданию мотивированных агентов.*»

Агентом является всё, что может рассматриваться как воспринимающее свою **среду** с помощью **датчиков** и воздействующее на эту среду с помощью **исполнительных механизмов**.^[3]

Содержание

Значение термина «агент»

Связанные понятия

Связь АОП и ООП

От задачи к реализации

Причины возникновения

Постановка задачи

Реализации

Структура агента

Структура поведения

Примеры

JADE

Примечания

См. также

Литература

Значение термина «агент»

Шохем, представляя агентно-ориентированный подход, как новую парадигму достаточно подробно обсуждает в каком смысле в ней используется слово *агент*. Дело в том, что данное слово может использоваться в различных близких смыслах. И хотя Шохем под *агентом* понимает программного агента, он базируется на теории искусственного интеллекта и отталкивается от понятия интеллектуальный агент, которое уже существовало ранее. Но, замечая, что это понятие достаточно расплывчато, он ставит перед собой цель преобразовать его для более формального применения в программировании, предлагая специальный АОП фреймворк. В итоге понятие агента, в отличие от простого объекта, наделяется рядом ментальных

конструкций, таких как вера, обязанности и способности. Поэтому в языке программирования будут появляться различные ментальные категории, а семантика программирования будет связана с семантикой ментальных конструкций.^[1]

Связанные понятия

Объект

Программная сущность заданной структуры и конкретизированных механизмов взаимодействия с другими объектами посредством передачи сообщений. Сообщения формируются и отсылаются в ответ на пришедшие сообщения. Формирование сообщений происходит процедурами на основании данных.

Актор

Программная сущность заданной структуры и механизмов взаимодействия.

- Содержит данные и процедуры.
- Обладает инкапсуляцией, отношениями, наследованием и может порождать сообщения.

Агент

Программная сущность для выполнения поставленных задач. Обладает *поведением*, а именно: взаимодействует с внешней сложной и динамично-развивающейся *средой*, способной модифицироваться или быть модифицированной другими агентами в зависимости от конкретных условий. Взаимодействие подразумевает^[4]:

- восприятие динамики среды;
- действия, изменяющие среду;
- рассуждения в целях интерпретации наблюдаемых явлений, решения задач, вывода заключений и определения действий.

В зависимости от степени свободы среды, подразумевающей наличие в ней соответствующего типа агента, среды подразделяются на ^[5]:

Замкнутые

Конечное детерминированное или вероятностное описание всей среды, которое известно агенту априори или путём исследования.

Открытые

Конечное детерминированное или вероятностное описание локальной области среды, в которой находится агент и которое известно ему априори или путём исследования.

Трансформируемые

Динамически развивающиеся среды, развивающей структурой которых является агент.

Связь АОП и ООП

Агентно-ориентированный подход является частным случаем (специализацией) объектно-ориентированного программирования (ООП). В ООП вычислительный процесс понимается достаточно широко как система, собранная из объектов, которые взаимодействуют друг с другом через сообщения. АОП специализирует эти понятия, устанавливая состояние (называемое психическим состоянием) объектов (называемых агентами), состоящих из компонентов таких как верование (убеждения) (включая убеждения о мире, о себе, и о друг друге), способности, и решения, каждое из которых обладает определенным синтаксисом. Кроме того различные ограничения помещены в психическое состояние агента. Вычисление состоит из информирования агентов, выполнения их требований, выполняя их предложения, принимая, отклоняя, конкурируя, и помогая друг другу.^[1]

| | ООП | АОП |
|--|---------------------------------------|--|
| Основная единица | объект | агент |
| Параметры, определяющие состояние основной единицы | неограниченно | убеждения, обязательства, способности, выбор |
| Процесс вычисления | поступление сообщений и методы ответа | поступление сообщений и методы ответа |
| Виды сообщений | неограниченно | информирование, просьба, предложение, обещание, отказ |
| Ограничения на методы | нет | честность, последовательность |

Наличие у агента механизма целеобразования обеспечивает принципиально новый уровень автономии. Агент не обязательно выполняет распоряжения какого-либо другого агента или пользователя, он просто зависит от условий среды, включая цели и намерения других агентов. В отличие от объекта, агент может принять на себя определенные обязательства или, наоборот, отказаться от выполнения некоторой работы, мотивируя это отсутствием компетентности, занятостью другой задачей и т. п. В то же время, агент может выполнять такие действия как порождение, подавление и замена других агентов, активизация функций (как своих, так и у других агентов), активизация сценария деятельности, запоминание текущего состояния других агентов и пр.^[6].

От задачи к реализации

Причины возникновения

Причины возникновения агентно-ориентированного подхода:

1. необходимость преодоления границ операционных сред;
2. устранение разнородности объектных моделей, вызванных тем, что классы и объекты, построенные в различных инструментальных средах, имеют определённые отличия.

Постановка задачи

В целом, система АОП должна включать следующие базовые компоненты:^{[7][8]}

- ограниченный формальный язык с соответствующими синтаксисом и семантикой для описания внутреннего состояния агента, которое определяется несколькими параметрами типа убеждений, желаний, намерений и обязательств;
- язык программирования для спецификации агентов, включающий примитивные команды типа REQUEST и INFORM;
- агентификатор, преобразующий нейтральные компоненты в программируемые агенты.

Основными свойствами, которыми должны обладать агенты, считаются:

- Автономность — способность выполнять действия самостоятельно;
 - Гомогенность/гетерогенность — способность объединять однородные или разнородные функции;
 - Наличие «интеллекта», способности к обучению, коррекция поведения для улучшения собственной эффективности;
1. Активное поведение, постоянный обмен информацией «внутри» агента и между агентом и средой;
 2. Коммуникативность — обмен данными с внешней средой;
 3. Восприятие среды — наличие специальных «средств» восприятия среды функционирования агента;
 4. Мобильность — перемещение агента внутри других программных и физических сред и/или компонентов.

Реализации

Распределённые объектные архитектуры (CORBA, DCOM, Java RMI, WEB-services), обладая многими перечисленными свойствами, имеют следующие недостатки:

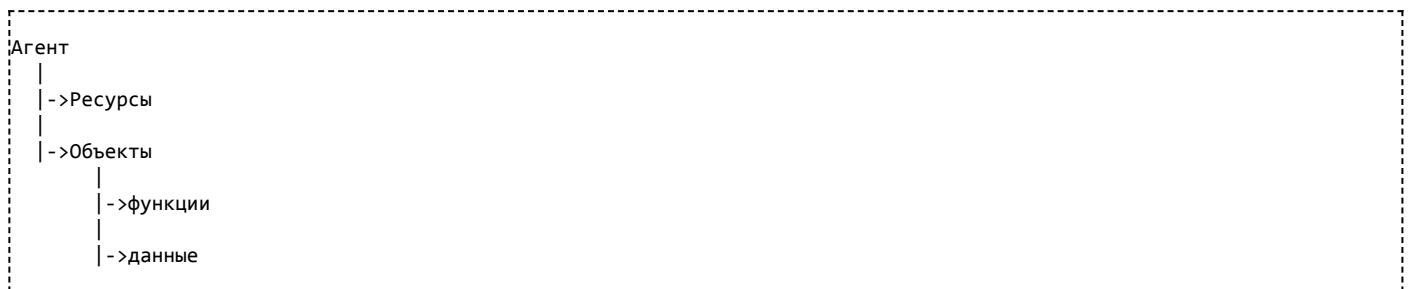
1. необходимость перекомпиляции программных кодов при внесении изменений в объекты и интерфейсы;
2. невозможность динамической адаптации поведения программных объектов в зависимости от состояний и поведения среды;
3. невозможность работы в явной форме с моделями знаний;

Распределённая объектная среда JADE (основана на Java RMI, Java CORBA IDL, Java Serialization и Java Reflection API) придаёт Java агентам следующие свойства:

1. Автономность, способность выполнять действия самостоятельно: агент следует списку поведений, который может быть пополнен асинхронно, то есть и в режиме ожидания, и в режиме «занят — идёт работа».
2. Устойчивость: не имея побочного эффекта после каждого поведения, имеет возможность отката.
3. Активное поведение: заключается в том, что в случае неудачи, выполняются циклические попытки выполнить текущее, а затем перейти к следующему действию поведения. Во время выполнения поведения агент занят — не взаимодействует со средой. После завершения текущего поведения, агент переходит к следующему по списку, а в случае отсутствия списка — ожидает появления такового. Объект же в случае успеха не имеет спуска, а в случае неудачи — бросает задачу.
4. Мобильность: возможность передавать агент в другой контейнер. Данное свойство возникает из отсутствия после выполнения каждого поведения побочного эффекта.

Структура агента

Структурно агента можно изобразить следующим образом:



Структура поведения

Примеры

Типичным примером агентно ориентированного подхода можно назвать специальные программы, имитирующие действия игроков в компьютерных играх. Распространенное название для таких программ — боты. Однако, использование ботов не ограничивается применением в компьютерных играх. Их можно встретить как элемент программы в чатах и форумах, выполняющих роль «собеседника» и поддерживающих «дискуссию». Однако некоторые боты выполняют не только роль имитаторов в процессе общения или игры, они используются для редактирования текстовой информации (в том числе и в Википедии), для наладки сетевых соединений, модерирования ресурсов и т. п.

Рассмотрим предметную область «Торговля книгами», на которой будут проиллюстрированы шаги, необходимые для создания агентно-ориентированных приложений.

Каждый покупающий агент получает название книги, которую он должен приобрести в качестве аргумента командной строки и периодически запрашивает всех известных ему продавцов-агентов, чтобы сделать запрос о покупке. Как только предложение получено, агент-покупатель подтверждает его и отправляет заказ. Если больше чем один агент-продавец предоставляет данную книгу, покупатель выбирает лучшее предложение (самую лучшую цену). Купив требуемую книгу, агент-покупатель завершает работу. Каждый агент-продавец

имеет минимальный интерфейс, с помощью которого пользователь может добавлять новые названия (и их цену) в локальный каталог книг, выставленных на продажу. Агенты-продавцы находятся в состоянии ожидания запросов от агентов-покупателей. Когда они получают запрос на книгу, они проверяют, имеется ли данная книга в их каталоге. Если да, то они отвечают предложением с ценой. Иначе отказывают. Когда они получают заказ на покупку, они обрабатывают его и удаляют запрошенную книгу из своего каталога.

JADE

Для создания агента в JADE достаточно определить метод его инициализации:

```
import jade.core.Agent;

public class BookBuyerAgent extends Agent {
    protected void setup() {
        // Printout a welcome message
        System.out.println("Hello! Buyer-agent "+getAID().getName()+" is ready.");
    }
}
```

Фактическая работа, которую агент должен делать, как правило, осуществляется в рамках «поведения агента». Поведение представляет собой задачу, которую агент может выполнять.

Агент может выполнять одновременно несколько моделей поведения. Однако важно заметить, что расписание нескольких моделей поведения в агенте имеет не упреждающий характер, а кооперативный. Это означает, что когда поведение исполняется по расписанию, его метод `action()` вызывается и работает до тех пор, пока не завершится. Поэтому именно программист определяет, когда агент переключается от исполнения данного поведения к выполнению следующего.

С учетом описанного механизма планирования важно подчеркнуть, что поведение, подобное представленному ниже, препятствует выполнению любого другого поведения, т.к его `action()` метод никогда не завершается.

```
public class OverbearingBehaviour extends Behaviour {
    public void action() {
        while (true) {
            // do something
        }
    }

    public boolean done() {
        return true;
    }
}
```

Примечания

1. *Shoham Yoav*. Agent Oriented Programming (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.5119&rep=rep1&type=pdf>) (англ.) : Technical Report STAN-CS-90-1335. — Computer Science Department, Stanford University, 1990.
2. *Shoham Yoav*. Multiagent systems: Algorithmic, Game-Theoretic, and Logical Foundations (<http://www.masfoundations.org/downloading.html>) = Artificial Intelligence: A Modern Approach. — Cambridge University Press, 2009. — 504 с. — ISBN 9780521899437. (Проверено 1 июня 2010)
3. *Стьюарт Рассел, Питер Норвиг*. Искусственный интеллект: современный подход = Artificial Intelligence: A Modern Approach. — второе. — М.: Вильямс. — 1407 с. — ISBN 5-8459-0887-6.
4. Б.Хэйес-Рот
5. цитируется по Д.А.Поспелов (<http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book8/>) Архивировано (<https://web.archive.org/web/20120114171328/http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book8/>) 14 января 2012 года.
6. *Тарасов В.Б.* Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте (<http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book8/>) // Новости искусственного интеллекта :

- Сб.. — 1998. — № 2. — С. 5-63. Архивировано (<https://web.archive.org/web/20120114171328/http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book8/>) 14 января 2012 года. (Проверено 25 мая 2010)
7. *Shoham Yoav*. Agent-oriented programming (<http://www.cs.unb.ca/~ulieru/Teaching/CS6705/Shoham.pdf>) (англ.) // Artificial Intelligence. — 1993. — No. 60(1). — P. 51-92. Архивировано (<https://web.archive.org/web/20120113022101/http://www.cs.unb.ca/~ulieru/Teaching/CS6705/Shoham.pdf>) 13 января 2012 года. (Проверено 25 мая 2010)
8. *Тарасов В.Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. — М.: Эдиториал УРСС, 2002. — С. 352. — ISBN 5-8360-0330-0.

См. также

- Агентно-ориентированные модели
- Агентное моделирование
- Искусственный интеллект
- Многоагентная система
- Обучение с подкреплением
- Программный агент

Литература

- Зайцев И. М., Федяев О. И.* Агентно-ориентированный подход к моделированию интеллектуальных распределённых систем (<http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book0/>) : Сб. — Донецк: ДонНТУ, 2008. — С. 337-338. Архивировано (<https://web.archive.org/web/20120114171039/http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book0/>) 14 января 2012 года. (Проверено 1 июня 2010)
- Камерон Хьюз, Трейси Хьюз.* Параллельное и распределенное программирование с использованием C++. — М.: Вильямс, 2004. — 672 с. — ISBN 5-8459-0686-5.
- Giovanni Caire.* JADE tutorial for beginners (<http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book1/>) (рус.) = original (<http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>) / перевод И. М. Зайцев. Архивировано (<https://web.archive.org/web/20110307164030/http://www.masters.donntu.edu.ua/2009/fvti/zaytsev/library/book1/>) 7 марта 2011 года. (Проверено 25 мая 2010)
- А.Н. Швецов.* Агентно-ориентированные системы: от формальных моделей к промышленным приложениям (http://window.edu.ru/window_catalog/files/r56179/62333e1-st20.pdf) . Проверено 27 октября 2010. Архивировано (<https://www.webcitation.org/66eQxo6Ug?url=http://window.edu.ru/resource/179/56179/files/62333e1-st20.pdf>) 3 апреля 2012 года.

Источник — https://ru.wikipedia.org/w/index.php?title=Агентно-ориентированный_подход&oldid=94858589

Эта страница в последний раз была отредактирована 2 сентября 2018 в 20:02.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.