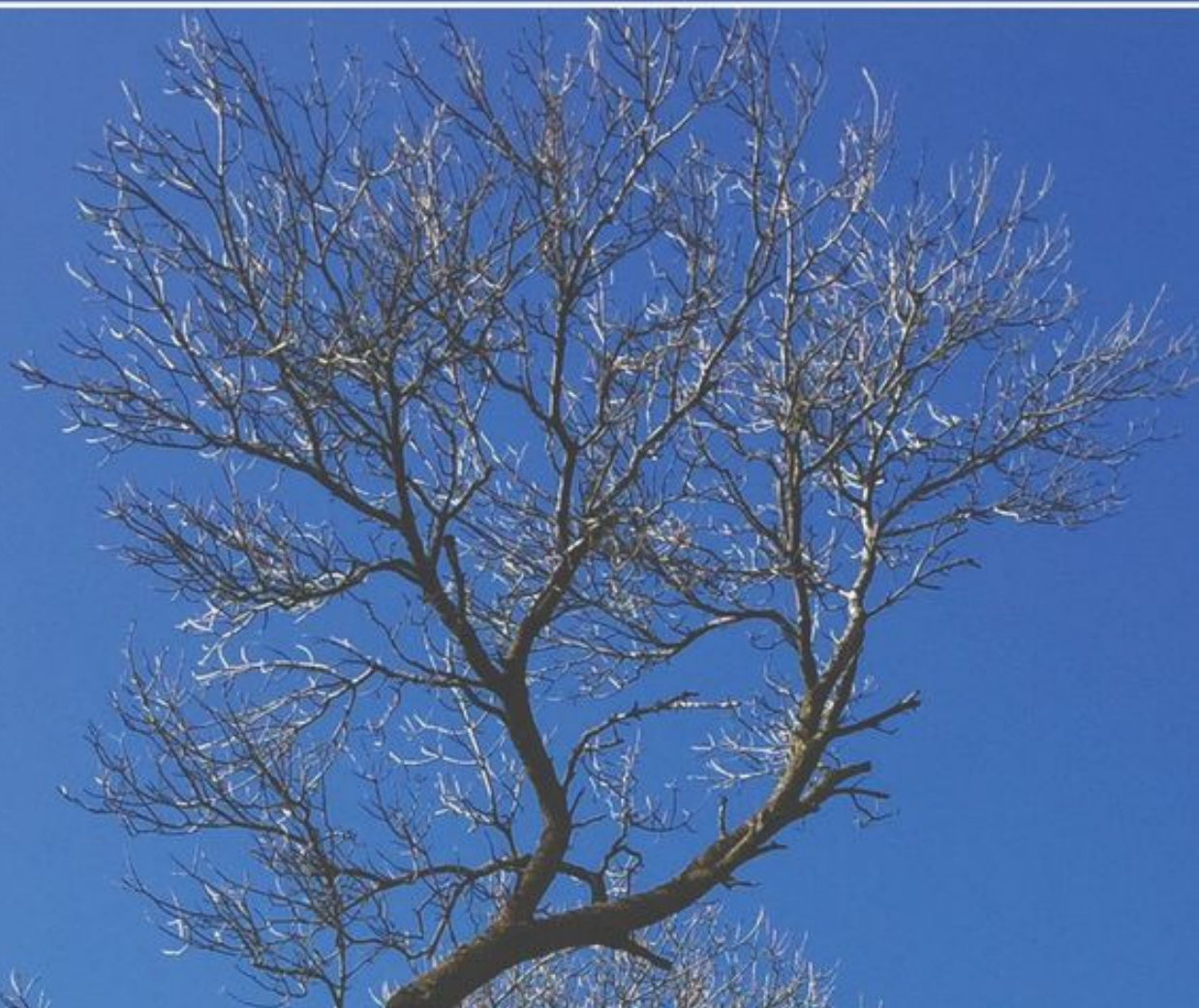


18+ Сергей Толкачев

*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
Усилители, нейроны и Интернет	15
Программирование реакций	21
Среда нейронного программирования	25
Взаимодействующие системы	29
Конец ознакомительного фрагмента.	33

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 ($I_{ож}$), 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 ($I_{ож}$). 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 ($I_{ож}$) и ($I_{ож}$). *желаемый результат a b c*

. Траектории развития процессов взаимодействия **Рис. 1**

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:

где

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:

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 и 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. *программа результат*

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 «» [7], в которой он сформулировал принципы построения программных систем, существенно отличающиеся от широко принятых тогда технологий программирования (см., например, Ф. Брукс «» [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. *The Structure of „THE“-Multiprogramming System Мифический человеко-месяц*

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 в 2001 году, рассуждая о сути программирования, он говорит следующее: *VPRO*

<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 . 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 во многом послужила прототипом для веб-страниц, однако его возможности

до сих пор во многих отношениях превосходят средства динамического программирования .
HyperCard HyperCard DHTML

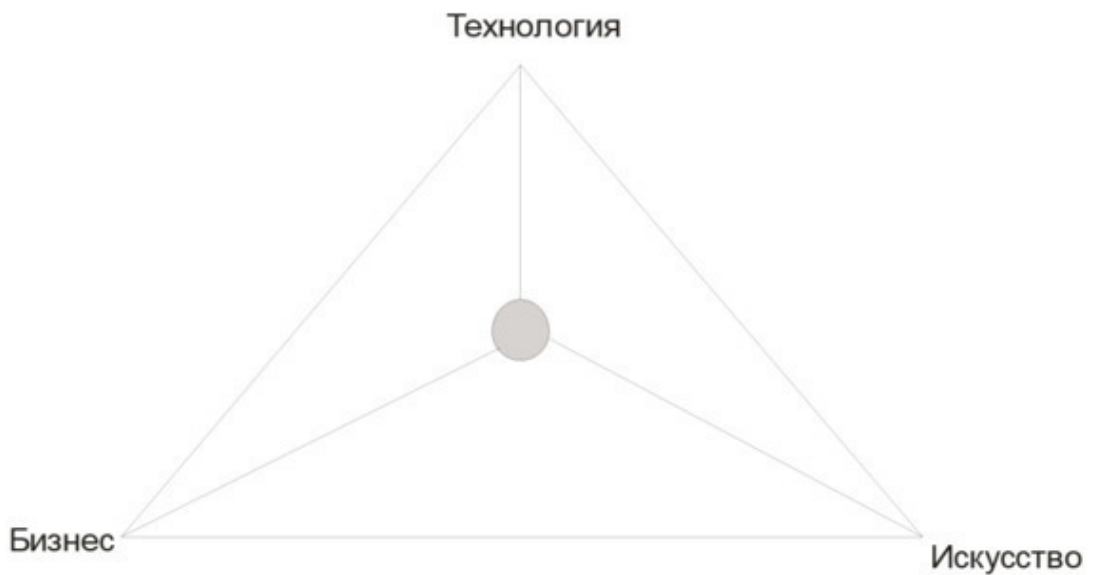
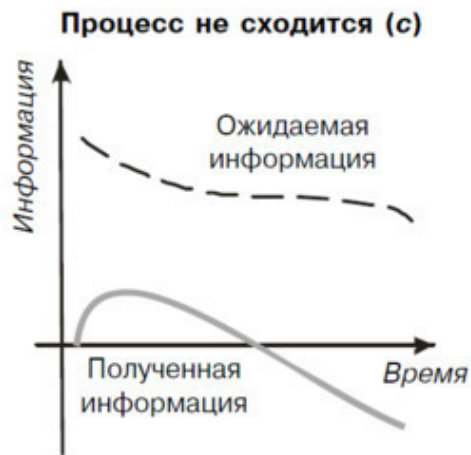
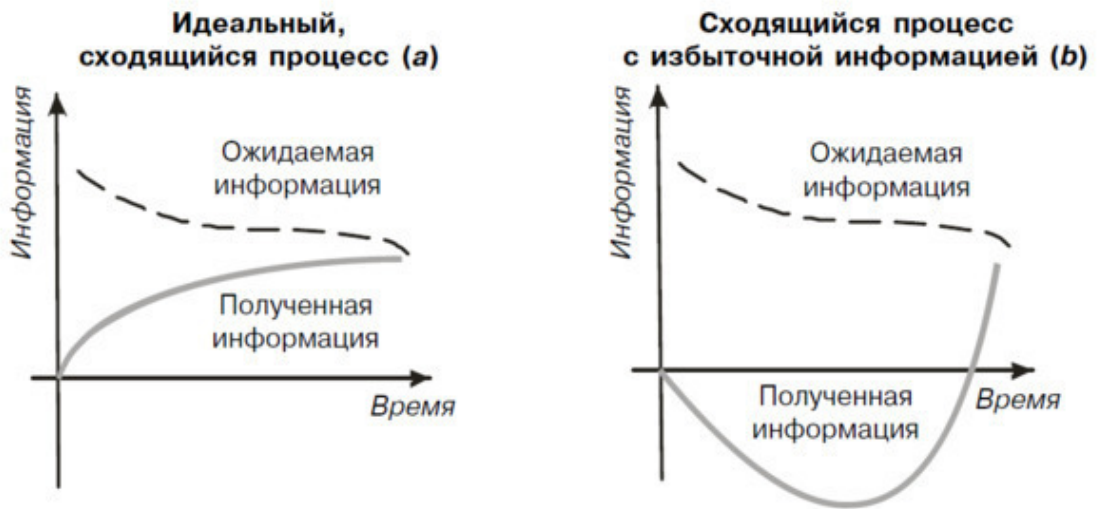
К сожалению, в 90-х годах система была заморожена и в настоящее время практически прекратила свое существование. Причина в том, что руководство не смогло понять и оценить ее стратегическое значение. В своем интервью, данном в 2003 году, Джон Скали, бывший в 80-х годах президентом компании, рассказал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

В этом смысле между системой программирования 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. *HyperCard*

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 – «», в котором он призывает к коммерческому профессионализму в программировании (все программы, которые прилагаются к этой книге, относятся к категории 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. *An Open Letter to Hobbyists Open Software*

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.



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

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

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

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

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 завершенном 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.

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 TSP/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.

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 (). 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. *Call Centers*

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.

Виртуальный представитель **Рис. 2.**

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 феноменологические 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 «» [20] и статья Вэннивера Буша в журнале – «» [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: *First Draft of a Report on the EDVAC The Atlantic Monthly As We May Think*

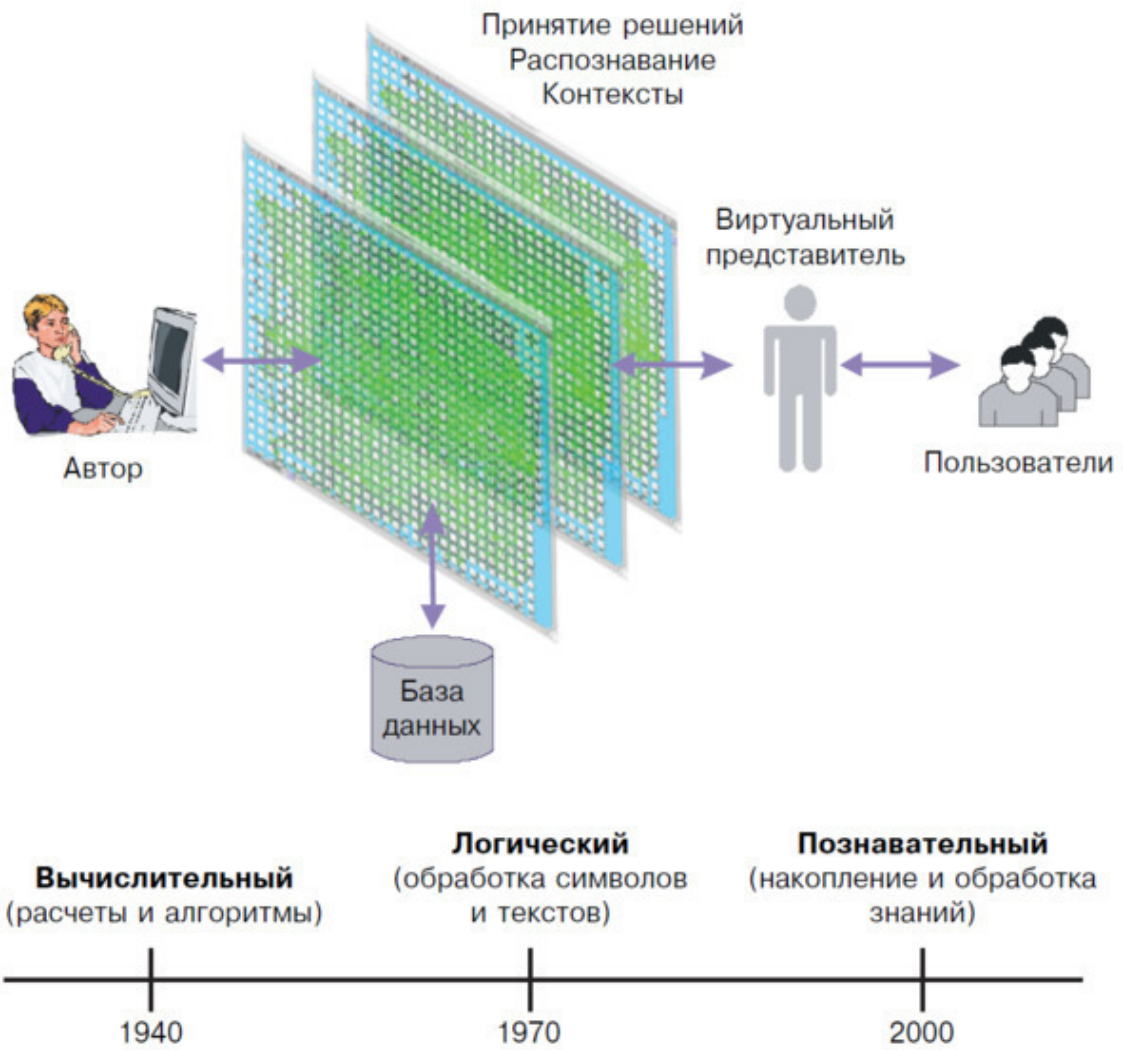
Математические расчеты и вычислительные алгоритмы в той или иной форме доминировали в теории и практике вычислительной техники вплоть до середины 70-х годов, когда акту-

альным становится логическое и функциональное программирование, предназначенное для обработки символьных и текстовых данных. В середине 90-х, с возникновением , 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. *Web*

Соответственные изменения произошли за это время и в методах программирования – языки, ориентированные на обработку структурированных данных и потоков сообщений, например, и , существенно отличаются от первого поколения языков для числовых и алгоритмических вычислений, таких как и . 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. *C++ Java FORTRAN ALGOL*

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 () и, наконец, в наиболее широком виде, в динамических свойствах веб-страниц нового поколения. Концептуально можно предположить, что будет обладать значительно более широким спектром динамических свойств, программирование которых будет осуществляться с применением , , и других аналогичных средств, способных организовать взаимодействие большого количества гетерогенных объектов, образующих сложные семантические сети. Биологические системы в этом смысле являются хорошим образцом для сравнения, поскольку они прекрасно приспособлены для взаимодействия с большим количеством сложных разнородных объектов во внешнем мире и в настоящее время в достаточной степени изучены, чтобы знания об их свойствах могли быть использованы при разработке новых методов программирования. *Agent Oriented Programming Web 2.0 Ajax DHTML XMLHTTP*

Можно считать, что в той или иной мере, все задачи обработки информации связаны с обработкой знаний. Применение эффективных методов накопления знаний является главной целью всех современных информационных технологий как в глобальном смысле, так и в локальных приложениях. Индивидуальные знания отдельных сотрудников в сочетании с корпоративными знаниями и данными образуют весомую составляющую системы ценностей в современном информационно-индустриальном мире. Так, например, уход или перемещение работника предприятия не должны сопровождаться потерей его индивидуального опыта, накопленного во время работы на определенном месте. Одним из путей решения этой проблемы является применение систем обучения (), способных воспринимать знания эксперта и передавать эти знания заинтересованным лицам. Такие системы являются традиционными в среде образования – в 2003 году объем их продаж составил более 2 миллиардов долларов, однако в последние годы они находят все более широкое применение в промышленности и сфере обслуживания. Динамические и семантические свойства могут позволить организовать процесс накопления и передачи знаний естественным эволюционным путем, сохраняя при этом привычную «человеческую» форму общения между экспертом и пользователями. Однако превращение данных в знания в такой же степени, как и превращение энергии в полезную работу потребует специальных механизмов в качестве которых мы и будем использовать нейронные модели. *E-learning Web 2.0*



Усилители, нейроны и Интернет

В статье «» [4], рассуждая о последующих поколениях систем, которые смогут повысить эффективность исследовательской работы человека, Буш очень образно пишет о принципиально новых формах энциклопедий, которые будучи соединенными с персональным устройством – , смогут комплексы ассоциативно-связанных знаний и тех следов, которые в этих знаниях может оставить исследователь. Перефразируя Буша, можно сказать, что информационное усиление – это извлечение гипертекстовых данных из динамических распределенных сетей и их преобразование в процессе интерактивного взаимодействия с человеком. Буш был по-видимому первым, кто достаточно конкретно определил основные компоненты интеллектуального усилителя: ассоциативно-связанные динамические данные, персональное устройство-преобразователь, пользовательский интерфейс. *As We May Think memex* **усиливать**

Усилитель электрических сигналов на транзисторе **Рис. 3.**

Для того чтобы лучше понять принцип работы информационного усилителя, рассмотрим, как происходит усиление в простейшем транзисторе. Если опустить технические детали, связанные с его внутренним устройством, транзистор является преобразователем, подсоединенным к источнику энергии, на управляющий вход которого поступает слабый сигнал, а на выходе получается сигнал, усиленный по мощности (рис. 3).

В самом начале своей статьи, Буш подчеркивает, что экономика – это фактор который превращает идеи в реальность. Ни талант Лейбница, который в 1673 году сумел создать вычислительную машину, близкую по своим характеристикам к современному арифмометру, ни все ресурсы фараона, обладай он при этом всеми знаниями о современных технологиях, не в состоянии превратить идею в практически работающий продукт, если при этом стоимость разработки и эксплуатации не будет находиться в пределах экономической целесообразности [4].

Понадобилось шесть десятилетий со времени появления работы Буша, для того чтобы информация наряду с энергией оказалась экономически доступным ресурсом для большей части человечества. И этот ресурс может быть практически свободно использован любым человеком на Земле через Интернет – огромную, динамичную и постоянно расширяющуюся инфраструктуру данных (рис. 4), которая впитывает, хранит и отражает практически все, что происходит в естественной природе или создается человеком. Большинство данных в Интернете первого поколения представлено в виде страниц, созданных так, как если бы они были предназначены исключительно для чтения человеком, хотя физически он в состоянии прочесть только очень малую часть того, что может быть получено из миллионов веб-серверов.

Картина Интернет на 1 января 2000 года **Рис. 4.**

За сравнительно короткий период, прошедший с момента появления первых статических страниц в формате , Веб стремительно эволюционировал в направлении повышения функциональности страниц и расширения способов их передачи. Дж. Мартин подметил, что когда в нашем мире появляются принципиально новые технологические достижения, некоторое время они продолжают использовать старое содержание [17]. Например, в телевидении достаточно долгое время, начиная с момента его появления, большую часть новостей заполняли читающие текст дикторы, а телевизионное изображение было лишь фоном для аудиосообщений. Так же и в первом поколении Интернет, используются страницы, которые фактически дублируют печатную продукцию. Однако уже сегодня произошла практическая интеграция всех основных способов передачи данных, и у разработчиков появилась возможность применять как комбинации различных устройств (компьютеры, телефоны, радио и телевидение), так и новые функционально-активные объекты, вместо пассивных страниц текста, для организации общения с человеком. *HTML*

Статический, динамический и интерактивный Веб **Рис. 5.**

Гипертекстовые ссылки внутри страниц, которые первоначально образовывали статические сети, интенсивно развиваются, трансформируясь в динамические образования, в которых они интегрируются вместе со сложными структурами хранимых данных (рис. 5). Страницы превращаются в активные системы, которые в свою очередь, объединяются в кластеры, которые могут порождать новые структуры, и так далее, что удивительно напоминает поведение биологических или физических групп взаимодействующих объектов и вполне соответствует представлениям Буша о следах, которые человек может оставлять в системах знаний.

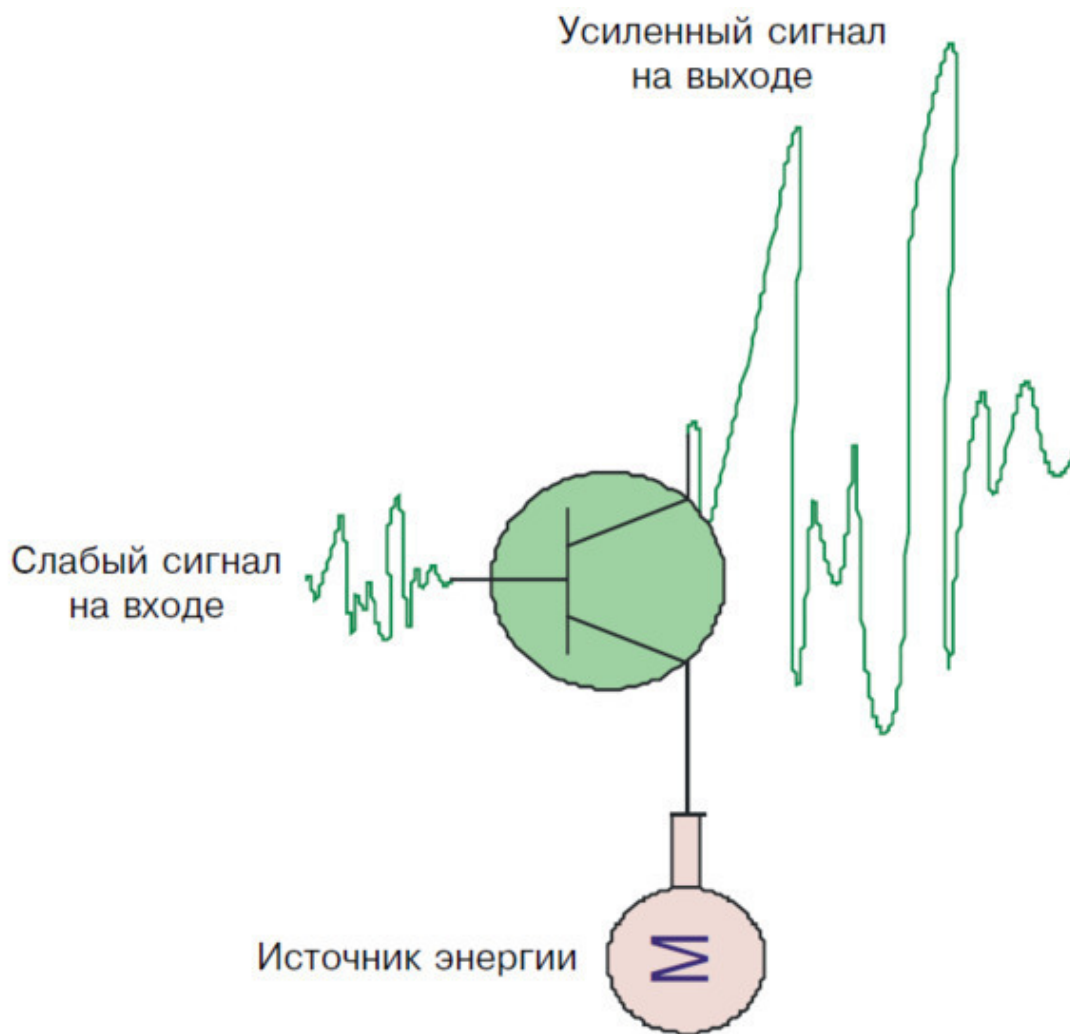
Вольтамперная характеристика транзистора **Рис. 6.**

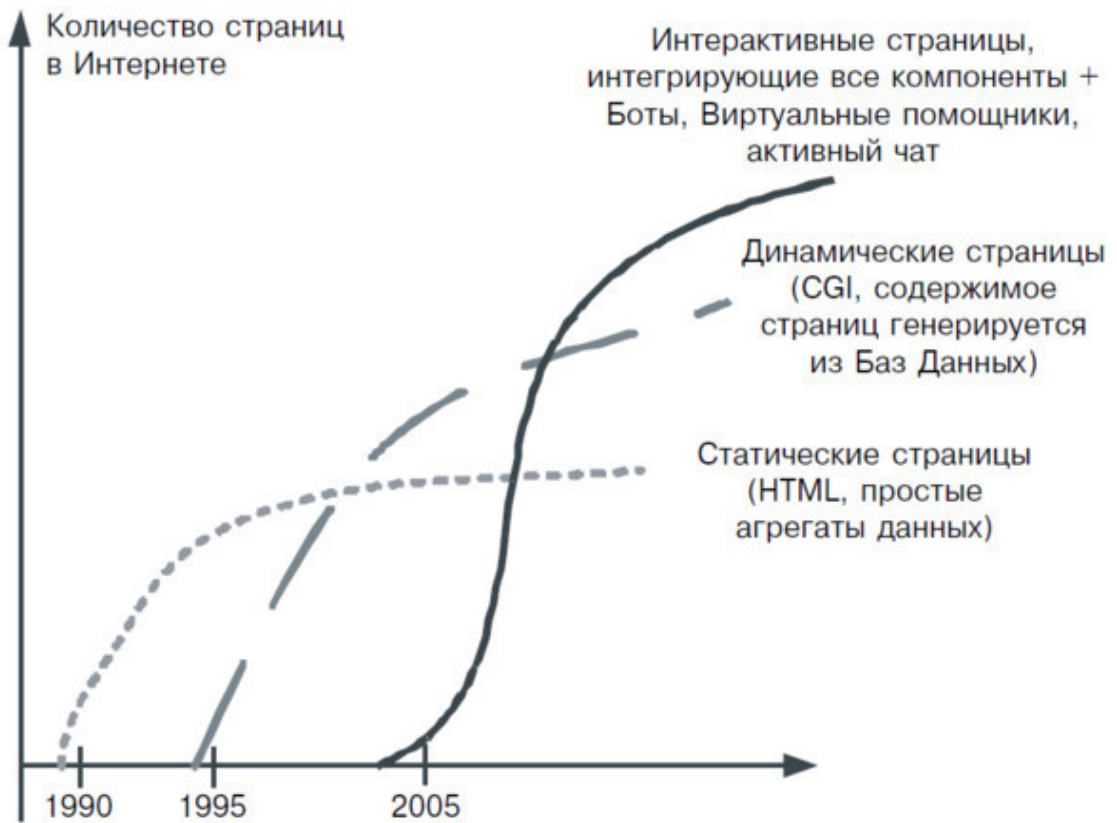
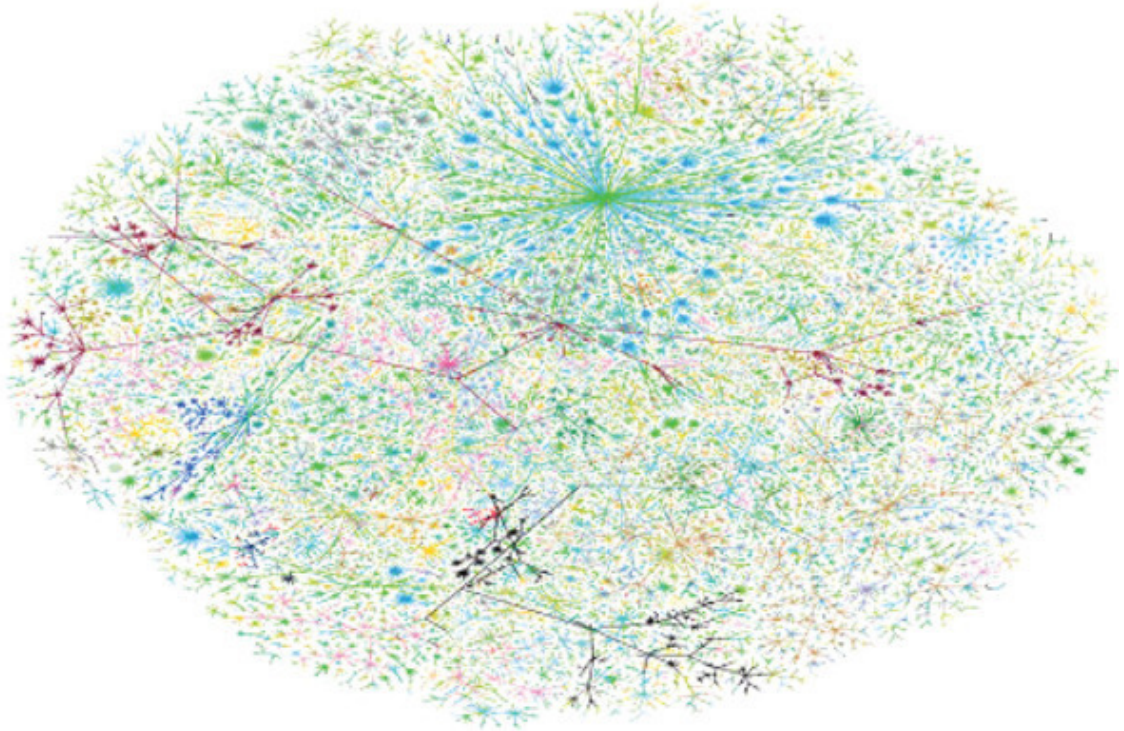
В своем отчете по структуре ЭВМ фон Нейман использует нейрон в качестве прототипа для базовых вычислительных элементов автоматической цифровой вычислительной системы и модель биологической нервной системы лежит в основе его более общих рассуждений о вычислительных структурах. Поскольку проблема, которую решали разработчики ЭВМ, в то время имела исключительно вычислительный характер, фон Нейман рассматривает в первую очередь арифметические свойства нейрона и использует его как цифровой двоичный элемент, который может выполнять базовые математические функции.

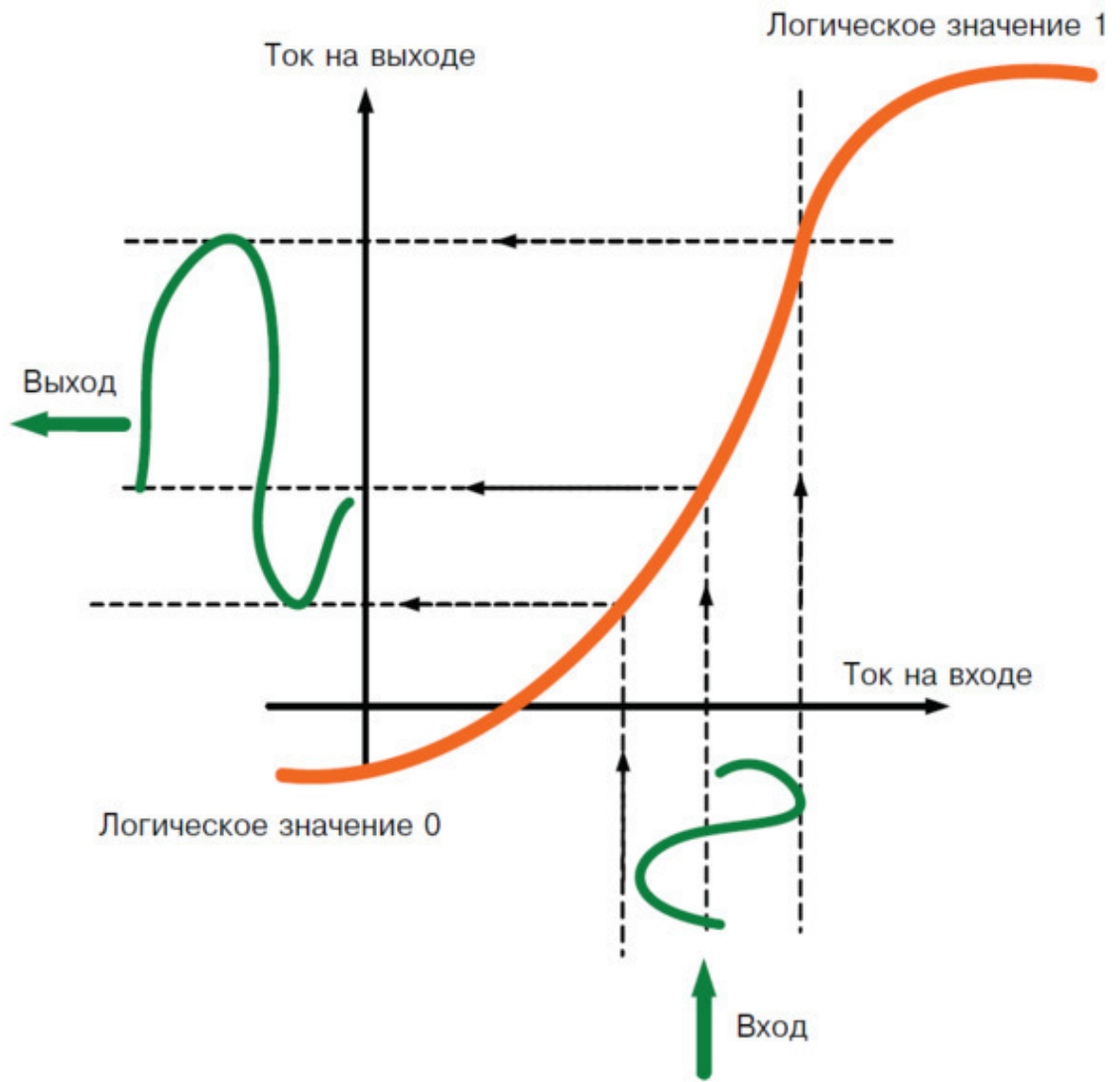
Появившиеся в 1948 году транзисторы, обладающие двумя устойчивыми состояниями (рис. 6), оказались вполне удачным решением для представления двоичных данных, и аналогии между транзистором и нейроном, как двоичными устройствами, надолго закрепились в теории и практике вычислительной техники. В это же время в других направлениях электроники широко используется свойство транзисторов по преобразованию сигнала в зоне переходного процесса между устойчивыми состояниями. Такое преобразование в транзисторах всегда носит приблизительный характер и обладает искажениями, которые могут быть в определенной степени скомпенсированы при помощи различных дополнительных элементов. Дуализм транзистора, в котором дискретность сочетается с непрерывностью, является хорошим примером комбинации точных и приблизительных () свойств в одном устройстве. Нейрон вполне соответствует этой аналогии, соединяя в себе определенную устойчивость вместе с промежуточными приблизительными состояниями, которые он может сохранять в зависимости от характера возбуждения и своего функционального назначения. *fuzzy*

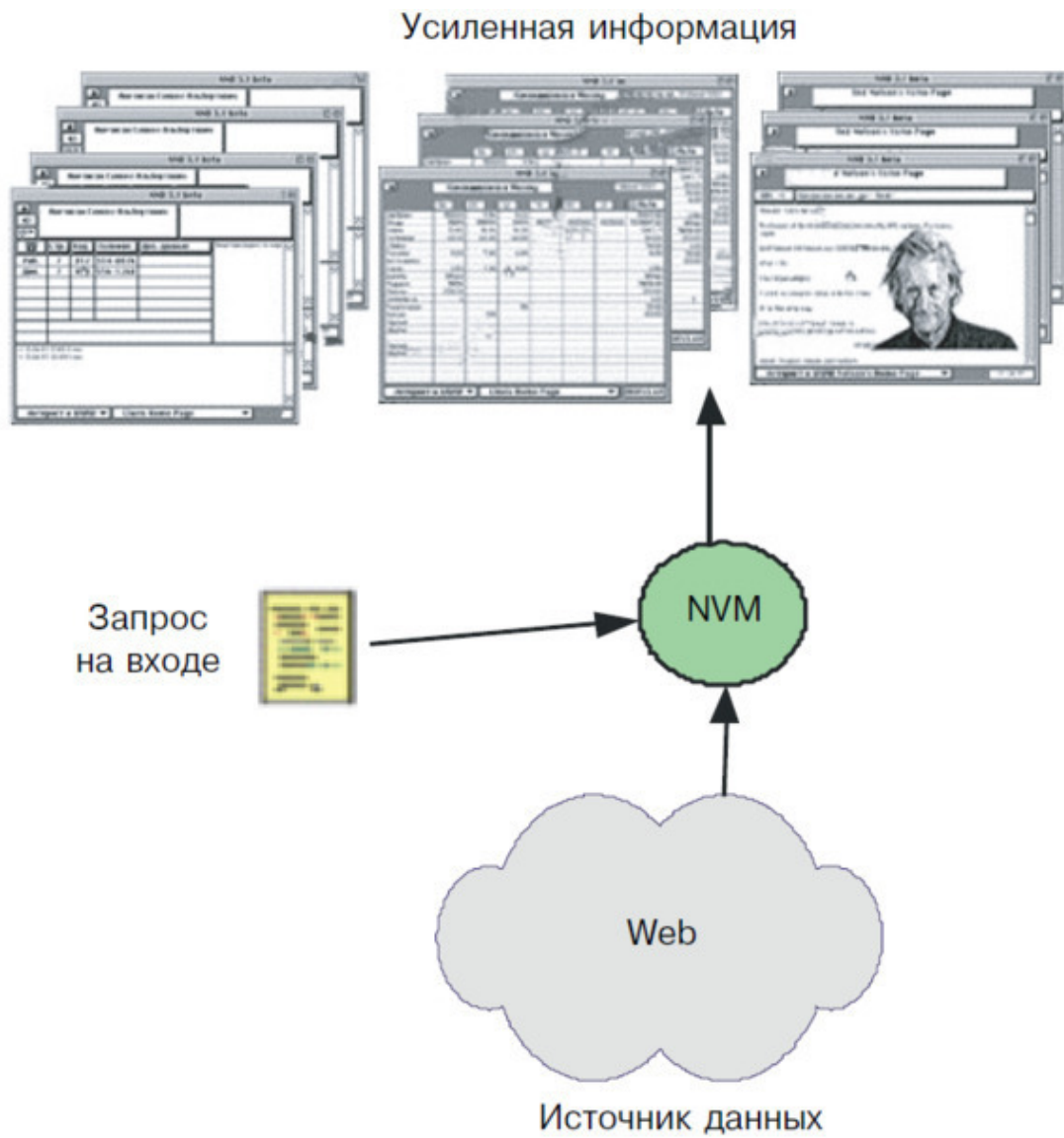
Если допустить, что информационный усилитель (рис. 7) имеет структуру, в общем виде похожую на структуру усилителя электрических сигналов, тогда в качестве универсального источника информации в такой схеме может быть использован Интернет, входные и выходные сигналы могут быть представлены посредством аудио и текстовых сообщений устной или письменной речи, а преобразователь может быть реализован в виде некоторой виртуальной машины – (), которая может быть загружена в персональную ЭВМ или иное устройство, аналогично (). Такой информационный усилитель может быть встроен в самые разнообразные системы. Можно предположить, что уже в недалеком будущем успехи нано-технологии позволят иметь сенсорные устройства, встроенные непосредственно в человеческий организм, и в этом случае форма общения будет отличаться от речевой. Именно поэтому мы надеемся, что нейронные модели являются тем самым адаптивным механизмом, способным к естественной интеграции с будущими симбиозными человеко-машинными системами. *Neural Virtual Machine NVM Java Virtual Machine JVM*

Информационный усилитель **Рис. 7.**









Программирование реакций

Продолжая поиск аналогий, которые могут помочь нам в проектировании такого усилителя, обратимся к работе И. М. Сеченова «» [37]. Анализируя его поведение, Сеченов рассматривает мозг как «черный ящик», который в конечном итоге, реагируя на возбуждения чувствующих нервов, все свои внешние проявления сводит к мышечному движению. Попробуем представить информационное устройство в окружении программных модулей, каждый из которых может быть вызван к исполнению в результате возбуждения связанного с ним нервного окончания. Тогда все внешние проявления внутренних процессов этого устройства могут быть сведены к вызову и запуску соответствующих программных модулей. Такой вызов может произойти в тот момент, когда уровень возбуждения, связанного с этим модулем нейрона в выходном слое информационного устройства, превысит некоторое пороговое значение.

Рефлексы головного мозга

Если допустить, что с распределенными в интернет-сети программами может быть ассоциирован уникальный адрес и своя интерпретирующая среда, тогда их выполнение сводится к посылке запроса (например,) из одной распределенной системы в другую (рис. 8). В этом случае результатом работы нейронного слоя будет являться исполнение множества асинхронных параллельных процессов, каждый из которых может возвращать данные обратно в исходную систему. Такой способ вызова программных модулей позволяет существенно упростить реализацию интерфейсов и свести задачу к ответу на вопрос: каким образом усиление и торможение сигналов может привести к адекватному реагированию в тех случаях, когда с этими сигналами ассоциированы слова и смысловые значения? *HTTP*

Вызов и выполнение программных модулей в нейронной среде **Рис. 8.**

В физиологии разделение всех реакций на безусловные и условные связано с практической невозможностью проследить все логические цепочки последовательных действий, которые в конечном счете вызывают соответствующие мускульные сокращения. Такое разделение, на первый взгляд, не имеет принципиального значения в программировании, где все потоки действий обладают определенным детерминизмом, что позволяет быть уверенными в результатах и оценках точности решения в каждом отдельно взятом случае. Ситуация меняется существенным образом, когда речь идет о сотнях миллионов компьютеров, распределенных в сети Интернет. В этом случае программист попадает в ситуацию, аналогичную для физиолога и физика, когда переход от одного уровня представления к другому предполагает применение иного способа восприятия и исследования – точные знания о коде и предсказании его поведения, не имеют смысла в системах, где изменения самих кодов и данных происходят в таких масштабах и с такими скоростями, что мы не можем получить детерминированную картину всех состояний.

Уже упоминавшаяся выше система была включена компанией в состав операционной системы еще в 1987 году, задолго до появления . Его автор – Билл Атkinson, один из ведущих программистов в , построил интегрированную систему, которая в самое короткое время сумела увлечь миллионы пользователей к процессу, который сегодня называется -программирование. Основу составляет рабочее поле, которое называется карта. На поверхности карты можно создавать как в ручную, так и динамически, предопределенные объекты – кнопки, текстовые поля и изображения. С каждым объектом может быть связан интерпретируемый код – скрипт. Исполнения скриптов осуществляются через простой механизм управления событиями. Карты объединяются в группы, группы объединяются в наборы. работает в режиме непосредственной интерпретации и имеет очень своеобразную и легкую систему идентификации и адресации объектов. Концепция n-мерного программируемого пространства в является весьма привлекательной по причине удобства и простоты доступа и исполнения скриптов. Мы будем исполь-

зовать в качестве прототипа исполнительных среды терминологию и архитектуру, близкую к , поскольку в современном существуют практически все необходимые инструменты для создания его аналога. Погружение нейронной виртуальной машины в такую исполнительную среду можно сравнить с подключением нервной системы к мускулатуре и органам чувств (рис. 9).
HyperCard Apple Mac OS DHTML Apple Web HyperCard HyperCard HyperCard HyperCard DHTML

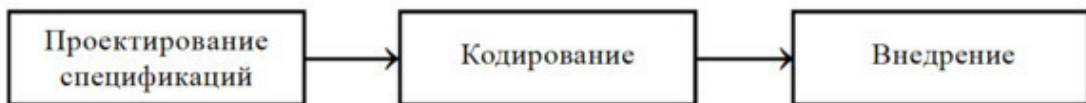
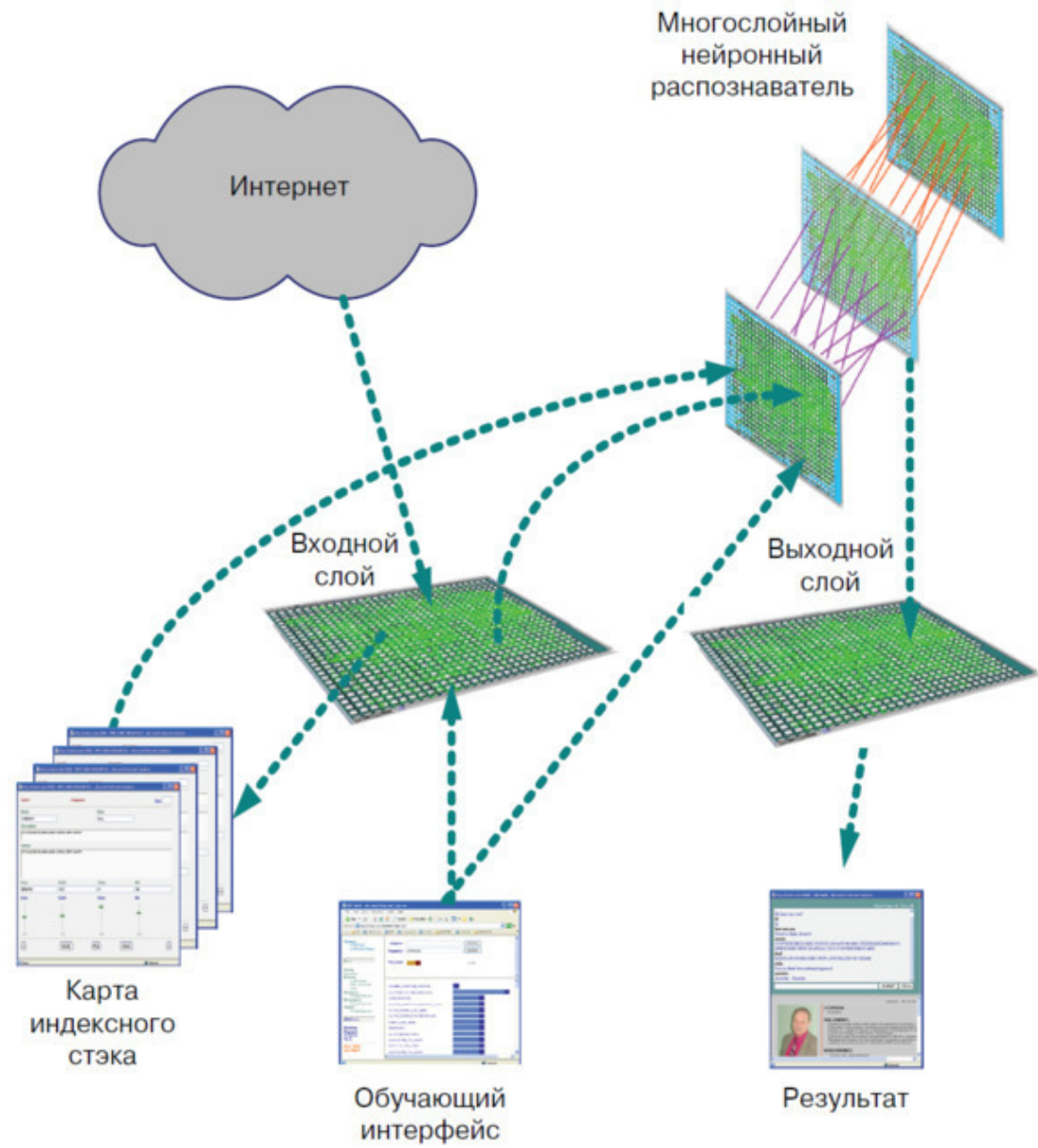
Взаимодействие между слоями нейронного ядра, Интернет и интерфейсными картами

Рис. 9.

Нейронное программирование по сути представляет собой процесс, который сводится к созданию объектов, установлению между ними ссылок-отношений и определению динамических правил их поведения по передаче возбуждений и исполнению реакций. В случаях, когда количество узлов не велико, построение нейронных моделей является тривиальной задачей. Затруднения начинаются тогда, когда возникает необходимость связать большое количество нейронов через большое число слоев с внешним механизмом, в составе которого может быть много исполнительных функций. Масштаб задачи определяет технологию ее решения, и в программировании используются различные методы анализа и синтеза сложных систем, предшествующие процессу кодирования. Эти методы можно подразделить на три большие группы: блок-схемы и диаграммы потоков данных и управлений, таблицы принятия решений и структурно-функциональные схемы. Применение предварительных методов анализа и спецификации позволяет значительно повысить производительность всего процесса программирования, в первую очередь, за счет специализации и распределения работ по проектированию наиболее оптимальным образом. Работы по структурно-функциональному моделированию, основанные на декомпозиции и определении функциональных свойств модулей, такие, например, как метод , разработанный в 70х годах фирмой [24], заложили основу для построения визуальных моделей достаточно сложных систем. Весь процесс программирования в этом случае можно представить в виде следующей последовательности шагов: *HIPO IBM*

Технология проектирования диалогов и их лингвистического содержания может быть сведена к традиционным методам проектирования программного обеспечения. На первом этапе, формируется лингвистическая модель, которая затем кодируется в виде нейронных структур. Нейронные структуры обучаются и корректируются в процессе общения с экспертом, однако в отличие от традиционных программ, они продолжают обучаться и развиваться на протяжении всего периода жизни, даже после внедрения их в эксплуатацию.





Среда нейронного программирования

Для создания и поддержания нейронных моделей нам понадобится исполнительная среда, в которой будут происходить интерпретации, структурно-функциональные модификации, а также выполнение связанных с этими моделями подпрограмм-реакций. На рис. 10 приведен пример одной из возможных конфигураций такой среды, объединяющей систему распознавания речи, динамическую библиотеку программных модулей, базу данных и функциональные компоненты.

Термин среда программирования () в нашем случае определяет языки программирования, протоколы, технологии и инструменты. Нейронная среда, которую мы будем использовать при разработке различных приложений, включает в себя: *programming environment*

Среда нейронного программирования **Рис. 10.**

Практически все компоненты этой среды свободно доступны в Интернет и могут быть использованы при построении приложений на базе разных платформ – , или . Некоторые из них, такие, например, как или для распознавания и синтеза речи, зависят от операционной системы, и их применение возможно только под управлением . Технология Агентов, разработанная в Микрософт, позволяет использовать еще одну координату в структуре страниц. Агенты, существующие как бы вне плоскости документа (рис. 11), могут представлять автора и выполнять роль виртуального помощника для посетителей Веб-сайта. *Microsoft Windows Unix Mac OS Microsoft Agent Microsoft Speech Application SDK Microsoft Windows*

Трехмерный дизайн с использованием Microsoft Agent Рис. 11.

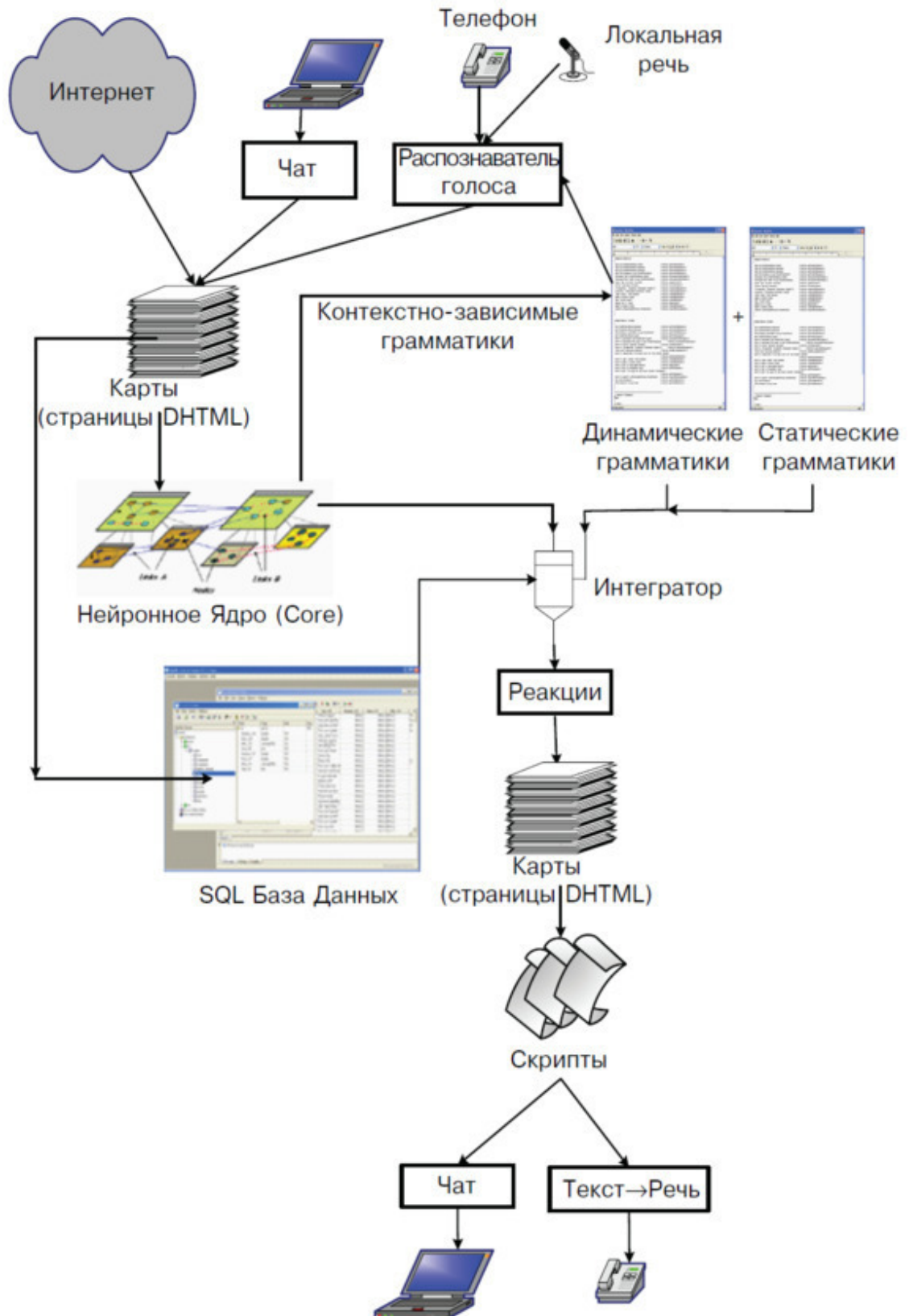
В отличие от индивидуальных систем связи, в основе которых лежат коммутации типа «точка-точка», персональные усилители способны одновременно соединяться со многими источниками данных, выделять из них определенную информацию и предоставлять ее пользователю в наиболее удобной форме. Такое решение является симметричным по отношению к виртуальным представителям в диспетчерских центрах (рис. 2) и может быть построено с применением все тех же базовых программных компонент. В этом случае виртуальный агент может выполнять роль индивидуального секретаря, динамически обучаясь и настраиваясь под знания конкретного пользователя и контекст его персонального компьютера.

В прикладном программировании широко применяются два способа разработки приложений – при помощи специализированных систем-оболочек и пакетов подпрограмм. В тех или иных вариациях на их основе разработано большинство современного прикладного программного обеспечения. В качестве примера интегрированной системы-оболочки можно назвать – одно из наиболее удачных решений в истории программирования. В специализированная система табличных вычислений интегрируется с пакетами подпрограмм разного уровня, что делает его практически идеальной прикладной средой, способной решать необычайно широкий класс задач. В можно выделить несколько последовательно усложняющихся уровней прикладного программирования: *Excel Excel Excel*

Если использовать аналогичную организацию уровней прикладного программирования в нейронном моделировании, их иерархия может выглядеть следующим образом:

- *интерфейсы к различным каналам ввода/вывода, по которым могут поступать речевые и текстовые сообщения;*
- *нейронное ядро (Core), в котором можно создавать искусственные нейроны вместе со связями (нервами), способными передавать возбуждения;*
- *база данных вместе с ODBC интерфейсом;*

- карты – объекты представления данных в виде динамических страниц;
 - скрипты – аналоги внешних реакций;
 - языковые оболочки Java и JavaScript;
 - XML-грамматики (динамические и статические);
 - интегратор – подсистема, способная исполнять динамические коды.
-
- создание рабочих таблиц;
 - добавление и модификация данных;
 - добавление и модификация формул;
 - добавление и модификация макрокоманд Visual Basic;
 - интеграция с другими системами через ODBC, XML, OLE и т. п.;
 - интеграция с другими системами через DLL, C++ и т. п.
-
- создание и обучение нейронных слоев;
 - программирование реакций на возбуждения, поступающие извне;
 - добавление нейронов и модификация их связей;
 - добавление и модификация макрокоманд Visual Basic или JavaScript;
 - интеграция с другими системами через ODBC и XML;
 - интеграция с другими системами через Java, DLL, C++ и т. п.



Взаимодействующие системы

Рассмотрим работу двух приложений – калькулятора и поисковой системы (рис. 12). И в первом и во втором случае, при нажатии одной из кнопок на панели управления первоначальный запрос, в виде цепочки символов, поступает на вход соответствующей программы, в результате чего порождаются множества процессов, связанных с исполнением детерминированных дискретных последовательностей команд. Наш практический опыт дает нам основание предполагать, что для всех одинаковых входных цепочек, поступающих на вход различных калькуляторов, независимо от времени и места, результат должен повторяться. В случае с поисковыми системами, все происходит с точностью до наоборот – мы ожидаем, что одна и та же входная цепочка символов на входе будет скорее всего приносить нам различные результаты в различных поисковых системах и в разное время.

Алгоритмический калькулятор и неалгоритмическая поисковая система **Рис. 12.**

И калькулятор, и поисковая система являются программами и, как любые программы, они состоят из точно заданных инструкций, которые как известно, должны всегда приводить к одинаковому результату при одинаковых исходных данных. Очевидно, что в случае поисковой системы новый результат получается каждый раз в результате изменений данных, которые в процессе исполнения длинной последовательности команд поступили на вход одного из программных модулей. Если бы мы захотели, используя, например, методы системного программирования при анализе дампов, мы могли бы, проследив все последовательности команд и событий, точно определить – где произошло это изменение. Однако значительно больше пользы нам может принести ответ на вопрос: существуют ли теоретические основания для различий в этих двух системах, и если да, можно ли построить более эффективную технологию и среду программирования для решения задач обработки данных в Интернет, учитывающую эти различия.

Взаимодействие системы и среды **Рис. 13.**

Предположим, что среда – это все доступные для восприятия и анализа объекты, процессы и события в окружающем нас мире, а система – часть среды, заключенная внутри некоторой границы (рис. 13). Человек, компьютер или организация являются примерами систем. Любая комбинация систем в свою очередь также может рассматриваться как система. $U S B$

В среде и внутри системы, могут происходить события E , которые определяются наборами параметров. Пусть S – множество, элементы которого мы будем называть событиями. Для каждого события из S мы поставим в соответствие действительное число P . События – это очень упрощенное представление об изменениях, происходящих в реальном мире. $U S P E e E P$

Мы будем предполагать, что любое изменение состояния среды или системы является результатом какого-нибудь события. Восприятие этих изменений в свою очередь, тоже является результатом событий. Так, например, зрительное восприятие есть возбуждение нейронов глазного дна или фоточувствительных элементов приемника, в результате отражения света от предмета, освещенного каким-либо источником; звуковые волны, передающие речь, которую мы можем слышать, являются результатом сокращения определенной группы мышц, в результате той или иной реакции человека на раздражение; электронные сообщения инициируются программой, запущенной в результате срабатывания механизма прерываний и т. д.

Для программистов, знакомых с Ассемблером и аппаратными архитектурами, системы обработки прерываний могут служить хорошим примером событийного программирования – в них данные поступают на обработку только после получения соответствующего прерывания (например, IRQ) об их готовности.

Любая система в реальном мире пронизана безмерным количеством всевозможных волновых и корпускулярных потоков излучений, которые постоянно проникают через ее границы

и несут в себе огромное количество информации. Человек способен использовать только очень малую часть сообщений, которые могут быть выделены из этих потоков. Если бы мы могли, выделив из всех проходящих через наше тело потоков сигналов, физиологически ощутить, как обычные звуковые и зрительные образы все телевизионные и радио каналы – наш мозг скорее всего переполнился и отказал в течение очень короткого времени. Учитывая ограниченные возможности мозга, наша способность улавливать только узкий диапазон электромагнитных волн является жизненно важным ограничением для человека, однако это же ограничение не позволяет нам без специальных устройств эффективно выделять полезную информацию из всех, окружающих нас потоков данных. Если в результате взаимодействия между системой и средой произошло изменение их состояния, мы будем называть такое взаимодействие – сообщением. Если изменение состояния системы, в результате получения сообщения, может быть измерено, мы будем называть такое взаимодействие информационным. Под информацией мы будем понимать определенным способом нормированную меру различия между состояниями системы до получения сообщения, и после. $M S M I \Delta S S^I$

Возможность измерять информацию является принципиальным свойством, позволяющим анализировать взаимодействия систем, и в некоторых случаях такое измерение может быть проведено в соответствии с определением информации, сформулированным Клодом Шенноном в работе «» [22]. Шеннон рассматривает модель коммуникационной системы, состоящей из пяти компонент: , , , и (рис. 14). *A Mathematical Theory of Communication Источник информации Передатчик Канал Приемник Получатель*

Схема коммуникационной системы по Шеннону **Рис. 14.**

Для того чтобы эту схему применить к современным информационным системам, нам понадобится внести в нее некоторые дополнения. Предположим, что информации – это набор страниц, размещенных на каком-либо сайте, – это Веб-сервер, а – персональный компьютер (рис. 15). В этой схеме используются все те же основные элементы коммуникационной системы Шеннона, только их количество увеличивается и, что является наиболее существенным, значительно увеличивается объем передаваемой информации. *Источник Передатчик Приемник*

Схема взаимодействия в Интернет **Рис. 15.**

Рассмотрим в качестве примера запрос в систему на поиск документов, в которых встречается комбинация слов – «». В результате мы получим список из ссылок на более чем 27 миллионов страниц! Если предположить, что искомая информация, которая представляет собой ответ на индивидуальный запрос, может содержаться в любой из этих страниц и считать, что объем средней страницы в Интернет составляет порядка 20 Кбайт, то суммарная длина сообщения которое все серверы готовы передать в канал для последующей обработки в персональный компьютер будет 5×10^{10} Байт. *Google information and energy L M, 14*

При скорости приемника 100 Мбит/сек понадобится больше года для того, чтобы один персональный компьютер сумел получить все страницы. Если, однако увеличить скорость передачи на порядок и использовать при получении этого потока не один, а 100 процессоров, время, необходимое для того, чтобы получить это сообщение, может быть сокращено до вполне приемлемого, однако совершенно очевидно, что человек при этом будет не в состоянии прочитать 27 миллионов страниц, с какой бы скоростью они не поступали на его письменный стол.

Шеннон рассматривал каналы с шумами, в которых элементом данных является символ, что можно сравнить с потоками индивидуальных молекул, перетекающих из одного сосуда в другой под действием некоторой силы. Такая модель позволяет совершенно точно определить физические характеристики каждой отдельной молекулы, но ничего не говорит о состоянии всего сосуда в целом. Для того чтобы говорить о температуре, необходимо перейти от молекул к объемам газа. Также и символьная теория информации – позволяет нам точно оценить передаваемые потоки данных на элементарном уровне, но не дает качественной картины в целом о сообщениях, состоящих из множества страниц.

Оптимальное соответствие между физиологическими ограничениями головного мозга и характеристиками выходных интерфейсов к информационным устройствам, предназначенным для индивидуальной фильтрации данных, может быть основано на скорости чтения, которая у людей колеблется от двухсот до пятисот слов в минуту, что соответствует примерно одной странице текста стандартного документа или приблизительно трем тысячам символов в минуту. Можно предположить, что информативность документов должна быть основана на иных критериях, и в первую очередь, она должна учитывать индивидуальные особенности получателя. Количество информации, содержащейся в документе в целом, и количество информации, содержащейся в символах этого документа, могут не совпадать и более того, обязательно будут отличаться для двух различных получателей.

Попробуем представить себе некий информационный измеритель, который может давать нам приближенные качественные характеристики состояний, подобных температуре физической системы. Такой гипотетический прибор мог бы ответить на вопрос, есть ли смысл человеку читать очередную страницу из списка, предоставленного поисковым сервисом и, в более общей форме, какие именно из всего множества страниц имеет смысл прочитать. Ответ на такой вопрос возможен, если мы сумеем ввести некоторую меру, которая позволит сравнивать индивидуальное человеческое и машинное представления об информации, содержащейся в сообщениях.

Согласование различных способов представлений в программировании является весьма деликатной задачей еще и по причине того, что круг пользователей, с которыми программистам приходится непосредственно соприкасаться при создании систем, необыкновенно широк. Терминология и определения могут принципиально отличаться даже в том случае, если речь идет об очень фундаментальных понятиях. Например, если в американском армейском терминологическом словаре сказано: «», – а большинство математиков, в свою очередь, считают, что «», то для программиста бессмысленно спорить и выяснять, кто прав – армейский устав или математическая теория коммуникаций. Важнее, следуя известному анекдоту, согласившись и с первым и со вторым определениями, выработать точку зрения, которая приведет к наиболее эффективному программному решению. По этой причине, в дальнейшем мы будем использовать некоторые компромиссные представления о понятиях и критериях, которые не всегда будут совпадать с традиционными для различных научных дисциплин. *Информация есть факты, данные или инструкции, записанные в любой форме и на любых носителях Информация есть мера снятой неопределенности об объекте*

Среди работ, лежащих в основе теоретических представлений об информации, необходимо особо выделить труды академика А. А. Колмогорова. В статье «» Колмогоров использует теорию алгоритмов применительно к определению «» [34]. Колмогоров предлагает оценивать сложность, а стало быть и информативность, объектов через минимальную длину программы, необходимой для получения u из x . Основываясь на этом подходе, можно понять причины проблем, возникших сегодня перед разработчиками поисковых систем, и затем найти пути их решения. *Три подхода к определению понятия «Количество информации» количества информации в чем-либо (x) о чем-либо (y)*

Действительно, современные поисковые системы достигли впечатляющих результатов в классификации -страниц. Однако, чем глубже и точнее проводится классификация информационного пространства, тем сложнее (а стало быть и длиннее) должен выглядеть запрос, в результате которого, пользователь может получить интересующие его данные. *Web*

Так, если поисковая система сумеет разделить страницы в Интернет на подмножества, которые соответствуют определенным критериям классификации (α , β , γ , δ , ...), то для того чтобы найти интересующие его страницы, пользователь должен знать список этих подмножеств и использовать его в явной или неявной форме при формулировке поискового запроса.

Как правило, интересующие нас страницы находятся на пересечении нескольких подмножеств, и логика запроса должна это отражать (рис. 16). ***A B C D E***

В явной форме запрос на поиск может выглядеть следующим образом:

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.