

Сергей Толкачев

*Нейронное
программирование
диалоговых
систем*



Сергей Толкачев

**Нейронное программирование
диалоговых систем**

«Издательские решения»

Толкачев С.

Нейронное программирование диалоговых систем /
С. Толкачев — «Издательские решения»,

ISBN 978-5-44-963928-8

В книге изложены методы построения динамических диалоговых интернет-приложений на основе нейронных лингвистических моделей. Описаны способы конструирования лингвистических нейронов и их применения для решения различных прикладных задач. В качестве приложения в книгу включены программы, позволяющие читателю самостоятельно экспериментировать с простейшими нейронными моделями.

ISBN 978-5-44-963928-8

© Толкачев С.
© Издательские решения

Содержание

Введение	6
Энергия, информация и знания	11
Конец ознакомительного фрагмента.	15

Нейронное программирование диалоговых систем

Сергей Толкачев

© Сергей Толкачев, 2019

ISBN 978-5-4496-3928-8

Создано в интеллектуальной издательской системе Ridero

Введение

...бывали случаи, когда из положительно дикого брожения умов выходила со временем истина.

И. М. Сеченов. «Рефлексы головного мозга».

Способность выслушать и понять собеседника является одним из наиболее ценных достоинств человеческого общения. Задавая вопросы или просто обмениваясь фразами в процессе разговора, мы стремимся получать ответы, адекватные нашим внутренним потребностям. Нас в равной степени не удовлетворяют как сообщения, не приносящие информации, так и ответы, в которых ее количество значительно превосходит внутренние ограничения, установленные нами для конкретного разговора. Смысловое содержание ответа или адекватная реакция собеседника является одним из основных критериев, определяющим качество разговора и влияющим на изменение внутреннего состояния человека в процессе коммуникативного общения.

Если предположить, что у субъекта разговора существует некоторая целевая функция, которая определяет ожидаемое количество информации (*желаемый результат*), а количество информации, в поступающих к этому субъекту сообщениях, изобразить в виде некоторой траектории, то в зависимости от характера разговора эти траектории могут принимать самую разнообразную форму (рис. 1). Иногда необходимый результат может быть достигнут оптимальным образом (*a*). В иных случаях, задав тот или иной вопрос, вместо полезного ответа может быть получено большое количество избыточной информации, которая может увести далеко от первоначальной цели (*b*) и (*c*).

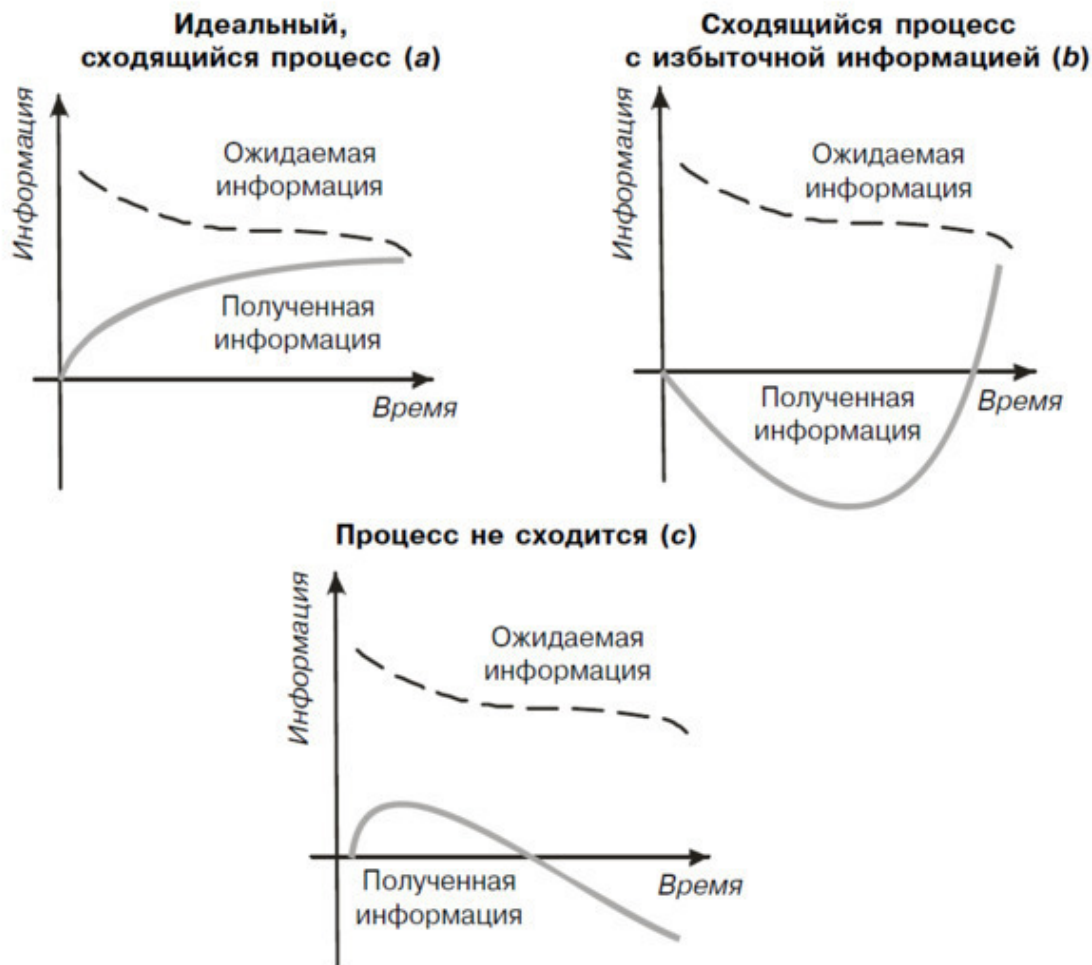


Рис. 1. Траектории развития процессов взаимодействия

Поток слов, поступающих к участнику разговора в процессе обмена сообщениями, вызывает возмущение сознания, которое можно представить себе, как волнение поверхности воды в результате падения капель дождя. Распространение информационных волн в сознании, их интерференция друг с другом и взаимодействие с глубинными внутренними процессами образуют сложную систему, исследование и моделирование которой может быть достигнуто с применением методов и средств, аналогичных тем, которые применяются в физике и позволяют описывать поведение полей и частиц, основываясь на корпускулярно-волновых свойствах материи.

Если считать, что ожидаемое количество информации, которое мы стремимся получить в процессе взаимодействия, должно быть адекватно количеству информации, которое нам будет передано в ответе, мы можем сформулировать принцип информационной адекватности следующим образом:

$$I(A) \sim I(Q)$$

где

$I(A)$ – количество информации в ответе A ;

$I(Q)$ – ожидаемое количество информации в ответ на заданный вопрос Q .

В вычислительной технике широко применяются информационные характеристики различных видов устройств: памяти, процессоров, каналов связи и т. п., а из теории информации хорошо известны способы измерения информационных свойств потоков сообщений. Мы можем использовать аналогичные меры для оценки качества процессов взаимодействия и состояния, участвующих в этом процессе систем. Однако если большинство информационных характеристик в вычислительной технике являются статическими, то процессы человеческого общения имеют ярко выраженный динамический характер. Моделирование таких процессов, определение их информационных характеристик, нахождение оптимальных траекторий взаимодействия и т. п. относятся к группе наиболее сложных и трудоемких задач в современном программировании. Среди основных факторов, определяющих их сложность можно выделить, в первую очередь, следующие:

- *большое количество динамических параметров;*
- *постоянная адаптация и развитие внутренней структуры и функций;*
- *отсутствие четких критериев качества поведения.*

Решение этих задач только лишь средствами традиционного программирования сопряжено с проблемами, корни которых лежат в логических основаниях алгоритмического моделирования. Программирование алгоритмических моделей, основу которого составляют последовательности пассивных логических конструкций, во многих случаях позволяет получить приемлемый результат, но при этом *программа* и *результат* принципиально различаются и отделены друг от друга. В нейронных моделях используются активные элементы, обладающие внутренними динамическими свойствами и способные самостоятельно принимать и передавать сигналы. При этом исполнительные элементы и результаты представляют собой единое целое – динамическое пространство параллельных процессов и их состояний. Одним из способов создания таких моделей является нейронное программирование, под которым мы будем понимать методы организации и управления активными элементами в интерпретирующих системах. Нейронное программирование обладает одним неоспоримым достоинством – природа в процессе эволюции уже построила огромное количество биологических прототипов, широкий спектр которых включает как нервные системы простейших многоклеточных, так и человеческий мозг, самую сложную из известных на сегодняшний день организованных систем, поэтому применение знаний биологии и нейрофизиологии может оказать существенную помощь при программировании искусственных нейронных систем.

Большой разброс в сложности моделируемых систем предполагает применение соответствующих технологий программирования. В такой же степени, как и в алгоритмическом программировании методы построения больших операционных систем отличаются от методов написания небольших индивидуальных программ, в нейронном программировании процесс построения больших моделей существенно отличается от создания небольших по своему объему систем.

Технологии построения информационных систем относятся к области особых интересов в программировании. Среди многих работ в этом направлении можно выделить раннюю статью Э. Дейкстры «*The Structure of „THE“-Multiprogramming System*» [7], в которой он сформулировал принципы построения программных систем, существенно отличающиеся от широко принятых тогда технологий программирования (см., например, Ф. Брукс «*Мифический человек-месяц*» [3]). В этой работе Дейкстра рассматривает не только общие технологические принципы, но также вводит синхронизирующие примитивы – семафоры, при помощи которых ему удается решить проблему асинхронного распараллеливания процессов. Технология

и новые структурно-функциональные решения в его работе взаимосвязаны, и это в совокупности позволяет ему получить решение, значительно опережающее известные в то время системы.

Представления Дейкстры о программировании, его элегантные теоретические и практические решения продолжают оставаться привлекательными для всех, кто связан с этой дисциплиной. В одном из своих последних интервью, голландской телекомпании *VPRO* в 2001 году, рассуждая о сути программирования, он говорит следующее:

Существуют совершенно различные стили программирования. Я могу сравнить их с тем, как сочиняли музыку Моцарт и Бетховен. Когда Моцарт приступал к записи партитуры, композиция у него уже существовала в завершённом виде. Он записывал партитуру с первого раза набело. Бетховен был скептик и борец, который начинал писать до того, как он имел сложившуюся композицию... Для того, чтобы сочинять музыку, нужно уметь записывать ноты. Но быть композитором это вовсе не означает умение записывать ноты. Чтобы быть композитором, нужно чувствовать музыку.

<http://www.cs.utexas.edu/users/EWD/videos/NoorderlichtVideo.html>

Дейкстра затрагивает особенную тему в программировании, связанную с индивидуальным творчеством и использует при этом аналогии с музыкой. Продолжая эту аналогию, можно сравнить необходимые для построения нейронных моделей интерпретирующие системы с различными музыкальными инструментами. Для исполнения больших и сложных музыкальных произведений в симфонических оркестрах используются самые разнообразные группы инструментов. Для исполнения произведений камерной музыки, оказывается вполне достаточно одного. Среди инструментальных систем, предназначенных для индивидуального (камерного) программирования, особый интерес представляет система *HyperCard*. В этой системе удачно совмещены средства представления данных, и система динамической интерпретации скриптов, что позволяет использовать ее как внешнюю оболочку-интерфейс к нейронным моделям. Архитектура системы *HyperCard* во многом послужила прототипом для веб-страниц, однако его возможности до сих пор во многих отношениях превосходят средства динамического программирования *DHTML*.

К сожалению, в 90-х годах система *HyperCard* была заморожена и в настоящее время практически прекратила свое существование. Причина в том, что руководство *Apple* не смогло понять и оценить ее стратегическое значение. В своем интервью, данном в 2003 году *CNET News*, Джон Скали, бывший в 80-х годах президентом компании *Apple*, рассказал об этом так:

Если я обращаюсь назад, на то, чтобы я хотел сделать по-другому тогда, когда я был в Apple, я думаю, что самая большая упущенная возможность – это система HyperCard. Она была создана в 1987 году первым программистом Apple – Билом Аткинсоном. Мы никак не могли понять, что же она из себя представляет на самом деле. Мы думали, что это была система для создания прототипов. Мы думали, что это была база данных. Был пример, когда она использовалась как интерфейс с протоколом TCP/IP для работы с суперкомпьютером Cray. У нас не хватило проницательности, чтобы понять, что все, что было внутри HyperCard, было на самом деле то, что позднее будет успешно разработано Тимом Бернес-Ли, с протоколами HTTP и HTML.

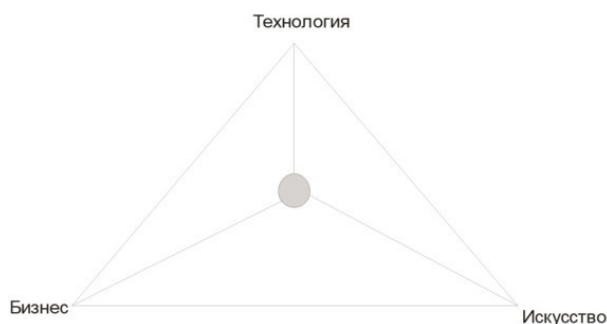
В этом смысле между системой программирования *HyperCard* и нейронными методами моделирования есть много общего. Нейронные модели в конце 60-х годов были определены как

бесперспективные, и понадобились два десятилетия для того, чтобы интерес к этому направлению возродился вновь. Однако до сих пор этот интерес имеет скорее математический, чем программистский характер. Современные исследования свойств нейронных систем, в первую очередь, связаны с математическими задачами классификации и соответственно с нахождением оптимальных методов формирования весов межнейронных связей. Для программистов же, в первую очередь, интерес представляют динамические свойства нейронов, их уникальные способности к соединению, реконфигурации и размножению. Эти свойства нейронных моделей удивительно совпадают со свойствами Интернет, который является саморазвивающейся, децентрализованной системой и в которой происходит огромное количество параллельных процессов.

В основе Интернет лежит обмен сообщениями, которые могут быть представлены в самой разнообразной форме, что позволяет рассматривать его как большую интерактивную систему. При этом взаимодействие может происходить как с участием человека, так и при помощи различных ботов – искусственных представителей, способных автоматически просматривать содержимое веб-страниц, отвечать на вопросы посетителей, делать ставки на аукционах и т. п. Для построения таких ботов используются различные методы программирования, включая нейронное моделирование, которому в последнее время уделяется все большее внимание.

Результатом программирования – нейронного или алгоритмического – в конечном итоге является программный продукт, обладающий определенными коммерческими свойствами. По мере увеличения общего количества пользователей в Интернет, потребности рынка в высокотехнологичных системах, способных упростить процессы общения человека с компьютером, постоянно возрастают. Можно по-разному относиться к известному письму Билла Гейтса – «*An Open Letter to Hobbyists*», в котором он призывает к коммерческому профессионализму в программировании (все программы, которые прилагаются к этой книге, относятся к категории *Open Software* и могут быть использованы свободно, в соответствии с общепринятыми нормами), однако тот факт, что на рынке к нейронным системам сегодня проявляется повышенный интерес, позволяет надеяться, что помимо профессионального любопытства, нейронное программирование сумеет привлечь коммерческий интерес разработчиков.

Судьба программного продукта зависит от множества разнообразных и зачастую противоречивых факторов. Творческие устремления и поиски интересных решений сталкиваются с коммерческими требованиями и технологическими ограничениями. Если представить процесс программирования, как постоянное нахождение упругого баланса между точками в пространстве, которые задают эти ограничения: то искусство является одним из его важных составляющих. Искусство играет принципиальную роль в балансе между практическим смыслом и теоретическими ограничениями именно потому, что оно помогает находить компромиссы и соединять противоречивые взгляды в практике и теории [21]. Стремление к такому соединению, поиск различных подходов, методов и представлений, которые позволят решать все более широкий круг постоянно возникающих перед программистами задач, и является главной целью этой книги.



Энергия, информация и знания

В основе процессов преобразования как энергии, так и информации лежат общие по своей сути принципы, что позволяет предположить, что практические конструкции энергетических машин и информационных систем должны обладать некоторыми подобными свойствами. Основываясь на этих аналогиях, мы попытаемся применить такие физические понятия как работа, мощность, к. п. д., принцип неопределенности и другие к анализу информационных систем и надеемся, что накопленный в индустриальном мире опыт проектирования и эксплуатации механизмов и энергетических устройств поможет в проектировании информационных машин.

С развитием информационных технологий возникли такие виды деятельности как виртуальные предприятия, электронная торговля, дистанционное обучение, удаленная диагностика, информационно-поисковые сервисы и т. п., открывшие перед человечеством принципиально новые возможности, но одновременно с этим появился и ряд новых практических и теоретических проблем. Среди них – извлечение знаний из больших распределенных источников данных, удаление мусора из потоков сообщений, взаимодействие на естественных языках, интеграция разнородных каналов связи, и другие, которые в свою очередь вызвали необходимость пересмотра и расширения некоторых из методов и технологий современного программирования. Так, например, увеличение доступных объемов, данных во многих случаях не только не приводит к положительному результату, а наоборот, вызывает переполнение каналов связи и ухудшение качества принятия решений. Попытки создания унифицированных интерфейсов и инвариантных, по отношению к конкретному человеку, форм представления данных, отторгаются пользователями вследствие естественного стремления людей к индивидуализму. Навигационные меню и подсказки, зачастую вместо сокращения траектории просмотра, значительно увеличивают время поиска нужной информации. Один и тот же запрос в поисковый сервис, посланный из двух соседних компьютеров – может вернуть различные результаты. Если до появления Интернет, задачи обработки данных были в основном связаны с поиском и восстановлением недостающей информации из ограниченных локальных объемов данных, то в настоящее время все более актуальными становятся индивидуальная фильтрация и преобразование информации из постоянно возрастающих и практически неограниченных потоков сообщений, приходящих извне. При этом, статические, основанные на однозначных логических выражениях формы представления, используемые в традиционном теоретическом программировании, не соответствуют реалиям Интернет, где программы, структуры и данные находятся в постоянной динамике.

Среди множества практических задач, которые решаются в вычислительной технике на протяжении всего периода ее существования – подготовка данных относится к категории «вечных». На ее примере можно проследить эволюцию нескольких поколений технических и программных средств организации данных, которая сегодня привела к возникновению центров дистанционного обслуживания (*Call Centers*). В 2005 году в мире насчитывалось более 70 000 таких центров, в которых работало более 3,5 миллионов человек. Одним из главных критериев успешного развития бизнеса является постоянное повышение качества обслуживания пользователей и становится очевидным, какое стратегическое значение приобретают эти центры в современном деловом мире.

С момента появления перфолент и перфокарт, структуры данных и технологии их обработки постепенно трансформировались из системы подготовки и накопления данных, в системы реагирования на запросы, поступающие в реальном времени. Современные центры дистанционного обслуживания являются сложными коммутационно-диспетчерскими комплексами, которые во многих случаях территориально распределены по всему миру. Одна

из ключевых функций таких центров – принять сообщение, определить его смысл и перекоммутировать абонента к соответствующему сервисному подразделению. На сегодняшний день сложилась ситуация, когда количество людей, инициирующих различные запросы или сообщения по телефону, в виде электронной почты или в чате, уже значительно превосходит доступный персонал, и очевидно, что по мере глобализации международного сообщества эта проблема будет все более усугубляться. Создание виртуальных представителей, способных в определенной мере понимать входные запросы и адекватно реагировать на них, по возможности отвечая или переключая на соответствующие сервисы, является одним из возможных решений этой проблемы (рис. 2). Для того, чтобы это решение было эффективным, необходимо, чтобы технологии обучения виртуальных агентов были просты и доступны для авторов, которые во многих случаях не являются профессиональными программистами.

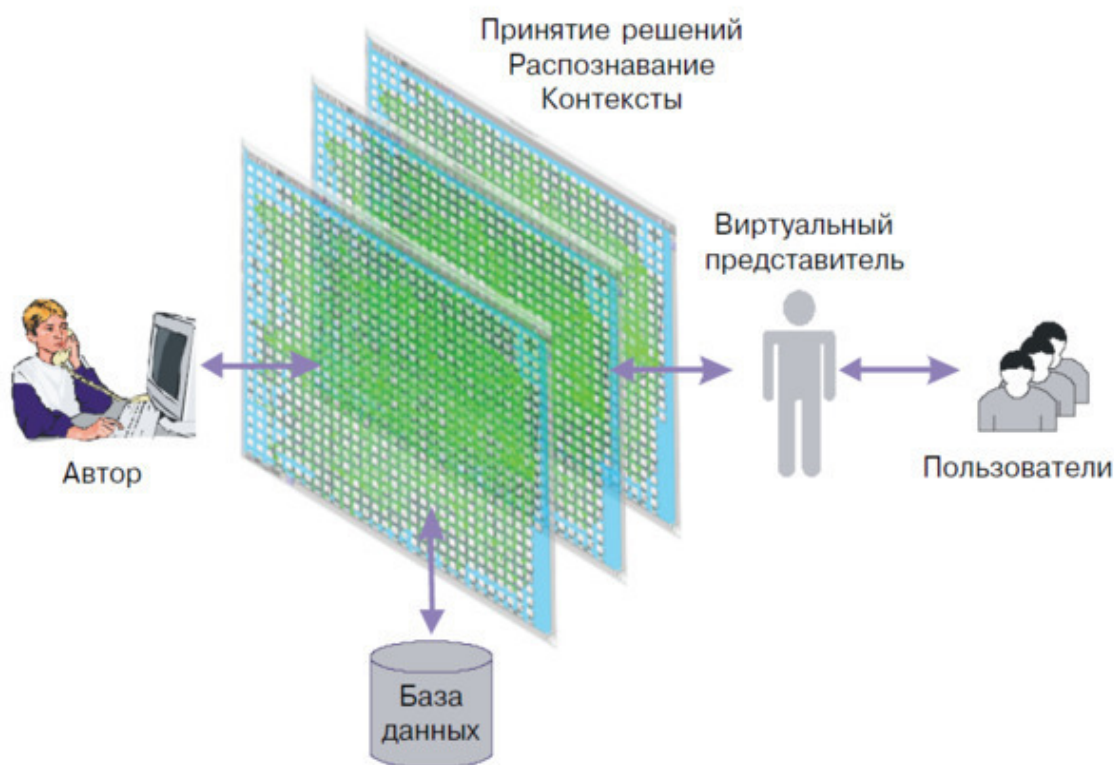


Рис. 2. Виртуальный представитель

Количество циркулирующих в Интернете бит информации сопоставимо с числовыми характеристиками физических объектов на микро- и макроуровнях. В физике при переходе от одного уровня представления к другому кардинально меняется аппарат исследования – статистическая термодинамика, механика, молекулярная физика, представляют собой принципиально различные, но в то же время хорошо согласованные разделы одной науки. В отличие от физики, в вычислительной технике в настоящее время еще не сложились общепринятые теоретические основания, в рамках которых конструктивно объединяются представления, методы анализа и моделирования, подобные соответствующим разделам физики. Классическая теория информации, заложенная Шенноном в 40-х годах и основанная на анализе последовательностей символов, поступающих из источников, данных в приемник, равно как и булева логика, имеющая дело с двоичными, точно заданными значениями, перестают работать в тех случаях, когда речь идет об информационных сообщениях, на много порядков превосходящих по своей мощности возможности приемника. В физике традиционно применяются фено-

менологические и аналитические методы, которые позволяют легко переходить от профессиональных теоретических моделей к упрощенным представлениям, доступным для широкой публики. Например, такие соотношения как зависимость между температурой t , давлением P и объемом V , в термодинамике, или понятия к. п. д., работа, мощность, энергетические потери, хорошо известные из школьной программы, с достаточной точностью и степенью взаимопонимания согласовывают представления потребителей, инженеров и ученых. Такое масштабирование теоретических и практических знаний является одним из необходимых условий для успешной интеграции научных исследований вместе с проектированием, производством и применением как энергетических, так и информационных машин и систем.

Можно считать, что идеи информационного усиления и нейронного программирования впервые были сформулированы в 1945 году, когда появились две работы, во многом предопределившие развитие вычислительной техники на несколько десятилетий вперед – отчет фон Неймана «*First Draft of a Report on the EDVAC*» [20] и статья Вэннивера Буша в журнале *The Atlantic Monthly* – «*As We May Think*» [4]. Удачно взаимодополняя друг друга, формальные модели элементов и структур автоматических цифровых вычислительных устройств сочетаются в них вместе с эскизами и перспективой развития будущих систем. Функциональная схема ЭВМ фон Неймана и гипертекстовая модель знаний Буша появились в тот момент, когда абстрактные рассуждения о природе вычислений, логике мышления и познании начали находить практическое воплощение в виде реальных информационных систем. В это время завершается латентный этап в истории вычислительной техники и начинается последовательное развитие ее архитектурных направлений, которые на сегодняшний день можно условно разделить на три периода:



Математические расчеты и вычислительные алгоритмы в той или иной форме доминировали в теории и практике вычислительной техники вплоть до середины 70-х годов, когда актуальным становится логическое и функциональное программирование, предназначенное для обработки символьных и текстовых данных. В середине 90-х, с возникновением *Web*, на передний план выходят задачи обработки знаний – управление распределенными потоками сообщений, лингвистический анализ и синтез, распознавание образов, интеграция различных форм и способов общения (телефоны, радио и телевидение) вместе с компьютерами.

Соответственные изменения произошли за это время и в методах программирования – языки, ориентированные на обработку структурированных данных и потоков сообщений, например, *C++* и *Java*, существенно отличаются от первого поколения языков для числовых и алгоритмических вычислений, таких как *FORTRAN* и *ALGOL*. Главное отличие между ними заключается в степени интеграции программных элементов и объединении функций обработки вместе со структурами данных, что привело к появлению объектов – качественно новой категории в языках программирования.

Можно предположить, что развитие направления обработки знаний приведет к дальнейшей интеграции программных объектов вместе с другими, в первую очередь, внешними структурами данных и распределенными функциями, и как следствие, к появлению новых способов

управления ансамблями разнородных систем. Результаты этой интеграции уже сейчас можно наблюдать в современных объектных базах данных, в системах программирования Агентов (*Agent Oriented Programming*)

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.