

# АГЕНТНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ СЕТЕВЫХ ПРИЛОЖЕНИЙ ПОИСКА ИНФОРМАЦИИ

Е. В. Болдырев, Г. С. Кирякова, А. В. Шилкин

*Красноярский государственный технический университет, Россия*

e-mail: igm@hotmail.ru

Classic client-server architecture does not suit well for the design of the distributed network applications. In this article we describe a new approach to creation of such applications for instance in a distributed data search system in a set of databases of different types.

Классическая клиент-серверная архитектура работает хорошо, если есть только один сервер (одна база данных), а клиентские приложения совершают четко определенные простые действия с данными. Однако по мере возрастания сложности системы, когда количество серверов и источников данных возрастает, количество и алгоритмическая сложность приложений увеличиваются, а клиент-серверная система очень быстро исчерпывает возможности по развитию. С усложнением клиентского приложения увеличивается и его размер, что, в свою очередь, приводит к ужесточению требований к клиентскому компьютеру, его вычислительной мощности, размеру оперативной и долговременной памяти. А поскольку количество пользователей в системе также имеет тенденцию к росту, мы неминуемо должны увеличивать мощность серверной аппаратуры. Экстенсивный путь развития системы рано или поздно приводит к серьезным проблемам в ее поддержке, расширяемости, безопасности. Кроме того, процесс создания сетевых приложений, используя клиент-серверный подход, зачастую требует знания множества языковых и инструментальных средств разработки и умения решать вопросы конфигурирования этих средств (например, Apache — CGI — Perl — HTML — Javascript). Очевидно, что необходим новый, интенсивный, путь развития программного обеспечения. В работе предлагается использовать для этого технологию программных агентов.

Области применения программных агентов могут быть очень обширны, на рис. 1 приведена классификация агентов в соответствии с классами решаемых задач [1].

Использование агентных технологий упрощает:

— процессы размещения (deployment) программного обеспечения в условиях сети, автоматизируя процессы перемещения программного кода, его установки и конфигурирования;

— процессы обновления программного кода, заменяя неактивные в данный момент компоненты новыми версиями, сохраняя при этом ассоциированные с компонентами данные, совершенно прозрачно для работы всей системы в целом;



Рис. 1. Классы решаемых задач.

— удаленный доступ и управление в условиях сети, в том числе с использованием мобильных устройств, таких как сотовые телефоны и PDA;

— разработку программного обеспечения за счет возможности повторного использования уже существующего программного обеспечения (ПО) и параллельной разработки ПО (новый агент может использовать в своей работе множество уже существующих агентов, посылая к ним свои запросы либо наследуя их свойства и функции, кроме того, за счет упрощения процесса обновления исключается останов системы во время обновления).

Кроме того, обеспечиваются:

— качественно новый способ работы с данными как со знаниями с учетом контекста существования информации за счет использования онтологий;

— уменьшение сетевого трафика и общей избыточности информации за счет возможности непосредственного перемещения кода агента к обрабатываемым данным вместо перемещения самих данных, как в случае использования клиент-серверного подхода;

— уменьшение влияние качества и латентности каналов связи за счет возможности асинхронной обработки данных по следующим схемам: 1) отправка запроса агенту на обработку — отключение от сети — подключение к сети — получение результатов обработки; 2) отправка запроса агенту — выполнение другой работы — получение уведомления о завершении обработки и ее результаты;

— балансировка нагрузки между несколькими вычислительными ресурсами и возможность параллельной обработки данных за счет связи между агентами и перемещения кода агента на другие агентные платформы;

— сокращение времени и затрат на администрирование за счет возможности коммуникаций между агентами и перемещения их кода;

— ведение персонализированной обработки данных с ориентацией на предпочтения пользователя.

Сравнительный анализ клиент-серверного и агентного подходов по основным признакам приведен в таблице.

Оценочная классификация клиент-серверного и агентного подходов по пятибалльной шкале

Признак	Название подхода	
	Клиент-серверный	Агентный
Прозрачный доступ к данным	3	5
Удобство размещения (deployment)	2	5
Удобство обновления компонентов	2	5
Удобство администрирования системы	2	4
Повторное использование компонентов	3	5
Поддержка уже существующего ПО	3	5
Снижение избыточности информации	2	5
Объем сетевого трафика	3	4
Работа в режиме реального времени	5	3
Простота разработки	4	3
Уровень безопасности системы в целом	3	5
<b>Средняя оценка</b>	<b>2.9</b>	<b>4.5</b>

Таблица составлена по результатам опроса экспертов в области разработки программного обеспечения и профессиональных программистов, занимающихся разработкой информационной системы поддержки бизнес-процессов промышленного предприятия PROVIS (Process Visualization System). Анализ результатов показывает, что агентный подход превосходит клиент-серверный по большинству признаков, проигрывая лишь в работе в режиме реального времени и простоте разработки. С использованием агентного подхода разработана система поиска информации в распределенных по сети базах данных. В качестве средства разработки были использованы язык Java и платформа JADE [2].

Платформа разработки Java-агентов JADE (Java Agent Development Framework) используется для создания мультиагентных систем и приложений в соответствии со стандартами FIPA [1, 3] для интеллектуальных агентов. Она включает в себя среду выполнения агентов. Агенты регистрируются и работают под управлением этой среды, в ней предусмотрены механизмы создания, уничтожения, взаимодействия, поиска агентов (рис. 2).

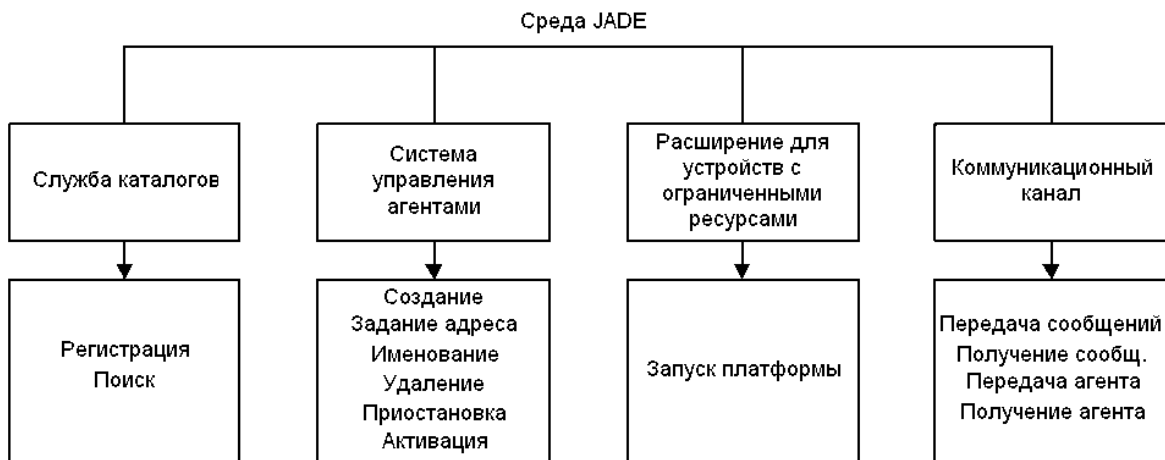


Рис. 2. Основные компоненты среды JADE.

Среда включает систему управления агентами (Agent Management System), которая предоставляет средства именованного и управления жизненным циклом агента, и службу каталогов (Directory Facilitator), включающую регистрационный каталог (каталог Желтых страниц). Используя этот каталог, агент может связаться с другими агентами, необходимыми ему для достижения цели.

Кроме того, JADE содержит библиотеку классов, используемых для разработки агентов и агентной системы, и набор графических утилит для администрирования и наблюдения за жизнедеятельностью агентов, функционирующих в системе. Важной особенностью JADE является поддержка исполнения агентов на устройствах с ограниченными возможностями (CLDC, Connected Limited Device Configuration), например мобильных телефонах, карманных компьютерах (PocketPC, Palm и др.), за счет использования расширения LEAP (Lightweight Extensible Agent Platform) (рис. 3).

Архитектура разработанной системы поиска информации в системе распределенных в сети базах данных приведена на рис. 4 [4].

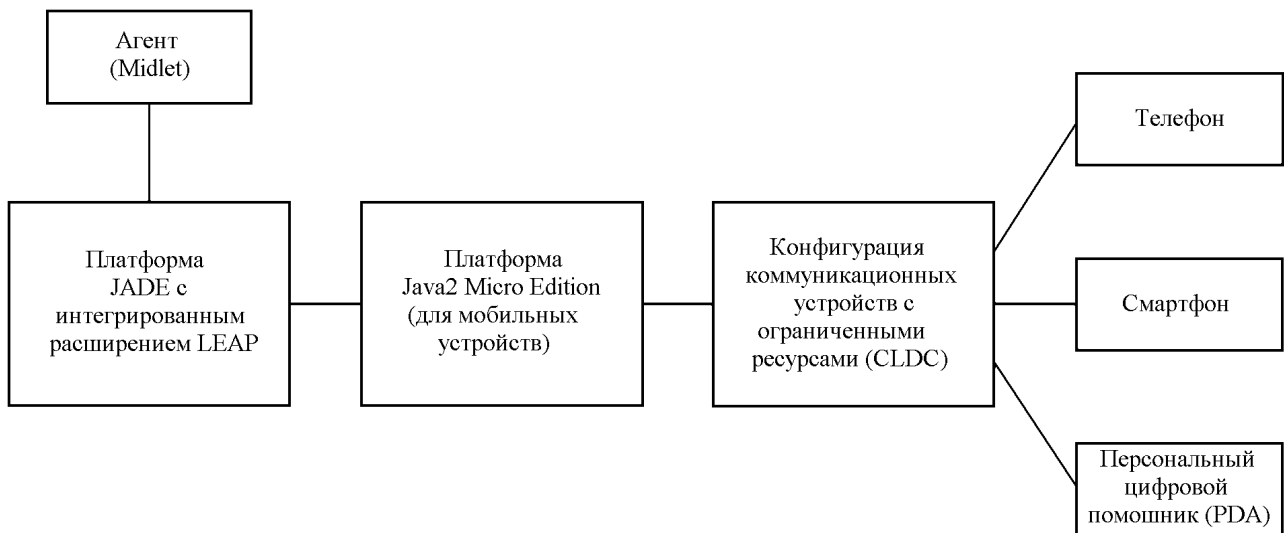


Рис. 3. Схема работы агента на устройстве с ограниченными возможностями.

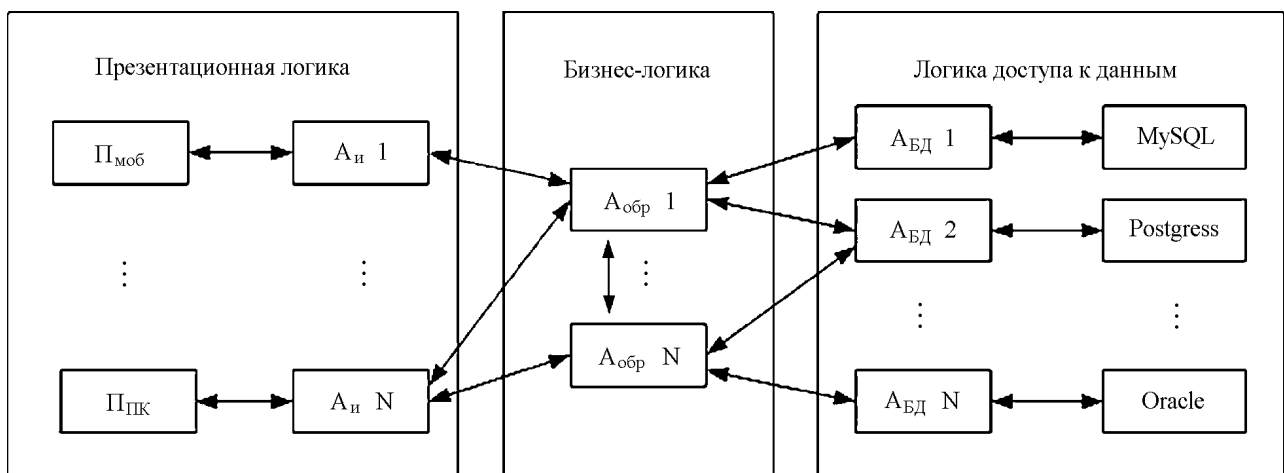


Рис. 4. Схема работы агента на устройстве с ограниченными возможностями.

В систему включены три типа агентов: интерфейсные агенты ( $A_{и}$ ), реализующие презентационную логику и непосредственно взаимодействующие с пользователями сотовых телефонов ( $P_{моб}$ ) и персональных компьютеров ( $P_{ПК}$ ), агенты-обработчики ( $A_{обр}$ ), реализующие бизнес-логику системы, и агенты доступа к базам данных ( $A_{БД}$ ), предоставляющие унифицированный интерфейс работы с источниками данных и реализующие логику доступа к данным. Такое разделение агентов системы позволило сделать ее более гибкой и расширяемой — реализация новых возможностей по обработке и доступу к новым источникам данных не требует внесения существенных изменений в систему или в ее компоненты. Агенты системы имеют возможность работать параллельно на одном или множестве вычислительных устройств (ПК, мобильный телефон и др.). Для минимизации времени ожидания результатов поиска и обработки агенты доступа к базам данных функционируют непосредственно на машинах, на которых работают СУБД. Интерфейсные агенты работают на устройствах пользователей, и ввиду ограниченности возможностей этих устройств (малого объема оперативной памяти, слабого процессора, низкой скорости передачи информации) в них нет функций по обработке данных, их задачей является реализация пользовательского интерфейса. Агенты-обработчики реализуют основную функциональность системы, осуществляя прием заданий от интерфейсных агентов, формирование заданий агентам доступа к базам данных, обработку полученных от них результатов, исключение повторяющихся данных, возврат результатов обработки интерфейсным агентам.

Предложенная система была протестирована в ООО “Мобилон” (оператор ip-телефонии) при работе с корпоративными базами данных компании, функционирующими на нескольких серверах в сети Интернет. Результаты показывают, что время ожидания результатов поиска снижается за счет псевдопараллельности операций поиска, значительно сокращается время разработки и внедрения системы, реализованной с использованием агентного подхода по сравнению с клиент-серверным.

## Список литературы

- [1] ТАНИЕНБАУМ Э., ВАН СТЕЕН М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003.
- [2] BELLIFEMINE F., CAIRE G., POGGI A., RIMASSA G. 2003, JADE. A White Paper. <http://jade.cselt.it/papers/WhitePaperJADEEXP.pdf>
- [3] FIPA: the foundation for intelligent agents. 2000. <http://www.fipa.org/specifications>.
- [4] БОЛДЫРЕВ Е.В. Архитектуры мультиагентных систем. Деп. ВИНТИ. Рег. № 236–В2005.

*Поступила в редакцию 2 ноября 2005 г.*