

Р.М. Алгулиев, М.Ш. Гаджирагимова

НЕКОТОРЫЕ АСПЕКТЫ ОРГАНИЗАЦИИ И РЕАЛИЗАЦИИ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ПОИСКА ИНФОРМАЦИИ В РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННОЙ СРЕДЕ

*Институт информационных технологий НАНА, Баку, Азербайджан
secretary@iit.ab.az, depart1@iit.ab.az*

Одной из наиболее важных проблем современных информационных систем является задача эффективного доступа пользователя к распределенным информационным ресурсам. Вопрос, возникающий перед разработчиками информационных систем различного назначения, применяемых в разных сферах, как обнаружить данные, разбросанные по электронным объектам, решается путем применения интеллектуальных агентов [1]. На данный момент имеется множество мнений и ряд определений о том, что такое агент. С точки зрения объектно-ориентированного подхода (ООП) агент-объект представляет собой комплекс из набора данных и процедур (функций) с интерфейсом, способным получать и посылать информацию.

В зависимости от среды агенты обладают разными свойствами, характеризующими понятие агента. Обычно агент обладает следующими свойствами, которые приведены в таблице 1 [2-8].

Таб.1

Свойство	Описание
Автономность	Агент функционирует без вмешательства со стороны человека
Способность общения	Возможность взаимодействия и коммуникации с другими агентами
Реактивность	Восприятие среды и проявление соответствующей реакции на ее изменения
Активность	Умение действовать рациональным образом для выполнения поставленных задач
Базовые знания	Знания агента о себе, окружающей среде, включая других агентов
Мобильность	Способность изменять местоположение в окружающей среде

Если взять общепринятую формулу "класс = данные + методы (функции для работы с ними)", то можно сказать, что агент - это объект, обладающий функциями для приема информации из внешнего мира, воздействия на внешний мир и принятия решения о роде воздействия.

Понятие интеллектуального агента является развитием методов ООП. Упрощая можно сказать, что "интеллектуальный агент = данные + методы + знания", причем методы здесь - это не только функции работы с данными, но и функции работы со знаниями и возможные методы воздействия на окружающую среду агентом [3].

Важным аспектом интеллектуальных агентов является возможность реализации полнофункциональной интеллектуальной системы. Так, например, виртуальный системный администратор, реализованный по агентной технологии, может реально следить за безопасностью работы сети (сохранностью информации, сбоях в сети и прочее). При этом все его действия не выйдут за рамки программных воздействий.

Разработка многоагентных систем идет по двум направлениям: системы на основе распределенного искусственного интеллекта и системы на основе сценариев [4].

В многоагентных системах (МАС), на основе распределенного искусственного интеллекта каждый агент рассматривается как нераспределенная система, основанная на знаниях с добавлением компонентов, обеспечивающих безопасность, мобильность,

качество обслуживания, взаимодействие с другими агентами, сетевыми ресурсами и пользователями.

Многоагентные системы на основе сценариев ориентированы на использование в больших компьютерных сетях. Агенты данного класса систем разрабатываются с помощью интерпретируемых языков, а конкретнее Java, PHP и др.). Использование интерпретируемых языков облегчает поддержку гетерогенных систем. Эти языки ориентированы на реализацию асинхронного процесса и удаленное исполнение приложений. Нужно отметить, что разрабатываемые МАС на основе сценариев рассматриваются как средства для реализации мобильной телекоммуникации через компьютерные сети общего пользования.

Интеллектуальные агенты представляют собой программы (приложение и интерфейсы пользователя), предназначенные для автоматического сбора, фильтрации и организации информации в распределенных системах. В связи с тем, что МАС обладают рядом преимуществ по сравнению с другими способами организации распределенных вычислений (уменьшается время, стоимость передачи данных и нагрузка на сеть, вычисления выполняются автономно и асинхронно и могут адаптироваться к условиям своего выполнения и др.), для решения задач организации поиска данных в распределенных информационных системах наиболее эффективным и целесообразным является использование мультиагентного подхода, в рамках которого система строится как совокупность агентов (агенты пользователя, агенты менеджера и агенты исполнителя). Выбор мультиагентной технологии в качестве базовой при проектировании распределенных систем доступа позволяет легко сочетать в единой системе как универсальные протоколы так и любые другие средства работы с конкретными типами баз данных. Ещё на этапе проектирования в такую систему закладывается гибкость, горизонтальная и вертикальная расширяемость, существенно упрощается решение задач распределения нагрузки между серверами.

Программно реализованные агенты, в том числе и интеллектуальные, относятся к классу программного обеспечения, которое способно действовать самостоятельно от лица пользователя. Созданию программных агентов предшествовал опыт разработки так называемых открытых систем, результатом внедрения которых в практику явилось создание архитектуры «клиент-сервер». В настоящее время наибольшее распространение получили две модели такого взаимодействия: «толстый клиент - тонкий сервер» и «тонкий клиент - толстый сервер». В первой модели серверная часть реализует доступ к ресурсам, а приложения находятся на компьютерах клиентов. Во второй модели клиентское приложение обеспечивает только реализацию интерфейса, а сервер объединяет все остальные части программного обеспечения. При создании МАС используются обе модели. При этом применяется либо статический подход, либо динамический подход, обеспечивающий также передачу программного кода [5].

Динамический подход опирается на мобильных агентов, которые в отличие от статических могут перемещаться по сети. Они могут покинуть клиентский компьютер и перемещаться на удаленный сервер для выполнения своих действий, после чего могут возвращаться обратно.

Для реализации МАС, основанных как на статических, так и на динамических распределенных приложениях, наиболее перспективными являются следующие технологии: DCOM (Microsoft Distributed Component Object Model), Java RMI (Java Remote Method Invocation) и CORBA (Common Object Request Broker Architecture) [6].

Появление и широкое развитие сетевых технологий привело к возникновению необходимости создания согласованных средств обеспечения условий эффективного поиска в разнородных видах баз данных, предоставляемых их пользователям для удаленного доступа. При организации систем доступа конечного пользователя к информационным ресурсам, одним из ключевых технических характеристик системы является время отклика на запрос пользователя. Именно эта характеристика и является наиболее узким местом всех распределенных систем. И хотя в последнее время

пропускные способности каналов связи стремительно растут, нагрузки на них также повышаются и поэтому, коммуникация через Сокеты является вполне логичным выбором, для организации распределённой агентной системы, ориентированной на работу конечного пользователя в онлайн-режиме (минимально возможное время отклика системы).

Традиционная технология процесса организации поиска информации в базах данных распределенной информационной системы предполагает персональное обращение пользователя по сети Интернет к каждому серверу с запросом, формирование из полученных ответов сводного результата и его последующая обработка. Выполнение рутинных операций отнимает у специалистов достаточно много рабочего времени. В связи с этим становится актуальной задача разработки MAC для автоматизации процессов исполнения запросов в распределенной информационной системе, которая взяла бы на себя большую часть рутинных операций по организации поиска информации в базах данных системы серверов. Поиска информации в базах данных распределенной среды MAC имеет архитектуру, которая представлена на рис. 1.



Рис. 1. Архитектура мультиагентной системы

Под MAC будем понимать многокомпонентную систему, состоящую из совокупности взаимосвязанных программных модулей [7]. Основные компоненты (агенты) входящие в MAC, относятся к одному из следующих основных классов: интерфейсные агенты; агенты-менеджеры; агенты-исполнители. Их можно разделить на две части: клиентскую и серверную.

Предполагается, что агенты расположены на разных серверах системы с интегрированными базами данных, расположенных в удаленных географических местах и взаимодействуют друг с другом при выполнении запроса, поступившего от пользователя. В качестве агентов в данном случае выступают независимые исполняемые модули – приложения, написанные на одном из языков сценариев.

В серверную часть MAC входят агенты: ControlAgent; SQLAgent, а в клиентскую часть – InterfaceAgent (Web-интерфейс).

Основными функциями небольшого исполняемого модуля InterfaceAgent являются: регистрация пользователя на сервере; прием от пользователя запроса на поиск данных во всех базах данных распределенной информационной системы и оказание ему помощи при формировании sql-запроса; установление связи с ControlAgent и передача ему запроса.

Основными функциями ControlAgent являются: прием от InterfaceAgent запросов на поиск данных в распределенной информационной системе; внесение запроса в очередь

запросов, связь с другими серверами системы и передача им запросов; активизация SQLAgent для поиска требуемой информации в локальной базе данных.

Основными функциями SQLAgent являются: отбор данных в базе данных сервера по запросу, полученному от ControlAgent; перекодировка информации, получаемой из БД; запись полученных данных в результирующий поток-файл; формирование общего ответа на запрос для отправки ответа пользователю.

Общая схема работы пользователя с МАС выглядит следующим образом. Пользователь обращается по сети Интернет с использованием стандартных программ-браузеров к одному из серверов распределенной информационной системы и с помощью интерфейсной формы МАС формирует запрос на поиск информации в базах данных этой системы. МАС автоматически осуществляет поиск нужной ему информации на всех серверах системы, готовит общий ответ на запрос и уведомляет пользователя о полученном результате.

В качестве агентов в таких системах выступают независимые загрузочные модули ('exe') - приложения. Программной средой, необходимой для удаленной коммуникации между отдельными частями программы, служит сеть Internet. В такой среде общение между агентами осуществляется через Сокеты по протоколу TCP/IP, а работа с базой данных осуществляется по протоколу ODBC.

Итак, агентная технология, является перспективной областью исследования. Решение таких задач поможет автоматизировать и интеллектуализировать обработку информации и, как следствие, ускорить и улучшить эту обработку. Предложенный в работе подход можно использовать в решении поставленной задачи в тех случаях, когда приоритетным является экономия рабочего времени специалиста за счет передачи части рутинных операций по поиску данных на МАС.

Литература

1. Интеллектуальные информационные системы, Финансы и статистика, 2004, с. 307-332
2. Пономаренко Л.А., Филатов В.А., Е.Е. Цыбульник Агентные технологии в задачах поиска информации и принятия решений // Управляющие системы и машины, №1, 2003, с. 36-41.
3. Евгениев Г.Б. Принципы построения мультиагентных систем автоматизации проектирования и управления // Информационные технологии, № 4, 2000, с.2-7.
4. Чекинов Г.П., Чекинов С.Г. Применение технологии многоагентных систем для интеллектуальной поддержки принятия решения (ИППР) // Системотехника, №1, 2003.
<http://systech.miem.edu.ru/2003/n1/Chekinov.htm>
5. Распределенные системы. Принципы и парадигмы, Э.Таненбаум, М. ван Стеен, Питер, 2003, с.197.
6. Смирнов А.В., Шереметов Л.Б., Организация взаимодействия агентов в многокомпонентных САПР. //Автоматизация проектирования, №2, 1999, с. 36.
7. Федотов В.Б. Построение распределенной системы доступа к информационным ресурсам на основе многоагентной архитектуры / VII Международная конференция по электронным публикациям "EL-Pub2002", 23-27 сентября 2002 г., г. Новосибирск.
8. Зайцева С.В., Перевозчикова О.Л., Модел диалога по расписанию действий агента. // Управляющие системы и машины, №3, 2003, с. 56-63.