6. Агентные технологии

6.1 Понятие программного агента

В соответствии с определением, данным Э. Таненбаумом, *программный агент* – это автономный процесс, способный реагировать на среду исполнения и вызывать изменения в среде исполнения, возможно, в кооперации с пользователями или другими агентами [2].

Агенты могут применяться при решении следующих задач:

- *мобильные вычисления*: миграция агентов может поддерживаться не только между постоянно подсоединенными к сети узлами, но и между мобильными платформами, подключаемыми к постоянной сети на некоторые промежутки времени и возможно по низкоскоростным каналам. Клиент может подсоединяться к постоянной сети на короткий промежуток времени с мобильной платформы, отсылать агента для выполнения задачи и отключаться от сети. Затем клиент подсоединяется к другой точке сети и забирает результаты работы агента. Второй вариант – сервер, на который должен переместиться агент, подсоединяется к сети, а затем отсоединяется. В этом случае агент должен уметь переместиться на такой временно подсоединяемый сервер и вернуться в постоянную сеть;
- *поиск информации:* один человек может быть не в состоянии за короткий срок найти и проанализировать всю необходимую ему информацию. Использование агента позволяет автоматизировать данный процесс. Агент может странствовать по сети и собирать информацию, лучше всего удовлетворяющую поставленной задаче. Поисковые агенты могут содержать сведения о различных информационных источниках (включая тип информации, способ доступа к ней, а также такие характеристики информационного источника, как надежность и точность данных);
- *отбор (обработка) информации*. Из всех данных, приходящих к клиенту, агент может выбирать только те данные, которые могут быть интересны клиенту;
- *мониторинг данных*. Агент может осуществлять извещение пользователя об изменениях в различных источниках данных в реальном времени (например, мобильный агент перемещается на вычислительный узел, на котором расположен источник данных; это эффективнее, чем использовать статического агента, посылающего запросы источнику данных);

– *универсальный доступ к данным*. Агенты могут быть посредниками для работы с различными источниками данных, имеющими механизмы для взаимодействия друг с другом (например, агент создает несколько агентов, каждый из которых работает со своим источником данных).

С. Франклин и А. Грэссер в 1996 году предложили следующее обобщенное определение автономного агента [32]:

Автономный агент – это система, находящаяся внутри окружения и являющаяся его частью, воспринимающая это окружение (его сигналы) и воздействующая на окружение для выполнения собственной программы действий.

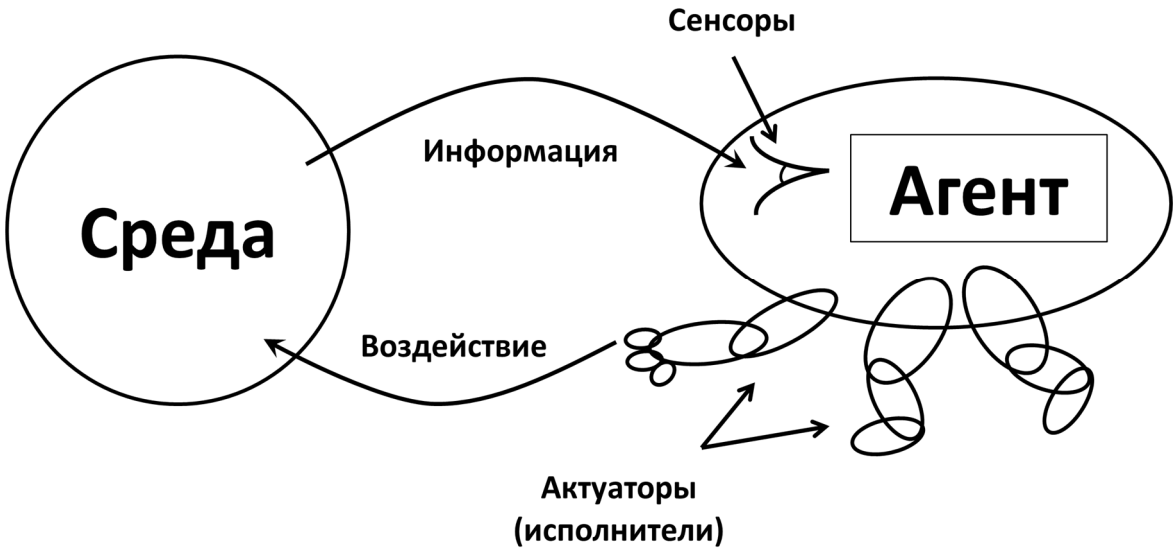

Рис. 17. Автономный агент

Можно выделить следующие основные составляющие автономного агента (рис. 17):

– *Сенсоры*: блоки агента, обеспечивающие получение информации об окружающей среде и других агентах;
– *Актуаторы*: блоки агента, обеспечивающие воздействие на окружающую среду.

При работе простой автономный агент руководствуется стандартным набором правил «Если-то» (рис. 18). Автономный агент должен обладать следующими свойствами:

– реактивность;
– автономность;
– целенаправленность;
– коммуникативность.

Разные авторы не совсем одинаково трактуют перечисленные свойства. Попытаемся объяснить их подробнее [1].

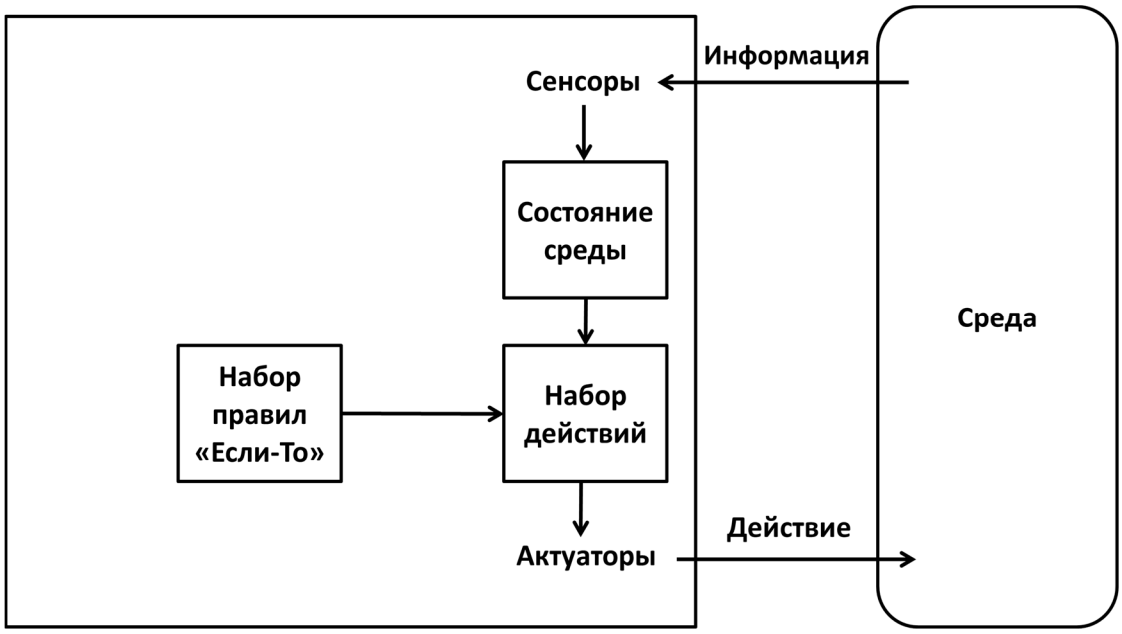

Рис. 18. Структура автономного агента

Свойство реактивности означает, что агент временами отвечает на изменения в окружении. Агент имеет сенсоры, с помощью которых получает информацию от окружения. Сенсоры могут быть самыми различными. Это могут быть микрофоны, воспринимающие акустические сигналы и преобразующие их в электрические, видеокарты захвата изображений, клавиатура компьютера или общая область памяти, в которую окружение помещает данные и из которой программный агент берет данные для вычислений. Не все изменения окружения становятся известными (доступными) сенсорам агента. Это вполне естественно. Ведь и человек не воспринимает звуки сверхвысокой частоты, радиоволны и т.д. Таким образом, окружение не является полностью наблюдаемым для агента. Аналогично, агент воздействует на окружение, путем разнообразных исполнительных механизмов, включая общую память. Разумеется, степень воздействия, как и степень восприятия, является ограниченной. Агент может перевести окружение из некоторого состояния в некоторое другое, но не из любого в любое.

Свойство автономности означает, что агент является самоуправляющимся, сам контролирует свои действия. Программный агент, находящийся на некотором сервере, обладает возможностью «самозапуска». Он не требует от пользователя каких-либо специальных действий по обеспечению его старта (подобно тому, как мы «кликаем» два раза по иконке некоторого файла).

Свойство целенаправленности означает, что у агента имеется определенная цель и его поведение (воздействие на окружение) подчинено этой цели, а не является простым откликом на сигналы из окружения. Иначе говоря, агент является управляющей системой, а не управляемым объектом.

Свойство коммуникативности означает, что агент общается с другими агентами (включая людей), используя для этого некоторый язык. Это не обязательно единый язык для всех агентов. Достаточно, чтобы у пары общающихся агентов был общий язык. Язык может быть сложным как, например, естественный язык. Но может быть и примитивным: обмен числами или короткими словами. Если многословные фразы сложного языка несут всю информацию, как правило, в себе, то слова простого языка предполагают «умолчание»: обе стороны диалога «знают», о чем идет речь (как в известном анекдоте о занумерованных анекдотах).

В отдельную категорию интеллектуальных агентов выделяют автономные агенты, обладающие свойством обучаемости. *Свойство обучаемости* означает, что агент может корректировать свое поведение, основываясь на предыдущем опыте. Это не просто накопление в памяти параметров окружения, т.е. использование исторических данных, но сопоставление истории собственных действий с историей их влияния на окружение, и изменение в связи с этим своей программы действий.

Одна из главнейших особенностей агента – это интеллектуальность. *Интеллектуальный агент* владеет определенными знаниями о себе и об окружающей среде, и на основе этих знаний он способен определять свое поведение (рис. 19). Интеллектуальные агенты являются основной сферой интересов агентной технологии. Важна также среда существования агента: это может быть как реальный мир, так и виртуальный, что становится важным в связи с широким распространением сети Интернет. От агентов требуется способность к обучению и даже самообучению [12]. Способность планировать свои действия делит агентов на *регулирующие* и *планирующие*. Если умение планировать не предусмотрено (регулирующий тип), то агент будет постоянно переоценивать ситуацию и возобновлять свое воздействие на окружающую среду. Планирующий агент может запланировать несколько действий на разные промежутки времени. При этом агент может моделировать развитие ситуации, что дает возможность более адекватно реагировать на текущие ситуации. При этом агент должен принимать во внимание не только свои действия и реакцию на них, но и сохранять модели объектов и агентов окружающей среды для прогнозирования их возможных действий и реакций.

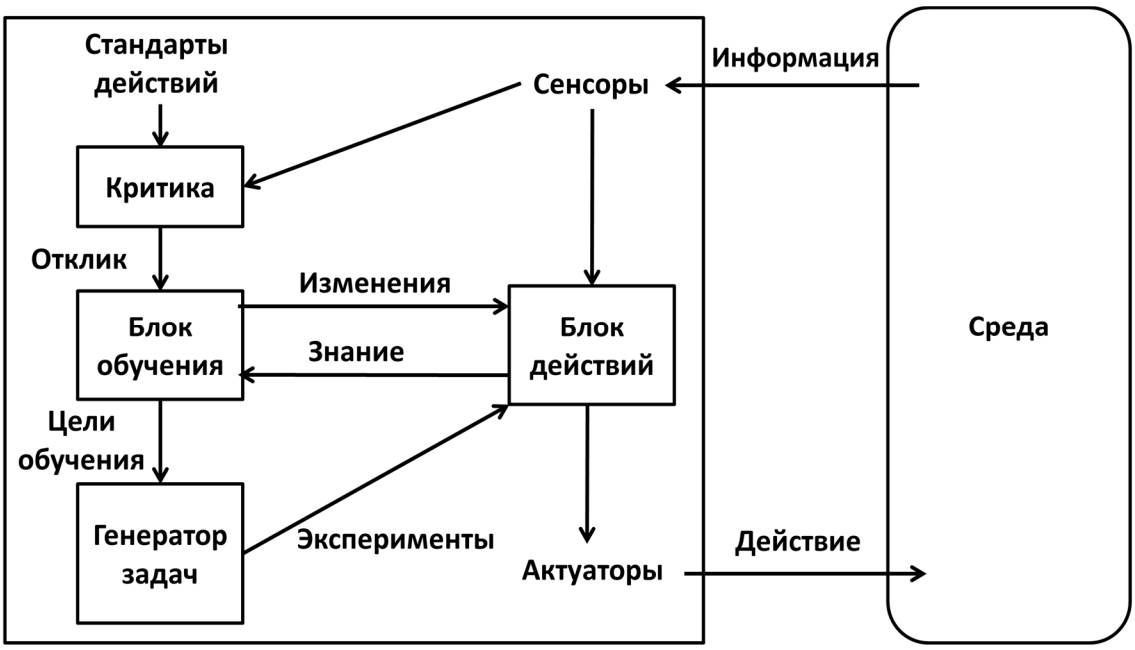

Рис. 19. Структура интеллектуального агента

6.2 Мультиагентные системы

Мультиагентная система (МАС, англ. Multi-agent system) — это система, образованная несколькими взаимодействующими агентами.

Мультиагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы.

В мультиагентной системе необязательно все агенты взаимодействуют (общаются) между собой. В крайнем случае, общения нет вообще. Такие системы назовем дискретными мультиагентными системами. Второй крайний случай – каждый агент общается с каждым. Такую систему можно назвать полносвязной мультиагентной системой [1].

Мультиагентная система, действующая как единый агент, должна характеризоваться и некоторой общей для всех субагентов целью и координацией действий по достижению этой цели. Поскольку встречаются и другие ситуации, когда агенты не связаны столь тесно, то такие системы можно назвать обществами агентов. Отсутствие единой цели, однако, не отрицает возможного группового поведения агентов. Но оно является, скорее, эпизодическим, чем систематическим.

Важным отличием мультиагентной системы от программы или одного агента является то, что входящие в систему программные агенты (по крайней мере, некоторые) не были спроектированы специально для этой системы. Может быть, это – повторно используемые агенты, или агенты, разработанные для

решения более универсальных задач. В этих случаях агенты имеют собственные цели, не совпадающие полностью с целями системы (организации), но совместимые с ними. Тем не менее, они могут быть полезны друг другу для решения стоящих перед ними задач и, поэтому, очень важным для них с этой точки зрения является свойство коммуникативности [1].

6.2.1 Агентные платформы

Агентная платформа – это промежуточное программное обеспечение, поддерживающее создание, интерпретацию, запуск, перемещение и уничтожение агентов. Как «воздух» для человека, агентная платформа обеспечивает агентам среду для жизни и работы.

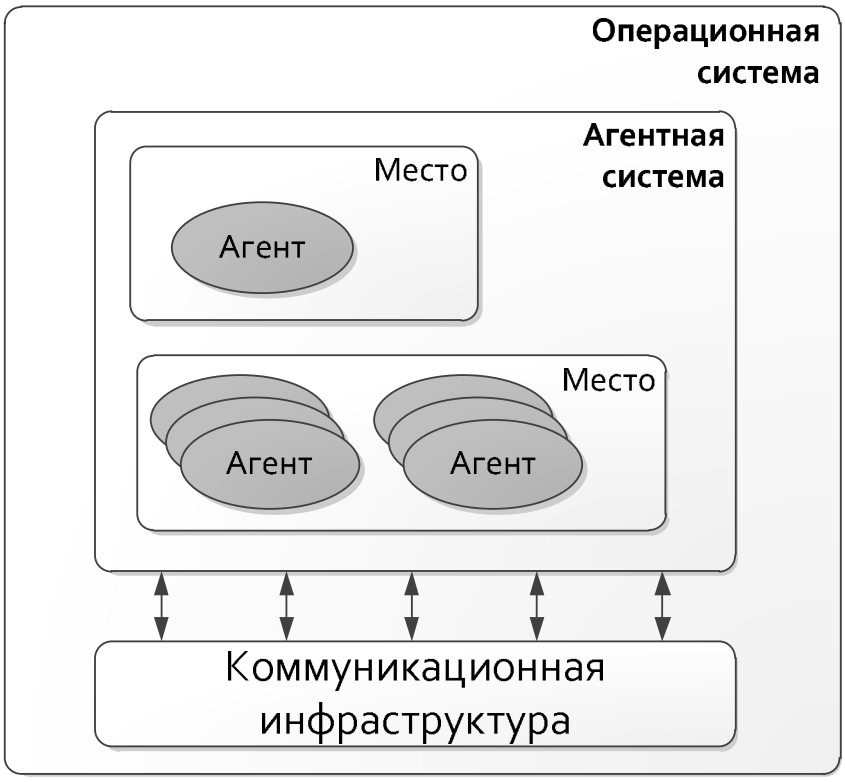

Рис. 20. Агентная система

Агентная система – это приложение, однозначно идентифицируемое именем и адресом, которое обеспечивает жизненный цикл агентов на конкретном узле РВС. На одной машине могут размещаться несколько агентных систем. Тип агентной системы определяется агентной платформой и описывает совокупность параметров агента.

Все общение между агентными системами осуществляется через коммуникационную инфраструктуру.

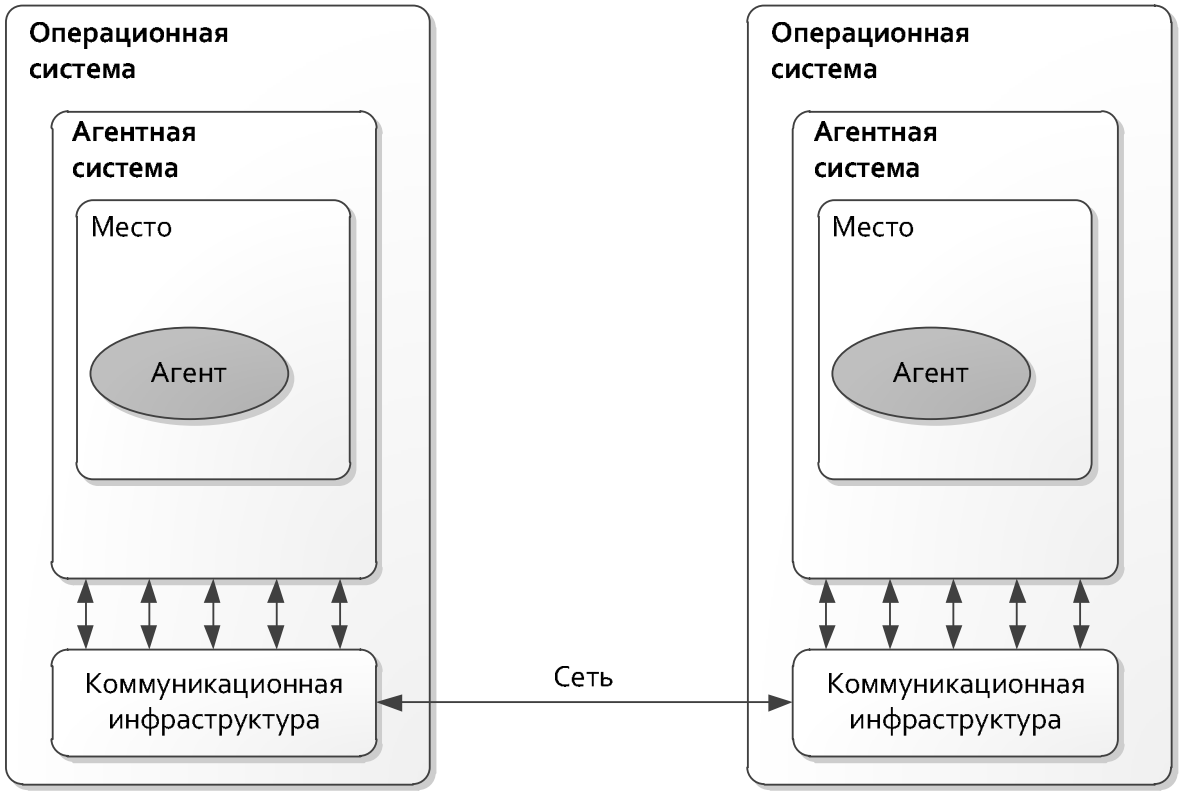

Рис. 21. Связи между агентными системами

Коммуникационная инфраструктура обеспечивает транспортные службы связи (например, RPC), службу имен и службу безопасности для агентных систем.

На сегодняшний день, можно выделить следующие наиболее распространенные агентные платформы.

1. JADE (Java Agent DEvelopment Framework) [39] – Инфраструктура для разработки агентов на языке Java. JADE включает в себя:
   – Динамическую среду, где JADE агенты могут «жить»;
   – Библиотеку классов для разработки агентов;
   – Набор графических инструментов, позволяющий управлять и следить за активностью запущенных агентов.
2. Cougaar (Cognitive Agent Architecture) [12] – это основанная на Java архитектура для построения высокомасштабируемых распределенных агентных приложений.

6.3 Безопасность в системах мобильных агентов

Агентные распределенные вычислительные системы нашли определенную нишу в сфере РВС, но их повсеместному распространению препятствует сам принцип их работы: вряд ли здравомыслящий администратор вычислительной сети позволит странствовать по компьютерам пользователей *автономным ин-*

теллектуальным изменяющимся самообучающимся программным системам, которые могут мигрировать и выполняться на любом вычислительном узле. В связи с этим возникает целый ряд серьезных проблем, связанных с безопасностью агентных платформ.

С одной стороны, агенты могут нести личную информацию для обеспечения собственной работы. Например, агент в системе электронной коммерции может содержать номер кредитной карточки и паспортные данные пользователя для того, чтобы от его имени совершать сделки. Соответственно, необходимо, чтобы среда обеспечивала безопасную для агента среду исполнения.

С другой стороны, сторонний агент может попытаться атаковать базовую среду и извлечь данные или заполучить иные ресурсы. В этом случае агентная платформа должна быть достаточно хорошо защищена и иметь возможность противостоять таким атакам.

Выделяют следующее возможные проблемы безопасности при работе агентных платформ:

- *Агент атакует Хост*: Агент может украсть или модифицировать данные хоста
- *Хост атакует Агента*: Хост может украсть или модифицировать данные Агента, изменить его состояние или код
- *Злонамеренный агент атакует другого агента*;
- *Атака другими элементами*.

Вариант атаки «Агент атакует Хост» является стандартной атакой, в которой код, полученный из ненадежного источника, пытается получить полный доступ к системе или же помешать нормальному выполнению задач, повысив свои полномочия на исполнение. В этом случае от атаки помогают традиционные средства защиты, как то:

- контроль уровня доступа;
- песочницы;
- аутентификация;
- криптография.

Также, существуют специфические для агентных платформ методы обеспечения безопасности, например анализ истории движения агента. В этом случае платформа может узнать об истории передвижения агента из лога и сделать вывод о его качестве на основе информации о том, на каких платформах он побывал до этого.

При варианте атаки «Хост атакует агента» традиционные средства обеспечения не работают, так как хост должен иметь полную информацию о коде агента для его исполнения. Соответственно, традиционные методы обеспечения безопасности оказываются бессильны. В этом случае могут быть применены следующие средства:

- *Мобильная криптография*: функции и данные агента шифруются таким образом, что хост не может разобрать, каким образом функции исполняются и извлечь код. Недостатком данного метода является необходимость поиска схемы шифрования для произвольных функций, а также необходимость переноса ключа кадрирования.
- *Безопасное перемещение*: миграция только на определенные (доверенные) хосты.
- *Использование фиктивных данных*: в базе данных системы, анализирующей работу агентов, хранится набор фиктивных данных, которые не изменяются в ходе нормальной работы агента.
- *Использование доверенного аппаратного обеспечения*: это могут быть смарт-карты, интегрированные микросхемы, и т.п.