

Развитие человеко-компьютерного взаимодействия

В.Л. Авербух^{1,А,В}

^А ИММ УрО РАН

^В Уральский Федеральный Университет

¹ ORCID: 0000-0002-4379-1450, averbukh@imm.uran.ru

Аннотация

Работа посвящена обзору развития взаимодействия человека и компьютера. В первых разделах кратко описывается история вычислений в «докомпьютерную» эру, а затем ранняя история современного компьютеринга, методы управления первыми ЭВМ и задачи программистов на этом этапе. Рассказывается о методах взаимодействия с первыми ЭВМ посредством элементов пульта, перфокарт и перфолент. В разделе, посвященном ЭВМ второго поколения, рассказывается о появлении операционных систем и языков программирования высокого уровня. На этом же этапе появляются такие средства взаимодействия с ЭВМ, как дисплеи и, соответственно, такие средства программирования как диалоговые языки и диалоговые отладчики. Также начинаются исследования по принципам взаимодействия человека и ЭВМ. Рассматривается становление дисциплины «компьютерная графика», развитие пакетов компьютерной графики и появление стандартов интерактивной компьютерной графики. В разделе «Революции в вычислительной технике» в контексте взаимодействия человека и компьютера описываются появление большого количества ЭВМ одной серии, появление первых супер-ЭВМ. Рассматриваются революционные изменения в компьютерной графике и оформление дисциплины «компьютерная визуализация» с ее подобластями «научная визуализация», «визуализация программного обеспечения» и «информационная визуализация», а также «программирование путём демонстраций». Приводится информация о попытке создания ЭВМ пятого поколения на базе использования средств логического программирования. Рассказывается о начальном периоде обучения школьников программированию. Важными этапами становления современного компьютеринга стали создание вычислительных сетей и появление персонального компьютеринга, а также создание средств современных параллельных вычислений. Важным средством компьютерной визуализации становится виртуальная реальность. Современное состояние человеко-компьютерных интерфейсов характеризуется прежде всего появлением естественных интерфейсов, к которым можно отнести интерфейс мозг-компьютер (нейрокомпьютерный интерфейс, Brain-Computer Interfaces), интерфейсы на основе непосредственного использования нервных импульсов, интерфейсы, основанные на распознавании речи, интерфейсы, основанные на распознавании движения губ, интерфейсы, основанные на распознавании мимики, интерфейсы, основанные на распознавании перемещения взгляда (Eye Gaze или Eye Tracking), тактильные интерфейсы, а также интерфейсы, дающие тактильную обратную связь (позволяющие ощутить прикосновение), интерфейсы, основанные на фиксации движений (motion capture) всего тела человека или отдельных органов (головы, всей руки, кистей рук, пальцев, ног), интерфейсы на базе инструментальных средств захвата движений, в частности, интерфейсы на базе движений ног (foot-operated computer interfaces), жестовые интерфейсы, жестовые языки. Кратко описывается деятельностный подход к проектированию интерфейсов, а также некоторые проблемы массовых интерфейсов. В заключении обсуждается ряд проблем, возникающих в связи с ростом возможностей современных компьютеров. Работа носит характер научно-популярной статьи, она во многом отражает субъективные впечатления автора.

Ключевые слова: история взаимодействия человека и компьютера, компьютерная графика, компьютерная визуализация, вычислительные сети, персональный компьютеринг, естественные интерфейсы.

1. Введение

Эта работа посвящена обзору развития взаимодействия человека и компьютера. В ней описывается ранняя история вычислительной техники и методы управления первыми ЭВМ. Затем рассматривается появление интерактивных средств и средств компьютерной графики, возникновение систем компьютерной визуализации. В контексте взаимодействия человека и компьютера кратко описываются революции в компьютеринге, такие, как появление вычислительных сетей и персональных ЭВМ. Рассматриваются более современные средства взаимодействия, например, естественные интерфейсы. Также обсуждаются некоторые проблемы, возникающие в связи с ростом возможностей современных компьютеров. Работа носит характер научно-популярной статьи, она во многом отражает субъективные впечатления автора.

2. Вычислитель – кто он?

Истории взаимодействия человека и машины, также, как и истории вычислений, можно посвятить целые книги. Мы лишь очень кратко коснёмся этих проблем. Хотя и машины, и вычисления появились в глубокой древности, мы начнём с более свежих моментов, а именно, с эпохи индустриализации, когда появились современные методы управления техникой, а также возникли задачи сложных расчётов, связанных с проектированием машин различного назначения. Управляли машинами рабочие, а вот расчёты вели высококвалифицированные специалисты в области вычислений – учёные и инженеры. И хотя уже давно были изобретены такие вычислительные инструменты, как логарифмическая линейка и арифмометр, вычисления в области техники, банковского и страхового дела, как правило, велись вручную, для надёжности «в четыре руки». То есть расчёты проводились двумя специалистами и на каждом шаге сравнивались. Если результат не совпадал, то оба расчётчика пересчитывали последний этап. Очевидно, что эти вычисления были очень трудоёмкими и дорогостоящими.

Чтобы ускорить и удешевить решение задач, создавались новые вычислительные методы, с использованием которых можно было организовать вычислительные бюро с работниками меньшей квалификации. Интересно, что уже в XX веке в США было создано вычислительное бюро, где работали женщины, не имеющие специальной подготовки, которых и называли компьютерами (вычислителями). В это же время создавались такие вычислительные устройства, как электро-механические арифмометры и перфорационные машины. Перфорационные машины, выпускаемые компанией IBM, использовались при подсчёте результатов регулярных переписей населения в США. В нашей стране такая техника использовалась в машиносчётных станциях (машиносчётных бюро) для бухгалтерских и иных расчётов.

В тридцатые годы XX века по всему миру началось проектирование и разработка первых электронных счётных машин. Большую роль в их внедрении в практику сыграли атомные проекты в США и СССР. Однако расчёты первых атомных бомб велись в основном вручную и их проводили выдающиеся физики-теоретики и математики, специализирующиеся в области вычислительных методов. Для какого-то облегчения и ускорения расчётов придумывались «протопрограммистские» методы, аналогичные используемым затем в ЭВМ. Например, в США физики проводили вычисления по схеме, аналогичной вычислениям в современных ЭВМ. Схема выглядела примерно так: выделялся вычислительный блок, который обчислял один человек, затем следовали выделение и проверка некоторого значения и его анализ, проводимые другим человеком,

который передавал дальнейшие расчёты либо одному либо другому «вычислителю». В рамках советского атомного проекта использовались методы, аналогичные параллельным и распределённым вычислениям. Специально подготовленные девушки-вычислители работали на электро-механических вычислительных машинах. Вычисления разбивались на отдельные блоки, которые одновременно обсчитывались «в четыре руки». Затем отдельные результаты собирались в общий результат счёта задачи.

Естественно, что учёные, участники атомных проектов, обратили внимание на созданные электронно-вычислительные машины. Существуют примеры первых программ для первых ЭВМ, написанные такими крупными учёными как американский физик и математик Дж. фон Нейман и советский физик Я.Б. Зельдович. На рисунке 1 представлен фрагмент рукописи программы Джона фон Неймана [1].

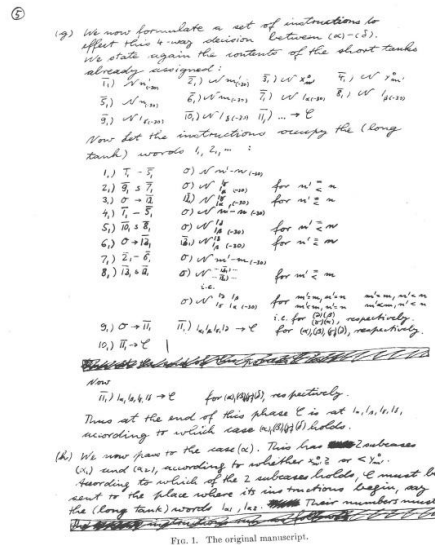


Рис. 1. Рукопись программы Джона фон Неймана [1].

3. Первые ЭВМ

В пятидесятых годах XX века началось создание ЭВМ. В США ведущим производителем компьютеров стала компания IBM. Однако IBM, как рассказывали, в конце сороковых годов считала выпуск компьютеров нерентабельным, так как, по мнению аналитиков, по всему миру нужно не более пяти таких машин – две в США, по одной Англии, Германии и СССР. Германия была разгромлена, а СССР стала противником США в Холодной войне, поэтому это число сокращалось до трёх. Наверное, на IBM повлияло американское правительство, чтобы компания занялась таким, как казалось, сомнительным делом.

В нашей стране первая электронно-вычислительная машина, МЭСМ, была сделана в Киеве под руководством будущего академика С.А. Лебедева. Рисунок 2 представляет коллектив создателей возле МЭСМ [2].



Рис. 2. Коллектив создателей возле МЭСМ [2].

К концу пятидесятих годов производство ЭВМ первого поколения (на базе электронных ламп) развернулось по всему миру. Были разработаны различные модели компьютеров с различными системами команд. Скорости этих машин составляли от нескольких сотен до тысяч операций в секунду. По нынешним масштабам память таких машин была очень маленькой. Программирование поначалу шло исключительно в кодах. Одна из первых советских машин, Стрела (Рис.3), была реализована на базе электронно-лучевых трубок (Рис.4), что было менее эффективно по сравнению с ламповыми машинами, но дало интересную возможность взаимодействовать с программистами.



Рис. 3. ЭВМ Стрела [3]



Рис. 4. Трубка электронно-лучевая ЛН-4 ЭВМ «Стрела» [4].

Компьютеры в это время использовались исключительно для научных вычислений, то есть для решения сложных задач, связанных с физикой, химией и инженерными расчётами. Математиками и физиками разрабатывались и/или использовались сложные методы вычислений. Именно эти специалисты и разрабатывали на базе этих методов новые программы.

В нашей стране в пятидесятых годах преподавание программирования началось на математических и физических факультетах университетов и других вузов. В Московском энергетическом институте (МЭИ) был создан факультет, готовивший специалистов в области ЭВМ. Сложилась новая профессия – программист. Эти специалисты, как правило, имели хорошую подготовку в области численных методов и занимались решением и компьютерной реализацией прикладных задач. В ряде технических вузов началась подготовка инженеров-электронщиков для работы с вычислительной техникой.

4. Взаимодействие с ЭВМ первого поколения

Теперь рассмотрим, как проходило взаимодействие программистов с ЭВМ первого поколения. Среди этих машин в нашей стране можно выделить замечательную машину М-20 со скоростью двадцать тысяч операций в секунду. Эта ЭВМ была разработана в ИТМиВТ АН СССР под руководством С.А. Лебедева.

Для работы на ЭВМ первого поколения выделялось «машинное время», то есть десятки минут или часы, во время которых программист был полноправным хозяином ЭВМ.

Программирование велось в машинных кодах. Программа записывалась с помощью номеров команд и ячеек. В пятидесятых годах программы уже вводились с помощью перфолент (Рис.5) и перфокарт (Рис.6), заимствованных из техники прошлых времён (например, телеграфа и перфорационных вычислительных устройств). Вывод на печать осуществлялся на сравнительно узкую (менее двадцати сантиметров) бумажную ленту. Хотя внутреннее представление чисел и команд было двоичным (но были и троичные машины, например, советская машина Сетунь), программист в основном работал с восьмеричными и десятичными числами. Интересно, что при обучении программированию новоявленные программисты пугались, что в дальнейшем им придётся и умножать, и делить числа в двоичной системе, но на следующем шаге оказалось, что можно обойтись привычными десятичными числами.

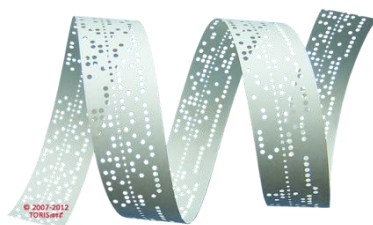


Рис. 5. Перфолента [5].

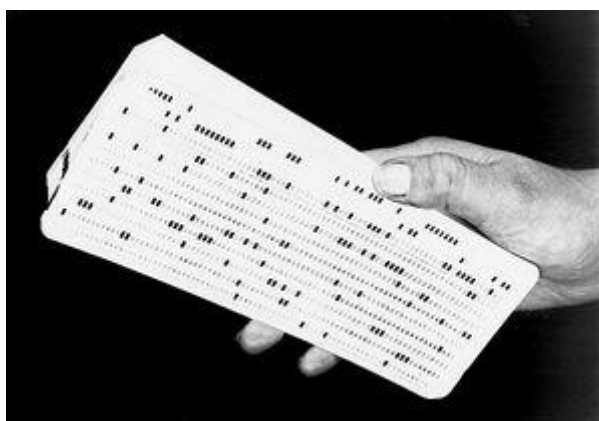


Рис. 6. Колода перфокарт [6].

Всё взаимодействие с ЭВМ проходило через пульт (Рис.7). На пульте был набор кнопок и тумблеров, а также индикаторы для отображения регистров и ячеек. Программист сам вводил программу и сам её запускал с помощью кнопок на пульте ЭВМ. Практически ни одна программа не запускалась с первого раза. Необходим был зачастую долгий и тяжёлый процесс отладки (debugging). При отладке программист в ключевых местах программы вставлял команды «останова» (stop) и мог увидеть, на какой из узлов было передано управление и каково состояние ячеек и регистров ЭВМ при выполнении этого участка программы. После отладки команды «останова» убирались.

Надёжность ЭВМ первого поколения была небольшая и машина выходила из строя чуть ли не каждый час. Так как работа велась днём и ночью, было необходимо организовывать круглосуточное дежурство инженеров-электронщиков и техников.

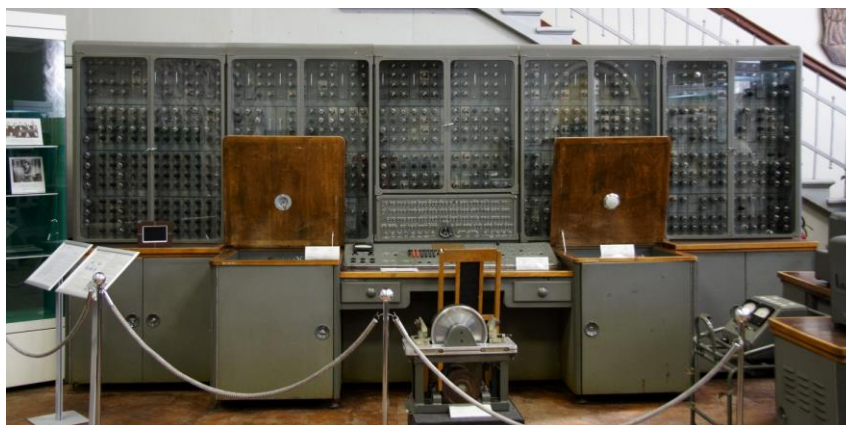


Рис. 7. ЭВМ Урал-1 [7].

5. Развитие вычислительной техники

Ситуация изменилась с созданием в шестидесятых годах ЭВМ на базе полупроводников (ЭВМ второго поколения), надёжность и производительность которых резко возросли. Например, советская БЭСМ-6 на 1965 год была самой быстрой ЭВМ в мире (Рис.8). Эта ЭВМ также была разработана в ИТМиВТ АН СССР под руководством С.А. Лебедева. Её скорость была миллион операций в секунду. Правда, оперативная память была по нынешним меркам малюсенькой, немногим более тридцати двух тысяч шестибайтовых ячеек. Эта машина бесперебойно работала практически круглые сутки с небольшим перерывом на профилактику. На машинах второго поколения активно использовались такие устройства внешней памяти как магнитные ленты и магнитные барабаны. Именно на машинах второго поколения были внедрены операционные системы и языки программирования высокого уровня, прежде всего ФОРТРАН и АЛГОЛ. Отметим, что в нашей стране для описания основных понятий использовался язык, отличный от западного и от того, что используется теперь. Например, говорили «ЭВМ» (точнее, «машина»), а не «компьютер», «автокод», а не «ассемблер», «транслятор», а не «компилятор».

Операционные системы (ОС) обеспечивали выполнение прикладных программ и их взаимодействие с внешними устройствами ввода и вывода, а также с внешней памятью – магнитными лентами, барабанами и дисками. Реализовывался пакетный, многозадачный режим работы ЭВМ, при котором операционная система обрабатывала пакет программ, на выполнение которых выделялся некоторый квант времени. При этом прикладной программист был полностью отстранён от общения с ЭВМ. Появились новые специальности – оператор ЭВМ, системный администратор и системный программист. Именно эти специалисты продолжали взаимодействовать с ЭВМ через всё более усложняющийся пульт. Разделение между прикладными и системными программами было достаточно жёстким. Например, в ранних инструкциях для БЭСМ-6 выделялось два режима выполнения команд: режим программиста (системного программиста с большими возможностями некоторых системных команд) и режим математика (прикладного программиста).



Рис. 8. ЭВМ БЭСМ-6 [8].

Кроме использования пульта, взаимодействие с операционной системой можно было организовать через алфавитно-цифровые дисплеи (Рис.10), а также в тех случаях, когда дисплеев практически не было (как в нашей стране), – через телетайпы (Рис.9), подключённые к ЭВМ. Оператор и «системщик» (системный администратор и системный программист, имевшие доступ к управлению операционной системой) отдавали команды ОС, связанные с управлением ходом общего счёта, и получали информацию от операционной системы о ходе выполнения отдельных задач.



Рис. 9. Телетайп [9].



Рис. 10. Алфавитно-цифровой дисплей [10].

Прикладной программист обычно передавал текст своей программы специальным сотрудникам (перфораторщицам), которые набивали текст на перфокартах. Затем программист собирал перфокарты в пакет и передавал его операторам ЭВМ, которые вводили программы через устройства ввода. Операционная система определяла, когда будет запущена программа, а результат счёта выдавался на «широкую печать», то есть на алфавитно-цифровое печатающее устройство (АЦПУ). У начинающих программистов большая часть запусков проходила впустую из-за многочисленных ошибок компиляции, связанных с ошибками при вводе перфокарт и других мелких ошибок новичков. Отладка программы занимала достаточно много времени.

Внедрение таких средств взаимодействия, как телетайпы и дисплеи, дали возможность появления диалоговых языков, таких как предшественник Basic'a – язык JOSS (отечественный вариант – Диалог БЭСМ-6). Работая с этим языком, программист вводил текст программы и запускал выполнение, имея возможность на каждом операторе проверить результаты вычислений. Началась разработка диалоговых отладчиков для автокодов (ассемблеров) и языков высокого уровня, таких как ФОРТРАН и АЛГОЛ. Отметим, что в шестидесятые и семидесятые годы были попытки создания русскоязычных автокодов (ассемблеров) и языков программирования. Например, русскоязычными были автокоды БЭСМ-6 – Автокод СОМИ и БЕМШ (тогда как автокод Madlen был англоязычным). Был создан русскоязычный вариант АЛГОЛа.

При отладке программы программист мог перейти в пошаговый (по операторный) режим выполнения или установить команду «останова» на некотором операторе и выдавать результаты вычислений на каждом шаге. Диалоговые отладчики резко сокра-

щали время отладки сложных программ. В дальнейшем широкое внедрение алфавитно-цифровых дисплеев и разработка соответствующего программного обеспечения позволили программисту ЭВМ с «пакетными» операционными системами вводить тексты программ без использования перфокарт и запускать программы со своего рабочего места.

Уже в первой половине семидесятых годов появились рекомендации по проектированию интерфейсов. В частности, указывалось, что важная информация на экране дисплея должна располагаться по диагонали, идущей с левого верхнего угла экрана, причем самая важная часть информации должна располагаться в центре экрана. Это соответствует привычному порядку чтения и написания текстов (слева направо и сверху вниз). Интересно, что для инокультурных пользователей, имеющих порядок чтения и написания текстов справа налево, рекомендовалось информацию располагать по диагонали, выходящей из правого верхнего угла. Также рекомендовалось проектировать диалоговые системы так, чтобы время ответа укладывалось в три секунды. Отметим, что в некоторых случаях пользователи не готовы ждать ответа даже одну секунду и начинают нервничать из-за ожидания. В других же случаях пользователи могут подождать несколько десятков секунд, особенно, если система отображает ход выполнения задания (например, процесс загрузки). После внедрения цветных дисплеев появились рекомендации по выбору цветовой гаммы выводимой информации, во многом аналогичные рекомендациям дизайна рекламы.

Изучение проблем взаимодействия с ЭВМ стало одним из направлений развития компьютеринга. Уже в начале восьмидесятых годов один из основателей отечественной компьютерной психологии А.Е. Войскунский описал важные аспекты диалога человека и компьютера [11].

6. Становление компьютерной графики

Теперь поговорим о становлении машинной (компьютерной) графики. Создателем первой системы компьютерной графики Sketchpad (Рис.11) является Айвэн Сазерленд (Ivan Sutherland). Отметим, что его научным руководителем являлся К. Шеннон, создатель теории информации. Однако по сведениям программистов «нулевого» поколения, инженеры самых первых машин начали использовать для вывода информации о результатах счёта приборы-самописцы и осциллографы. На ЭВМ Стрела, построенных на базе электронно-лучевых трубок, на трубки также можно было выводить изображение, дававшее информацию об устойчивости выполнения программы – если изображение было устойчивое, значит, с программой было всё в порядке. Когда что-то шло не так, изображение расплывалось или даже исчезало. (Это было прообразом визуализации программ, дисциплины, возникшей в восьмидесятых годах.)



Рис. 11. Айвен Сазерленд, работающий в системе Sketchpad [12].

Разрабатывались два типа графических устройств – устройства «твёрдой копии» (графопостроители, graphplotters) и устройства «мягкой копии» (графические дисплеи), по-английски hard copy и soft copy.

Первые графические дисплеи напоминали экраны радаров и, возможно, делались на их базе. Казалось естественным делать дисплеи на базе телевизионных приёмников, однако долгое время растровые дисплеи не могли использоваться, так как для хранения изображения было необходимо запоминать состояние всех пикселей (picture elements) экрана. Даже самый примитивный экран 128x128 строк, требовал хранения 16384 бит информации. Широко использовались векторные чёрно-белые дисплеи, в которых на экран передавались координаты точки начала вектора и сдвиги по координатам X и Y. Чтобы получить устойчивое изображение, было необходимо его непрерывно обновлять. Память дисплеев была сравнительно невелика и при большом количестве информации дисплей начинал мигать из-за того, что не хватало времени на обновление изображения. Из-за этого возможности по выводу изображений у таких дисплеев были невелики. Отобразить сложную трёхмерную картинку было практически невозможно.

В качестве устройств ввода использовалась алфавитно-цифровая клавиатура и телефонные диски. В дальнейшем, кроме алфавитно-цифровой и кнопочной клавиатуры использовалось такое устройство, как световое перо (Рис.13). Световое перо можно было использовать в трёх режимах – указание на графический объект (программа получала имя объекта; на базе этого можно было реализовывать «световые кнопки»), указание на точку на экране (программа получала координаты точки) и рисование (программа получала набор координат введённой линии). Графические дисплеи могли иметь собственную память, а могли использовать память компьютера, к которому были подключены. В последнем случае можно было легко реализовать анимацию двумерных изображений. Надо было переместить изображение на две дугие минуты (в случае небольшого экрана – приблизительно 2 мм) и сделать это чуть быстрее, чем за полсекунды.

Были внедрены различные устройства ввода, например, мышь (только называлась поначалу она жуком – bug) (Рис.14) и джойстик (joystick – палочка радости), трекболлы (trackball), тач-скрины (touch-screen – сенсорные экраны). Такие интерфейсы, реализуемые за счет операций с какими-либо устройствами, можно назвать девайсными интерфейсами (Device Interfaces).

В устройствах «твёрдой копии» (графопостроитель) (Рис.12) можно было выдавать разноцветные графики за счет переключения перьев (позднее – фломастеров). Цветные дисплеи в 70-х годах были слишком сложными и дорогими. В США, по некоторым данным, цветной дисплей стоил 100 000 долларов. В нашей стране цветные дисплеи выпускались в единичных экземплярах для специальных задач. Кстати, в середине 70-х годов зарубежный черно-белый дисплей (в неполной комплектации) стоил для советских научных институтов 10 000 золотых рублей, то есть более 11 000 тогдашних долларов.



Рис. 12. Графопостроитель. Первая половина семидесятых годов [13].



Рис. 13. Световое перо [14].



Рис. 14. Компьютерная мышь начала семидесятых [15].

Машинная (компьютерная) графика использовалась для визуализации результатов научного счёта и в системах автоматизации проектирования (САПР – по-английски Computer-aided design, CAD systems). Однако, вывод более-менее серьёзного чертежа на формат А1 мог потребовать нескольких часов работы графопостроителя.

Активно разрабатывалось программное обеспечение, прежде всего, пакеты машинной графики. В нашей стране в начале семидесятых годов был разработан ряд графических пакетов. Прежде всего необходимо упомянуть такие пакеты как СМОГ (ВЦ СО АН СССР, Новосибирск), [16] и ГРАФОР (ИПМ АН СССР, Москва), [17]. Разрабатывались также пакеты диалоговой (интерактивной) компьютерной графики, например, [18]. На Западе такие пакеты появились ещё в шестидесятых годах. В нашей стране разработка была проведена в середине семидесятых. Часто использовались зарубежные графопостроители и дисплеи, хотя разрабатывались и отечественные устройства.

Интерактивная компьютерная графика давала большой эффект в задачах компьютерного моделирования различных процессов. Можно было быстро просмотреть значительное количество кадров с графическим отображением различных элементов модели. В результате прикладной математик оказывался как бы внутри своей модели и мог наблюдать её изменения за счёт ввода параметров и взаимодействия с программой. Особенно важным было наличие графического ввода, то есть ввода координат точки или ввода кривой при помощи светового пера (в результате в программу поступало два массива с координатами введённых точек). Иногда работа с программой в диалоговом режиме и выводом графиков позволяла решить сложную задачу компьютерного моделирования за один сеанс, хотя до этого специалисты тратили на решение много недель, проводя сотни пусков.

Во второй половине семидесятых годов началась разработка стандартов интерактивной компьютерной графики. Был опубликован проект такого стандарта, разработанного в США – Core Graphics System. В нашей стране этот проект стандарта был реализован в нескольких организациях. Несколько позже в Германии был разработан другой проект – Graphical Kernel System (GKS), который и был принят в качестве международного стандарта. Несмотря на очень интересные идеи, заложенные в этих стандартах, они не дали ожидаемых результатов, так как уже в восьмидесятых годах началась

очередная революция в вычислительной технике и, соответственно, в компьютерной графике.

7. Революции в вычислительной технике

Вообще, развитие вычислительной техники представляет собой ряд последовательных революций, которые зачастую чуть ли не отменяют достижение предыдущих лет. Так, разработка сравнительно мощных и надёжных компьютеров второго поколения на базе полупроводников позволила внедрить операционные системы и языки программирования в практику. Развитое к тому времени искусство программирования в кодах и умение втиснуть серьёзную программу в ограниченные ресурсы ламповых ЭВМ оказалось излишним. В шестидесятых годах на полупроводниковой технике в IBM была разработана серия компьютеров IBM System/360 (Рис.15) с общей системой команд как для малых, так и для мощных (по тому времени) машин. У компьютеров этой серии было единое с точки зрения пользователей программное обеспечение (операционная система и система программирования). В дальнейшем уже после появления компьютеров третьего поколения на базе интегральных схем серия IBM System/360 была дополнена серией IBM System/370. Во многих странах мира были созданы аналоги IBM System/360. В нашей стране этот аналог назывался ЕС ЭВМ. Наличие компьютеров с едиными системами программирования позволило обеспечить лёгкое внедрение результатов, полученных в одной организации, в другую, даже находящуюся в другой стране. Количество компьютеров в восьмидесятых годах по всему миру составляло уже десятки тысяч экземпляров. Эти компьютеры широко использовались в коммерческих и научных вычислениях.



Рис. 15. ЭВМ IBM-360/30 [19].

Кроме компьютеров «средней» мощности, началась разработка супер-ЭВМ для задач научных вычислений. В частности, в США были разработаны суперкомпьютеры (по тогдашним меркам) Cray-1 (Рис.16) и Cray-2.



Рис. 16. ЭВМ Cray-1 [20].

Ещё в шестидесятых годах появились так называемые мини-ЭВМ, предназначенные для управления технологическими процессами и сложными объектами, например, морскими судами. Мини-ЭВМ также использовались в системах интерактивной машинной графики. Они подключались к «большим» ЭВМ, на которых осуществлялся основной счёт и формирование данных для визуализации. А мини-ЭВМ осуществляли непосредственный вывод на графические дисплеи и поддержку взаимодействия с пользователями за счёт ввода информации на устройствах ввода дисплеев [18].

Развитие авиационной и ракетной техники потребовало создания микро-ЭВМ, которую можно было использовать для управления самолётами и ракетами. Именно микро-ЭВМ стали основой очередной революции в мире компьютеров.

ЭВМ большой и средней мощности занимали машинные залы – большие, специально оборудованные, помещения сплошь заставленные металлическими шкафами с основными и периферийными устройствами, включая оперативную память, устройства внешней памяти (магнитные ленты, магнитные диски), устройства ввода и вывода, выпрямители тока и трансформаторы. Мини-ЭВМ были намного компактнее и были размером с один или два двусторчатых шкафа. Переход на интегральные схемы позволил уменьшить размеры компьютера, но все равно Сгау занимал обширное помещение.

В семидесятых – восьмидесятых годах началось развитие целого ряда направлений в вычислительной технике, среди которых были такие изменившие жизнь всех людей на Земле, как вычислительные сети, персональные вычисления, сотовая связь. Но сейчас мы рассмотрим изменения в компьютерной графике, произошедшие в восьмидесятых годах.

8. Изменения в компьютерной графике

Разработки в области вычислительных схем и памяти позволили создать надежные, компактные и недорогие цветные устройства вывода на базе растровой графики. Были разработаны графические станции – специализированные компьютеры, которые подключались к «большим» ЭВМ. В нашей стране была разработана серия графических станций Гамма (Новосибирск, Институт прикладной физики). С помощью одной из станций Гамма были визуализированы данные, полученные при исследовании планеты Венера. Графические платы Гамма-5 превращали алфавитно-цифровые дисплеи в графические.

Создание качественных графических устройств послужило предпосылкой для разработки алгоритмического обеспечения. В восьмидесятых годах американскими магистрантами и аспирантами были разработаны такие алгоритмы, как трассировка лучей (ray trassing) и излучательность (radiosity).

Трассировка лучей основывалась на старинной идее Аристотеля о том, что лучи из глаз человека падают на объекты. Идея, правда, используется, так сказать, в обратном варианте. Предполагается, испускание луча из объекта, расположенного на построенной алгоритмом на экране сцены (точнее, отдельного пикселя), на человеческий глаз или камеру, определяющую точку зрения. Если луч, исходящий из первого объекта, сталкивается с другим объектом, находящимся перед первым, то на экране отображаются элементы (пиксели) второго объекта и не отображаются соответствующие элементы (пиксели) первого. Отметим, что алгоритм считается недостаточно эффективным, но зато прекрасно распараллеливается.

Алгоритм излучательности основан на предположении, что все объекты на построенной алгоритмом сцене либо отражают либо поглощают свет. При программной реализации алгоритма решается соответствующее уравнение математической физики. Были разработаны и другие алгоритмы фотореалистической графики, послужившие основой современной компьютерной графики. Позже реализация алгоритмов была «защита» в специализированные графические процессоры (GPU), которые стали одной из основ современных вычислений.

Интересно, что некоторые разработчики алгоритмов фотореалистической графики приезжали на первые конференции Графикон [21] в Москву.

В настоящее время компьютерная графика стала не только научной дисциплиной, но и важной отраслью современной компьютерной индустрии.

9. Компьютерная визуализация

Развитие современной компьютерной графики послужило предпосылкой оформления компьютерной визуализации, как самостоятельной дисциплины.

Визуализация описывается как инструмент или метод интерпретации введенных в компьютер графических данных и генерации образов на основе сложных многомерных наборов данных. Очевидно, что визуализация, понимаемая как зримое представление ментальных моделей, существовала задолго до появления современной вычислительной техники. Более того, визуализацию, то есть перевод данных и информации в некоторые графические образы, можно рассматривать как составную часть нашей повседневной жизни. Первоначально, сразу после создания первых ЭВМ под визуализацией результатов счета понимался любой вывод цифр или символов на ленту примитивного печатающего устройства, лист АЦПУ или экран дисплея. Постепенно под визуализацией стал пониматься лишь графический вывод, например, рисование двумерных графиков или трехмерных поверхностей. Публикация доклада «Визуализация в научных вычислениях» в ноябрьском номере журнала ACM SIGGRAPH Computer Graphics за 1987 год положила начало новой эре в истории компьютерной визуализации. Отметим, что доклад был создан под эгидой национального научного фонда США.

Дадим сначала основные определения, а потом расскажем о истории развития дисциплины.

Под компьютерной визуализацией понимается методика перевода абстрактных представлений об объектах в геометрические образы, что дает возможность исследователю наблюдать результаты компьютерного моделирования явлений и процессов.

Традиционно выделяются следующие подобласти компьютерной визуализации:

- научная визуализация;
- визуализация программного обеспечения;
- информационная визуализация.

Под научной визуализацией понимается использование средств машинной графики и человеко-машинного взаимодействия для представления данных об объектах, процессах и явлениях, моделируемых при научных вычислениях.

Под визуализацией программного обеспечения понимается совокупность методик использования графики и средств человеко-машинного взаимодействия, применяемых для лучшего уяснения понятий и эффективной эксплуатации программного обеспечения, а также для спецификации и представления программных объектов в процессе создания программ.

Термин информационная визуализация относится к визуальному описанию и представлению абстрактной информации, получаемой в результате процесса сбора и обработки данных различного типа и назначения. Как правило, эти данные не имеют естественного и очевидного графического представления. Информационная визуализация объединяет методы научной визуализации и человеко-машинного взаимодействия. Методы информационной визуализации в значительной мере связаны с такими дисциплинами как получение новых знаний из баз данных (data mining или knowledge discovery) и визуальная аналитика (visual analytics).

Отметим также, что зачастую визуализация несколько упрощенно понимается лишь как непосредственное отображение трехмерных образов (рендеринг) на некоторую плоскость вывода или даже как простой набор визуальных и иконических интерактивных методов. Эти (кстати, весьма важные) вопросы следует скорее отнести к проблематике компьютерной графики и человеко-машинного взаимодействия.

Несмотря на различные области приложения визуализации, имеет место глубокое единство всех её подразделов как по методикам построения видов отображения (вплоть до методик рендеринга), так и по конечным целям и задачам – обеспечению интерпретации и анализа результатов компьютерного моделирования. Все это позволяет выделить компьютерную визуализацию как самостоятельную дисциплину со своим предметом и методом исследования.

10. Научная визуализация

Ясно, что использование графики для представления результатов научных вычислений началось с самого начала компьютерной эры. Разработки восьмидесятых годов в области аппаратного и программного обеспечения графики позволили американским исследователям быстро развернуть системы научной визуализации на новом уровне. Например, был опубликован спецвыпуск журнала *Computer* за август 1989 года, посвящённый научной визуализации и содержащий работы исследователей из NASA, лаборатории министерства военно-морских сил и других важных исследовательских центров. Качество визуализации было даже по нынешним меркам очень хорошее. Вскоре были опубликованы и другие спецвыпуски по этой тематике. История развития концепций и методов научной визуализации достаточно подробно представлена в работе [22].

В дальнейшем разработки в области научной визуализации реализовывались на мощных графических устройствах, включая среды виртуальной реальности, о которых будет рассказано ниже.

В настоящее время научной визуализации посвящается целая серия ежегодных международных конференций по всему миру. Типичный пример трёхмерной визуализации ядра человеческой клетки, реализованной в конце восьмидесятых годов, представлен на рисунке 17.

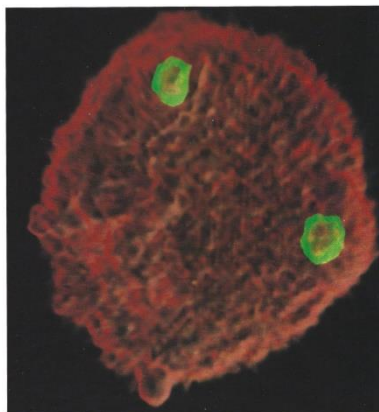


Рис. 17. Пример трёхмерной визуализации ядра человеческой клетки, конец восьмидесятых годов [23].

11. Визуализация программного обеспечения

В восьмидесятых годах издавалось большое количество научных журналов, посвященных компьютерной тематике. Материалы этих журналов были доступны отечественным исследователям либо через научные библиотеки, либо через реферативные журналы, в которых печатались рефераты практически всех научных статей. Полные тексты этих статей можно было получить в бумажном варианте или на микроплёнках. Анализ публикаций, относящихся к компьютерной графике, позволил выявить появление двух новых направлений – визуального программирования и визуализации программирования (*visual programming and program visualization*). Интересно, что сначала разделить эти понятия было сложно из-за близости названий.

Визуальное программирование подразумевает использование графики, в частности схем, диаграмм, иконических образов (иконов – icon) в процессе разработки программ. Визуализация программирования предполагает, что уже разработанные программы представляются в виде тех же графических элементов. В рамках визуализации программирования можно выделить такие разделы, как анимация алгоритмов, визуальная отладка и т. п.

Идеи визуальных языков программирования появились в семидесятых годах, когда системы компьютерной графики начали работать достаточно устойчиво. Визуальные языки семидесятых годов строились на базе блок-схем или диаграмм Насси – Шнейдермана (Nassi-Shneiderman-Diagramm), служивших для описания структурного программирования. К концу восьмидесятых годов это направление в достаточной мере развилось, разрабатывались диаграмматические и иконические языки программирования, создавались компиляторы визуальных языков, [24].

Первая система анимации алгоритмов была разработана еще в середине шестидесятых годов. Отдельные кадры были сняты на киноплёнку, и в результате получился фильм, описывающий работу алгоритма. В восьмидесятых годах системы анимации алгоритмов были реализованы на основе современной на тот момент компьютерной графики [25]. Была предложена интересная идея алгоритмических операций, которые служили основой «сценария» анимации [26].

Визуальные отладчики, в целом, основывались на идеях диалоговой отладки. Пользователь имел возможность просмотреть, например, трассу программы, увидеть графическое отображение интересующих его данных. Началась разработка систем визуальной отладки для параллельных вычислений. Для этого могла использоваться как естественная для отлаживаемого приложения графика, так и традиционные схемы и диаграммы.

Выдвигалась идея создания программных визуальных комплексов, заключающаяся в том, что и разработка программ, и их отладка, и их сопровождение должны осуществляться в рамках единой системы с одинаковым графическим представлением программных сущностей.

К концу восьмидесятых годов стало ясно, что оформилась новая дисциплина «Визуализация программного обеспечения». В начале девяностых годов появились первые публикации с ее описанием [27]. В нашей стране в 1995 г. было опубликовано учебное пособие по данной дисциплине [28]. Позднее монографии по визуализации программного обеспечения были изданы в США [29, 30]. Большое развитие визуализация программного обеспечения получила в связи с развитием супервычислений в девяностых годах. На рисунке 18 представлен пример трёхмерной визуализации архитектуры программного комплекса, выполненный в системе Vizz3D [30].

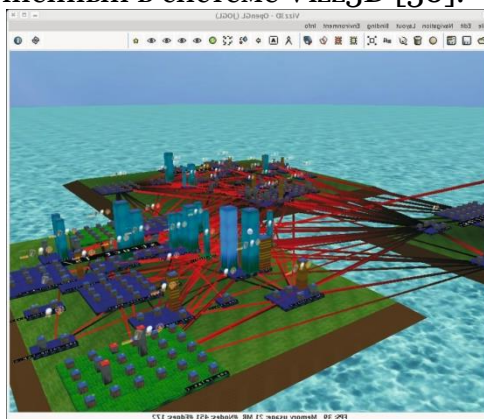


Рис. 18. Пример трёхмерной визуализации архитектуры программного комплекса, выполненный в системе Vizz3D [30].

12. Информационная визуализация

Информационная визуализация на первых порах базировалась на «докомпьютерной» статистической графике, то есть на графических методах представления статистической информации. Использовались различные типы графиков, схем и диаграмм, например, диаграмм Ганта (Рис.19) и Кивиата (Рис.20).

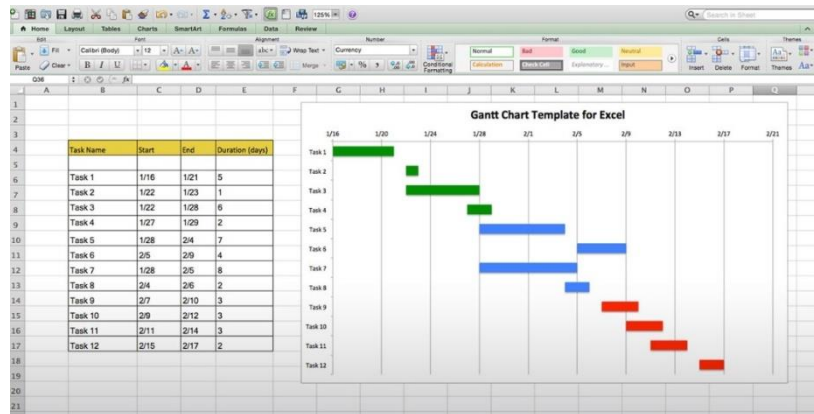


Рис. 19. Пример диаграммы Ганта [31].

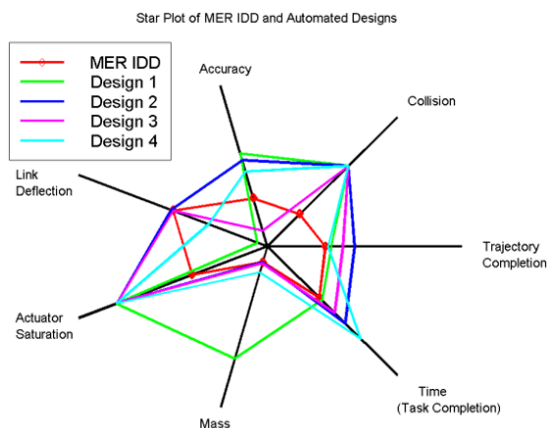


Рис. 20. Пример диаграммы Кивиата [32].

Даже простые методы визуализации могли дать большой эффект. Так, использование движущихся графиков, отображающих состояние больного (температура, состав крови, результаты анализов на инфекции и т.п.) и набора проведенных лечебных процедур (применение лекарственных средств, физиотерапия и т. п.) дало возможность специалисту быстро выявить эффективные процедуры для конкретных больных. Отметим, что вся исходная информация была в распоряжении медиков, но именно одновременное движение графиков помогло увидеть связь между состоянием больного и проводимым лечением (для определенного контингента больных).

В дальнейшем при необходимости визуализировать большие данные, например, данные об активности в социальных сетях, началось использование трехмерной графики, рисование сложных графов и т. п. Отметим, что методы информационной визуализации используются в системах отладки эффективности параллельных программ.

13. Программирование по образцам (программирование путём демонстраций)

К проблематике визуализации программного обеспечения примыкает раздел, популярный в восьмидесятых-девяностых годах, «Программирование по образцам».

Термин «Программирование по образцам», определяется для систем, которые позволяют программисту использовать образцы вводимой и выводимой информации в

течение процесса программирования. В программировании по образцам можно выделить два основных аспекта – формирование образцов (примеров) ввода и вывода и процесс логического вывода (или угадывания) программы по этим образцам.

В литературе также используются термины «программированием путём демонстрации», «визуальное обучение», «программирование путём репетиций» и другие. Хотя некоторые авторы противопоставляют эти термины, но мы будем использовать «программирование по образцам» и «программирование путём демонстраций» как эквивалентные.

Для многих визуальных аспектов программирования путём демонстраций нет эквивалента в текстовом последовательном программировании. Программирование путём демонстраций ведётся посредством манипуляций над данными на экране, демонстрирующими вычислительной системе, что программа должна делать. Преимущества такого подхода связаны с тем, что программисту легче выполнить нечто, чем описать это текстуально. Пользователь должен иметь возможность проинструктировать компьютер: «наблюдай за тем, что я делаю», и компьютер создаст программу, соответствующую действиям пользователя. Для многих систем программирование путём демонстраций заключается в непрерывном конструировании программы из трассы исполнения демонстрируемых пользователем примеров.

Принципы программирования путём демонстрации в значительной мере близки к методам обучения программированию, в которых значительную роль играет демонстрация готовых примеров правильных программ. Далее ученик обобщает полученный опыт для решения новых задач, аналогичных продемонстрированным задачам. Если демонстрация готовых образцов программ – могущественный метод обучения программиста-новичка, то программирование путём демонстраций – методика обучения вычислительной системы, методика демонстрации компьютеру образцов вводимой и выводимой необходимой пользователю информации, с тем чтобы на их базе были построены программы, взаимодействующие с пользователем.

Основная цель программирования путём демонстраций состоит в обеспечении конечного пользователя средствами создания готовых программ, при этом по возможности, не обременяя его необходимостью программировать (в обычном смысле) на языках типа Си, Паскаль или Пролог.

Основные аспекты программирования путём демонстраций – формирование образцов (примеров) ввода и вывода и процесс логического вывода (или угадывания) программы по этим образцам. Ясно, что существенной частью систем программирования путём демонстраций являются средства обеспечения ввода и вывода образцов необходимой информации. Естественно, что эти системы должны содержать знания о прикладных областях, в которых будут действовать полученные программы, а также набор правил логического вывода программ на основе знаний. Следовательно, охарактеризовать системы программирования путём демонстраций можно по следующим параметрам:

1. Область приложения и тип пользователей для программ, получаемых системой;
2. Методы взаимодействия с пользователями системы;
3. Набор правил логического вывода программ;
4. Знания об области приложения.

Системы программирования путём демонстраций прошли путь от простеньких систем начала семидесятых до достаточно изошрённых, снабжённых средствами искусственного интеллекта систем, разработанных в девяностых годах [33]. Многие из работ финансировались как правительственными ведомствами США и Канады (в основном военными), так и крупнейшими компаниями – IBM, Apple, Xerox, etc. Однако в дальнейшем интерес к этой тематике стал ослабевать. Возможно, это было связано с тем, что разработанные системы из-за бурного развития компьютерных технологий перестали быть актуальными. В то же время предложенные в рамках данного направления подходы могут использоваться, например, в системах обучения роботов. Представляет-

ся, что интересные идеи программирования путём демонстрации могут пригодиться в дальнейшем.

14. Пятое поколение ЭВМ

В восьмидесятых годах была предпринята еще одна попытка создать новое и важное направление развития компьютерных технологий. Была выдвинута идея ЭВМ пятого поколения. На тот момент были известны три поколения ЭВМ: на базе ламповых, полупроводниковых и ЭВМ на интегральных схемах. Четвертое поколение ЭВМ не рассматривалось. Пятое поколение предполагало прорыв в области вычислительной техники, программирования и человеко-компьютерного взаимодействия. Были выдвинуты идеи, связанные с искусственным интеллектом, сделаны попытки создания нового типа супер-ЭВМ и подходов к программированию, основанных на логическом выводе. Эти подходы к программированию должны были обеспечить принципиально новую работу с компьютером, который должен был понимать ввод на естественных языках, распознавание речи, перевод с языка на язык. Программирование компьютеров пятого поколения должно было вестись на языке Пролог. В рамках проекта активно развивались базы знаний и экспертные системы.

Базы знаний – это совокупность фактов и правил логического вывода в выбранной предметной области деятельности. Например, можно снабдить базу знаний о родственных связях таким правилом как «если X – отец Y, то Y – сын X». А затем задать информацию, что Сергей – отец Василия. Тогда система по запросу «найти сына Василия» сможет найти Сергея. В более сложных вариантах базы знаний, то есть базы данных, снабжённые предикатами, могут быть эффективным инструментом.

Экспертные системы должны включать в себя как знания специалиста, так и полученные от него правила логического вывода, актуальные для конкретной области.

Проект ЭВМ пятого поколения был предложен японскими исследователями и разработчиками. Казалось, что именно Япония сделает прорыв в будущее компьютеринга и программирования, обогнав весь мир. Однако, развитие вычислительной техники по целому ряду причин пошло иным путем. Свою роль сыграли многие факторы: и недостаточные мощности компьютеров восьмидесятых годов, и трудности логического программирования, и сложность реализации полноценного искусственного интеллекта, и, главное, та революция в области компьютеринга, которая связана с персональными компьютерами и интернетом. В то же время необходимо подчеркнуть важность разработки баз знаний и экспертных систем, которым сейчас уделяется недостаточное внимание.

Перед тем, как перейти к описанию появления персональных вычислений и вычислительных сетей, рассмотрим некоторые вопросы, связанные с начальным обучением программированию.

15. Программирование в школе

В нашей стране обучение школьников программированию началось в начале шестидесятых годов. Учащимся специализированных (математических) классов преподавали начальные сведения по системам счисления, математической логики, устройства ЭВМ, системам команд и методов вычислений. В ходе обучения проводилась летняя практика, в ходе которой школьники писали и отлаживали программы на реальных ЭВМ. Вплоть до середины семидесятых годов обучение программированию проводилось, в основном, в машинных кодах, так как многим казалось, что освоение языков программирования затруднит обучение. К этому времени такие языки, как Алгол и Фортран уже широко использовались в практике прикладного программирования, а теория алгоритмических языков и практика их применения привели к тому, что в мире были реализованы многие сотни языков и компиляторов. Отметим, что эти языки использовали английскую лексику, как базу для описания операторов и основных понятий. Правда, существовала точка зрения, что использование национальных языков в этом каче-

стве облегчит начальное обучение и освоение программирования. Кстати, был вариант реализации Алгола с полным переводом на русский язык.

В обучении школьников программированию в рамках школьного и внешкольного образования широко участвовали отечественные ученые. Следует отметить результаты исследователей под руководством академика А.П. Ершова из ВЦ СО АН СССР. Были разработаны такие языки начального обучения программированию на базе русского языка, как Робик и Рапира, а также система программирования для этих языков.

На стыке семидесятых-восемидесятых годов начали проводиться олимпиады по программированию для школьников. Однако из-за ограниченности доступа к загруженным реальными задачами ЭВМ олимпиады проходили «в сухую», и жюри проверяло только лишь написанные на бумаге тексты программ, а также решения логических задач.

Во второй половине восьмидесятых годов в нашей стране началось массовое обучение программированию учителей, преподавателей ВУЗов и, главное, всех школьников. В работе по компьютеризации образования активное участие принимали ученые АН СССР. Например, в Свердловской области этим процессом руководил академик Н.Н. Красовский. В это время за рубежом уже широко использовались персональные компьютеры, которые и были использованы, как база обучения. Было закуплено значительное количество персональных ЭВМ, которые разместили в учебных классах, в которые привозили старшекласников со всей области. В удаленные районы приезжали передвижные учебные классы, размещенные в автобусах. Преподавание программирования велось на языке BASIC.

В результате работы по компьютеризации стали проводиться всесоюзные олимпиады по программированию.

В девяностых годах предмет «информатика» вошел в программу средней школы. Преподавание программирования на базе персональных ЭВМ велось, как правило, с использованием языка Паскаль. Однако способные школьники самостоятельно осваивали новые компьютерные технологии и часто обгоняли своих учителей.

16. Персональные компьютеры

Кроме универсальных ЭВМ создавались и различные специализированные компьютеры, предназначенные для решения определенных задач. Например, еще в шестидесятых годах в нашей стране в ИК АН УССР под руководством академика В.М. Глушкова были разработаны специализированные ЭВМ серии МИР (машина инженерных расчетов), предназначенные для решения различных инженерных задач. Для ЭВМ МИР-2 в качестве входного языка был разработан язык высокого уровня АНАЛИТИК. Этот язык позволял на уровне своих команд непосредственно формулировать задания с аналитическими преобразованиями формул, непосредственно проводить операции дифференцирования и интегрирования. Для ввода и вывода информации на МИР-2 использовался дисплей.

Казалось, что появившаяся в семидесятых годах машина Apple II также является специализированным компьютером.



Рис. 21. Apple II [34].

Дело в том, что в отечественных программистских кругах прошли слухи о двух американских студентах Стиве Возняке и Стиве Джобсе. Рассказывали, что их выгнали из фирмы, в которой они подрабатывали, за компьютерные игры в рабочее время. В результате оба студента собрали в гараже специальный компьютер для игр, который и назвали Apple II. Дисплей этого компьютера работал в алфавитно-цифровом и графическом режимах, что было его преимуществом. Именно Apple II (Рис.21) положил начало современным персональным вычислениям, хотя ранее существовали компьютеры подобные этому по ряду своих параметров. Отметим, что на самом деле Apple II была универсальной и удобной в использовании ЭВМ. На ней реализовывались, например, серьезные программы визуализации, предназначенные для управления снабжением лекарствами больших больниц.

Позже пошли разговоры о том, что именно за персональными компьютерами будущее. Это уже казалось невероятным, ведь представлялось, что будущее за большими компьютерами, подобными Cray. Наверное, также рассуждали и руководители IBM, которые по слухам отказывались производить персональные компьютеры до тех пор, пока в зданиях корпорации не расположились офисы по продаже персональных компьютеров, и сотрудники не стали их покупать и использовать для работы. Кстати, позже многие были уверены, что фирма Microsoft является филиалом IBM.

Однако, именно персональные компьютеры стали основой нового компьютерного мира. Их пользователями стали не только и не столько программисты, но и исследователи различных категорий, инженеры, работники офисов. Хотя персональные ЭВМ выпускались во всем мире, но крупнейшими производителями в восьмидесятых – первой половине девяностых годов стали компании Apple (компьютеры линейки Macintosh) и IBM (компьютеры линейки IBM PC (Рис.22)).



Рис. 22. IBM PC [35].

Были произведены миллионы экземпляров компьютеров. Отметим основные качества персональных ЭВМ, позволившие им завоевать мир. Это компактность (ЭВМ можно установить на каждом рабочем месте), сравнительная дешевизна, надежность и простота использования (для обслуживания персональных компьютеров не нужен штат электронщиков, системных программистов и операторов), легкость освоения программного обеспечения, простота обновления программ и установки нового программного обеспечения, удобные средства взаимодействия с компьютером и с программным обеспечением.

Внедрение массовых персональных ЭВМ просто не могло состояться без появления визуальных средств взаимодействия, новых устройств и принципиально новых концепций организации интерфейса с пользователем, например, таких, как концепция “непосредственного действия” (direct manipulation).

Концепцию “непосредственного действия” предложил в начале восьмидесятых годов известный специалист в области вычислительных наук профессор Б. Шнейдерман (B. Shneiderman), собрав воедино и проанализировав новые тенденции в организации ин-

терфейса. В настоящее время эта концепция господствует при проектировании интерфейса [36].

Б. Шнейдерман определил следующие характеристики интерфейса, созданного на основе концепции “непосредственного действия”:

1) Постоянное отображение объекта, представляющего интерес;

2) Физические действия (работа с мышью, джойстиком, сенсорным экраном – тач скрином, etc.) или использование функциональной клавиатуры вместо команд со сложным синтаксисом;

3) Быстрые, разбиваемые на шаги, допускающие возврат операции, чьё воздействие на объект, представляющий интерес, немедленно становится видимым;

Суть этого подхода к созданию интерфейса состоит в создании у пользователя впечатления о том, что он непосредственно воздействует на представленные на экране объекты, а не ведет с ЭВМ диалог об этих объектах. Вместо использования командного языка для описания операций над объектами, пользователь манипулирует видимыми представлениями этих объектов на экране дисплея [36].

Использование персональных ЭВМ офисными работниками, прежде всего менеджерами различного уровня и их секретарями, послужило основой появления такого понятия как метафора интерфейса. Метафора интерфейса рассматривается как базовая идея сближения и аналогии между модельными объектами прикладной области и интерактивными объектами. Прежде всего метафора интерфейса начала использоваться в варианте метафоры рабочего стола (Desk Top metaphor).

Эта метафора объединяет объекты реального мира, находящиеся на поверхности стола офисного работника (например, папки с документами), а также иконическое представление программ и, главное, «волшебные» операции, такие как, двойной клик мышки, открывающий папки и запускающий программы. Интересно, что попытки расширить метафору рабочего стола, сделать стол трехмерным с ящиками и возможностью их открывания, а также разработать метафору офисной комнаты успехом не увенчались.

Возможно, что именно метафора рабочего стола стала основой для мощного продвижения персональных вычислений. Выпускались все новые версии персональных компьютеров, а со старыми обращались весьма своеобразно. Одно время в США проходили соревнования секретарш, в ходе которых девушки должны были на каблуках с системными блоками на руках пробежать на четвертый этаж и выбросить системный блок из открытого окна. Побеждала та девушка, системный блок которой падал первым. Были и более полезные идеи использования старых компьютеров – из их процессоров компоновали распределенные вычислительные системы с нулевой стоимостью.

В настоящее время ежегодно выпускаются миллионы и миллионы как традиционных персональных компьютеров, так и ноутбуков, планшетов и смартфонов.

17. Вычислительные сети

Считается, что основой современного интернета послужила сеть, разработанная в конце шестидесятых – начале семидесятых годов по заказу Агентства Министерства обороны США по перспективным исследованиям ([DARPA](#)) и предназначенная для управления военными объектами в случае ядерной войны. Следует отметить, что в начале семидесятых годов уже активно использовались сети, связывающие различные ЭВМ. В нашей стране академиком В.М. Глушковым предлагалось использовать вычислительные сети для эффективного управления народным хозяйством. Несколько позже началась работа над проектами глобальных (в масштабах всей страны) вычислительных сетей. Однако, в связи с разработкой сетей выдвигалась идея о том, что сети должны использоваться для перекачки задач с компьютера на компьютер примерно так, как перекачивается электроэнергия, то есть в то время, когда во Владивостоке уже ночь, и компьютеры не загружены, на них можно загрузить программы из перегру-

женных вычислительных центров Москвы. Это представлялось нелепым, так как в семидесятые – восьмидесятые годы компьютеры простаивали только во время профилактики или сбоев. Поэтому, хотя проекты таких сетей были частично реализованы, они не были внедрены в серьезную практику.

«Нормальные» вычислительные сети стали работать в нашей стране в конце восьмидесятых годов и были включены в мировой интернет. Постепенно был реализован повсеместный доступ к электронной почте. Уже в девяностых годах вместе с поисковыми системами появилась возможность искать, читать и скачивать научные публикации, размещенные на зарубежных сайтах. Это было особенно важно, так как буквально с сентября 1991 года в научные библиотеки перестали поступать зарубежные научные журналы, и исследователи в первой половине девяностых годов могли получить к ним доступ только через западных коллег.

Мощности вычислительных сетей были невелики. Так для того, чтобы скачать статью размером в 500–600 килобайт требовалось несколько часов, а иногда приходилось оставить скачивание на ночь. Срочная переброска файлов с визуализацией важных математических моделей из Екатеринбурга в Москву потребовала отключения на полчаса от интернета всех академических институтов и вузов города. Дело в том, что визуализация была получена буквально за час до открытия академической выставки в Москве. На открытии присутствовали руководители правительства России и было необходимо показать им новейшие результаты исследований института.

Позднее скорость и качество интернета возросли, появились социальные сети, онлайн компьютерные игры. Примерно в это же время стала распространяться сеть сотовой связи, основанная на иных принципах, но в дальнейшем, когда появились «продвинутые» (smart) телефоны, появилась и возможность выхода в интернет из сотовых телефонов. Пользователями интернета стали миллиарды людей со всего мира, имеющие доступ в сеть через персональные компьютеры, ноутбуки, планшеты и телефоны. В научной и ненаучной литературе появился термин «интернет-зависимость», который описывал тот факт, что многие пользователи проводят в сети чуть ли не всё своё свободное (и несвободное) время.

18. Параллельные вычисления

Уже в шестидесятых годах началось использование параллельных вычислений для различных приложений. Отметим, что в это время данная область не вызывала большого внимания у широких кругов разработчиков и пользователей. Однако ко второй половине восьмидесятых годов появились новые решения, которые сделали параллельные вычисления важнейшим направлением современного компьютеринга. Эффективными устройствами для организации параллельных вычислений стали транспьютеры, включавшие в себя центральный процессор и четыре канала связи для двустороннего обмена с другими устройствами. Создание параллельных вычислителей на базе транспьютеров подтолкнуло разработку средств параллельного программирования, например, языка программирования Оссам.

В середине девяностых годов в США был принят ряд программ, в рамках которых была обеспечена государственная поддержка создания супервычислений на базе параллельных вычислений. Прежде всего в разработке параллельных суперкомпьютеров приняли участие ядерные национальные лаборатории (Лос-Аламос, Ливермор, Sandia), исследовательские центры NASA, Министерства обороны. В этих исследовательских организациях параллельным супервычислениям уделяется очень большое внимание. Компьютерная база регулярно обновляется. Именно в этих центрах началось использование при разработке параллельных компьютеров графических процессоров (GPU) и процессоров игровых приставок.

Параллельное программирование в целом зависит от используемых параллельных архитектур. Так, для систем с распределенной памятью можно использовать MPI, про-

граммный инструментарий для обеспечения связи между ветвями параллельного приложения. MPI расшифровывается как Message passing interface (Взаимодействие через передачу сообщений). Для систем с общей памятью можно использовать библиотеку для параллельного программирования OpenMP.

В нашей стране в начале девяностых годов, несмотря на сложные условия, началась разработка параллельных вычислителей и соответствующего программного обеспечения. Так в первой половине девяностых годов академическими и отраслевыми институтами и организациями нашей страны был разработан параллельный компьютер МВС 100. Такие компьютеры работали в нескольких организациях. В дальнейшем была разработана серия суперкомпьютеров МВС, в частности, МВС 1000, установленный в Межведомственном Суперкомпьютерном Центре и доступный посредством удаленного доступа для широкого круга пользователей [37]. Суперкомпьютеры разрабатывались также в ряде других организаций страны. Для отечественных суперкомпьютеров было создано отечественное базовое программное обеспечение, в частности, операционная система, система управления файлами, средства машинной графики.

В настоящее время параллельные суперкомпьютеры массово используются по всему миру. Регулярно публикуется рейтинг самых мощных компьютеров мира. В нашей стране также созданы десятки суперкомпьютерных центров. Регулярно проводятся конференции, посвященные супервычислениям. На рисунке 23 показан российский суперкомпьютер «Ломоносов», расположенный в Московском государственном университете.



Рис. 23. Суперкомпьютер «Ломоносов» МГУ [38].

Использование столь мощной вычислительной техники приводит к проблемам в связи с затратами электроэнергии и воды на охлаждение. Рассказывают, что жители Калифорнии выходят на протестные акции при строительстве новых центров, опасаясь нехватки электричества и воды. Ведутся разработки новых методов охлаждения компьютеров.

Широкое применение параллельных суперкомпьютеров поставило новые задачи по визуализации получаемых данных для обеспечения анализа и интерпретации результатов. Огромный объем получаемых файлов и сложности используемых параллельных архитектур поставили задачи по организации вывода данных для систем компьютерной графики. В дальнейшем возникла необходимость в разработке средств удаленной и онлайн визуализации параллельных вычислений.

В связи с параллельными вычислениями с конца восьмидесятых годов проводились активные разработки средств визуализации программного обеспечения – языков визуального параллельного программирования, визуальных отладчиков правильности и отладчиков эффективности параллельных программ. Последние служат для того, чтобы предсказывать, находить и избегать возможную неэффективность исполнения параллельных программ.

Интересно, что уже в начале девяностых годов была разработана интересная система визуализации программного обеспечения Avatar [39], активно использующая средства виртуальной реальности и метафоры комнаты и здания в трехмерном варианте и функционирующая на базе среды виртуальной реальности типа CAVE. Система Avatar была предназначена для отладки производительности параллельных программ и позволяла представлять большие объемы данных о производительности параллельных процессов, полученных в ходе работы суперкомпьютера. По ходу работы пользователь как бы оказывается внутри трехмерного помещения, на стены которого проецируется видеоизображение. На пол и стены этого помещения выводятся кривые, описывающие метрики производительности параллельных программ в виде двумерных графиков. Вид отображения похож на стеклянный небоскреб, состоящий из комнат, каждая из которых содержит графический вывод, характеризующий различные аспекты поведения описываемой параллельной программы. Предусматривался режим прозрачности потолка и пола, что позволяло увидеть соседние данные в соседних «помещениях». Было определено визуальное отображение – «лента истории», для представления последовательности работы процессоров суперкомпьютера. «Виртуальный полет» по небоскребу на основе этой ленты давал возможность исследования совокупных данных о производительности прикладной параллельной программы.

Подробнее про виртуальную реальность будет рассказано в следующем разделе.

19. Системы виртуальной реальности

Виртуальная реальность – исторически устоявшийся термин, означающий создаваемую компьютером среду при помощи специальных устройств, таких как специальные шлемы (очки) Рис.24), экраны с иллюзией трехмерности, системы типа CAVE (Рис.25) (среды виртуальной реальности, в которых изображения проектируются на стены, пол и потолок помещения). Эти устройства позволяют создавать особую среду, воспринимаемую пользователем как реальный мир, в котором он действительно находится (но не наблюдает со стороны) и с которым взаимодействует напрямую, так же, как и с обычным миром.



Рис. 24. Современные очки виртуальной реальности, вид изнутри [40].



Рис. 25. Система виртуальной реальности CAVE-2 [41].

В литературе можно найти информацию о том, что первую систему виртуальной реальности разработал Айвен Сазерленд, один из основателей машинной графики. Однако представляется, что виртуальная реальность ведёт свою историю от авиационных тренажёров, в которых элементы моделирования полёта и даже воздушного боя выводились на большие экраны (в нашей стране) или на специальные шлемы (в США). Именно эти шлемы и послужили основой первых систем виртуальной реальности в восьмидесятих годах.

В конце восьмидесятих – начале девяностых годов среды виртуальной реальности использовались как для задач компьютерной визуализации, так в тренажёрах. Также началось использование этих сред для игр и развлечений. Для нас большой интерес представляет проект виртуальной аэродинамической трубы, реализованный в NASA для проектирования и испытаний космического корабля Шаттл. Так как провести испытания в обычной аэродинамической трубе для такого аппарата, как Шаттл, невозможно, была сделана полноценная модель, отображавшаяся в виртуальной среде. Исследователи могли менять параметры модели и видеть результаты [42]. Кстати, очень интересный доклад о виртуальной аэродинамической трубе для Шаттлов был сделан в 1994 году в Нижнем Новгороде на конференции Графикон ведущим разработчиком системы С. Брайсоном (S. Bryson).

Обобщением идеи виртуальной аэродинамической трубы является идея виртуального испытательного стенда для тех исследований, в которых натурный эксперимент либо сложно организовать, либо он невозможен. В этих интерактивных системах необходимо обеспечить возможность многократного запуска программы, моделирующей процессы с различными параметрами. В системах виртуального испытательного стенда предполагается проведение вычислительных экспериментов по моделированию таких сложных технических объектов, как космические корабли, ракеты или самолеты с визуализацией результатов средствами виртуальной реальности. Компьютерное моделирование, как правило, требует использования параллельных и распределенных вычислений. Возникает необходимость использования виртуальной реальности в режиме онлайн-визуализации [43, 44].

Виртуальная реальность активно используется в системах визуализации программного обеспечения. В этих системах часто используются метафоры визуализации, например, метафора города [45, 46]. Под метафорой визуализации понимается основная идея представления объектов, которая помогает восприятию визуальных образов.

Также в девяностых годах начались разработки средств дополненной реальности, накладывающие на реальные объекты сгенерированные компьютером изображения. Возникает возможность использования средств дополненной реальности для организации технологических процессов сборки сложных устройств. Специалист, который ведет сборку, может получать визуальные подсказки, что облегчает его работу и увеличивает её эффективность.

В настоящее время среды виртуальной реальности используются в развлекательных, образовательных, научных и терапевтических целях. Например, разрабатываются системы виртуальной реальности, позволяющие лечить фобии за счёт постепенного соприкосновения человека с пугающим объектом.

Важнейшим феноменом, связанным с виртуальной реальностью, является феномен присутствия (presence). Под присутствием понимается ощущение нахождения в виртуальной среде так, как если бы она была настоящей. Это переживание описывается как «быть там» [47].

Уже в девяностых годах появились предсказания того, что человечество скоро переселится в виртуальный мир. При этом часто путают среды виртуальной реальности и виртуальный мир интернета, в котором «сидят» миллионы пользователей. Эти предсказания активно распространяются по сей день. Представляется, что авторы до конца не понимают сущности виртуальной реальности как инструмента человеко-компьютерного взаимодействия.

Отметим также некоторые ограничения при использовании виртуальной реальности, связанные с возможными неприятными и даже болезненными ощущениями, которые могут быть вызваны у пользователя. Это может быть связано с проявлениями *киберболезни* (*cybersickness* – расстройства здоровья, сходного с морской болезнью). Киберболезнь часто связывается с отсутствием возможности активного управления событиями виртуальной реальности [47].

20. Естественные интерфейсы

Существует несколько определений естественных интерфейсов. В одних случаях предполагается, что пользовательские операции интуитивно понятны и основаны на естественном бытовом поведении. В других говорится о базирующемся на естественных элементах фактически незаметном интерфейсе (или становящимся таковым после его освоения пользователем). В данной статье под естественными интерфейсами понимаются интерфейсы, построенные на фиксации и распознавании какой-либо комбинации движений человека или активности его органов.

Естественные интерфейсы обеспечивают взаимодействие «с головы до пят». В частности, можно выделить следующие интерфейсы:

- Интерфейс мозг-компьютер (нейрокомпьютерный интерфейс, Brain-Computer Interfaces);
- Интерфейсы на основе непосредственного использования нервных импульсов;
- Интерфейсы, основанные на распознавании речи;
- Интерфейсы, основанные на распознавании движения губ;
- Интерфейсы, основанные на распознавании мимики;
- Интерфейсы, основанные на распознавании перемещения взгляда (Eye Gaze или Eye Tracking);
- Тактильные интерфейсы, а также интерфейсы, дающие тактильную обратную связь (позволяющие ощутить прикосновение);
- Интерфейсы, основанные на фиксации движений (motion capture) всего тела человека или отдельных органов (голова, всей руки, кистей рук, пальцев, ног);
- Интерфейсы на базе инструментальных средств захвата движений, в частности, интерфейсы на базе движений ног (foot-operated computer interfaces);
- Жестовые интерфейсы, жестовые языки.

При этом необходимо учитывать возможность комбинации в рамках одной реализации фиксации нескольких естественных активностей человека.

Нейрокомпьютерные интерфейсы (brain-computer interfaces) основаны на распознавании собственной электрической активности мозга, связанной, в частности, с движениями рук и ног и формировании команд на перемещение реальных или виртуальных объектов (Рис.26). Это направление активно развивается последние десятилетия, как за рубежом, так и в нашей стране. Интересны результаты по использованию интерфейсов данного типа в системах виртуальной и расширенной реальности как для манипуляции объектами, так и для осуществления навигации в виртуальном пространстве.



Рис. 26. Brain-computer interface [48].

Интерфейсы на основе непосредственного использования нервных импульсов могут использоваться при разработке протезов нового типа (Рис.27). Также известны примеры протезов рук, построенных на базе нейрокомпьютерных интерфейсов.



Рис. 27. Кадр видео манипуляции протезом руки, управляемого посредством нейроинтерфейса [49].

Распознавание речи – одно из самых популярных приложений идей искусственного интеллекта. Именно в данном направлении получены реальные результаты. Существующие приложения уже используются в бытовых условиях (например, запросы к мобильным устройствам, управление лифтами и пр.) Интерфейсы, основанные на распознавании движения губ можно рассматривать в качестве вспомогательных, позволяющих повысить точность систем распознавания речи (Рис.28).

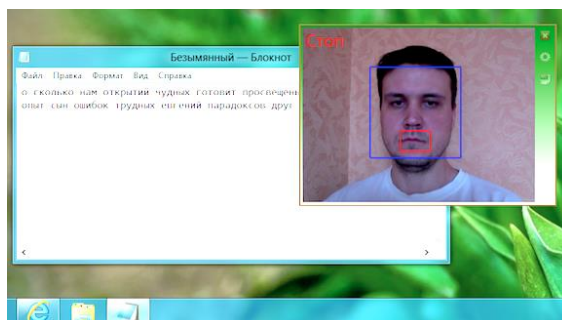


Рис. 28. Интерфейс на базе распознавания речи [50].

Распознавание направления взгляда и мимики человека может использоваться в целях организации человеко-компьютерного взаимодействия в системах визуализации (Рис.29).



Рис. 29. Интерфейс на базе отслеживания взгляда [51].

На шлемах, использовавшихся в авиационных тренажерах, послуживших основой для систем виртуальной реальности, изображение подстраивалось в зависимости от направления взгляда пилота. Интерфейсы, основанные на распознавании мимики и направления взгляда, также разрабатываются для обеспечения связью людей, поте-

рявших способность двигаться и даже говорить. В последнее время популярным стало использование интерфейсов на основе распознавания направления взгляда в играх. Часто такие интерфейсы применяются в комплексе с другими типами естественных интерфейсов.

Тактильные интерфейсы могут быть интересны в связи с созданием жестовых интерфейсов и обеспечением обратной связи при работе в средах виртуальной реальности и с «большими» экранами (Рис.30,31).



Рис. 30. Пример тактильного интерфейса управления автомобилем [52].

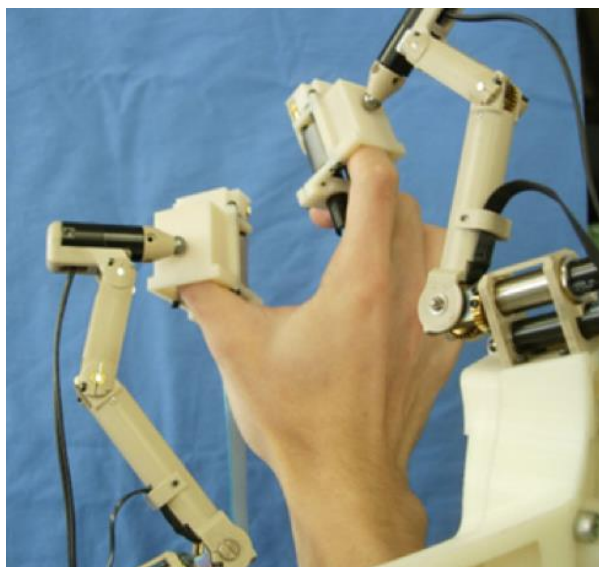


Рис. 31. Многопальцевый тактильный интерфейс: устройство, позволяющее имитировать тактильные ощущения прикосновения к мягким предметам [53].

При организации перемещения в виртуальном пространстве широко используются естественные интерфейсы, основанные на фиксации и распознавании движений всего тела человека или отдельных органов.

В первый период развития сред виртуальной реальности использовали специальные костюмы, фиксирующие движения ног. Активно использовались специальные панели и платформы, шаги и перемещения по которым связывались с перемещениями в виртуальном пространстве. Реальная ходьба предлагалась как способ организации перемещений в виртуальной среде. С другой стороны, возможно, что перемещение в абстрактных виртуальных пространствах проще организовывать за счет виртуального полета. (Напомним, что перемещение в виртуальном пространстве, которое неподконтрольно пользователю, может вызвать у него неприятные ощущения, описываемые по-

нятием киберболезнь.) Разрабатывались естественные интерфейсы на базе использования движений ног (Рис.32,33). При этом руки остаются свободными.



Рис. 32. Пример «ножного» интерфейса [54].

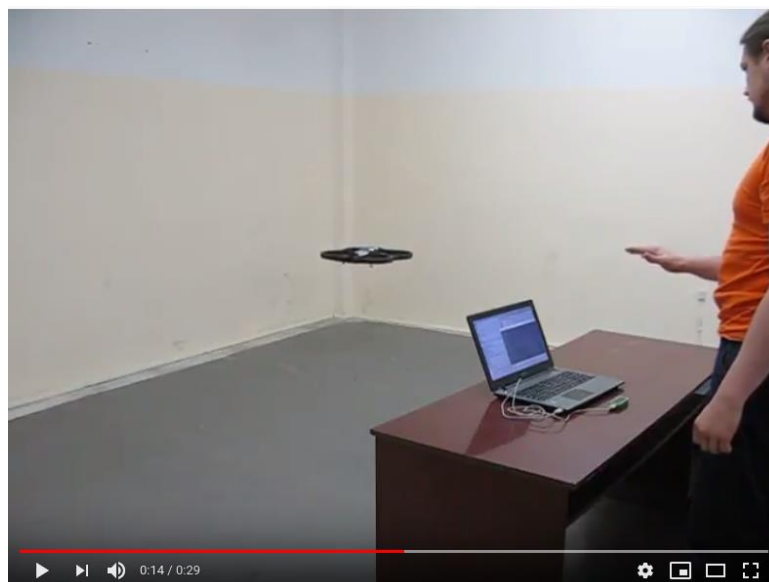


Рис. 33. Пример «ножного» интерфейса (другой ракурс) [54].

Также разрабатывались мультимодальные естественные интерфейсы, в которых используется сразу несколько методик человеко-компьютерного взаимодействия – жесты рук, движения ног, фиксация взгляда, тактильные интерфейсы.

Жестовые интерфейсы можно разделить на два типа – статичные, когда предъявляется некоторый набор зафиксированных положений рук, и динамичные, когда система воспринимает и анализирует движения рук или других частей тела. Анализ производится на базе захвата движения, который основывается на методах, используемых в теории обработки изображений, и относятся к области технического зрения. Жестовые интерфейсы могут использоваться в медицинских приложениях, например, для обеспечения безопасности врачей при рентгеновском исследовании пациента. Также жестовые интерфейсы используются для управления в средах виртуальной и дополненной реальности (Рис.34).

При использовании жестовых интерфейсов важно разработать жестовые языки, помогающие пользователям в их деятельности [55].



VirtualHand. Управление квадрокоптером с помощью жестов

Рис. 34. Кадр видео управления квадрокоптером [56].

21. Деятельностный подход к проектированию интерфейсов

При проектировании интерфейсов необходим серьезный анализ аспектов **деятельности** будущих пользователей. На этапе «сопровождения» и доработки системы необходимо уяснить, как изменилась деятельность пользователей после компьютеризации их работы. Теория деятельности, разработанная в нашей стране в середине XX века, связана, прежде всего, с именами А.Н. Леонтьева [57] и С.Л. Рубинштейна [58]. Публикации, в которых предлагалось использование теории деятельности при проектировании и разработке человеко-компьютерных интерфейсов, стали появляться во второй половине восьмидесятых – начале девяностых годов прошлого века. В качестве первопроходцев этого направления следует отметить видного отечественного ученого В.П. Зинченко [59, 60].

Деятельностный подход к проектированию интерфейсов подразумевает серьезный анализ аспектов деятельности будущих пользователей интерфейсов. На этапе «сопровождения» и доработки системы взаимодействия необходимо уяснить, как изменилась деятельность пользователей после компьютеризации их работы.

В связи с деятельностным подходом рассматриваются инструментальные интерфейсы. Под инструментальными интерфейсами понимаются интерфейсы для специалистов в какой-либо области, использующих их как средство для осуществления своей профессиональной деятельности, а также массовые интерфейсы-инструменты общего назначения, например, для систем бронирования и покупки билетов, использования банковских, социальных и государственных услуг и пр. Для таких интерфейсов мало применимы подходы к проектированию и критерии качества, применяемые для развлекательных сайтов или в социальных сетях.

Деятельностный подход к проектированию человеко-компьютерного взаимодействия для конкретной проблемы предполагает глубокое изучение работы будущих пользователей в «докомпьютерном» варианте, анализ всех возникающих задач и описание деятельности по их решению. Важно выявление основных целей и мотивов данной деятельности, описание отдельных этапов деятельности и выявление всех сущностей, с которыми работники имеют дело. Также необходим «деятельностный» анализ новой ситуации, возникающий после компьютеризации работы.

Для инструментальных профессиональных интерфейсов критерий качества может быть оценен через количество людей, удовлетворенных работой учреждения в течение

заданного отрезка времени. То есть мы рассматриваем количество клиентов, покупателей, пациентов и пр., получивших удовлетворяющий их результат и не получивших серьезного стресса. Измерить стресс можно как прямыми методами у пользователей профессиональных интерфейсов, так прямыми и косвенными методами у пользователей массовых интерфейсов и посетителей учреждений. Можно предполагать также, что уровень стресса у профессионала, использующего данный интерфейс, влияет на уровень стресса посетителя – из-за возможных задержек, сбоев в работе и общего раздражения. Практика реальной компьютеризации дает нам примеры удачных и неудачных результатов [61].

Как представляется, у деятельностного подхода к проектированию интерфейсов очень большие перспективы. Такие интерфейсы могут стать основой разработки современных удобных и надежных сред человеко-компьютерного взаимодействия в важнейших областях жизни и деятельности людей.

22. Массовые интерфейсы

Под массовыми интерфейсами будут пониматься массовые инструментальные интерфейсы, интерфейсы развлекательных программ, интерфейсы социальных сетей, игровые интерфейсы, а также интерфейсы мобильных устройств.

К классу массовых инструментальных интерфейсов отнесем интерфейсы предназначенные, в частности, для систем бронирования и покупки билетов, пользования медицинскими, банковскими, социальными и государственными услугами и пр.

В случае массовых инструментальных интерфейсов проектировщик, формулируя требования к интерфейсу, участвует в формировании будущей деятельности. Пользователь не может отказаться от использования соответствующей системы, так как через нее он получает доступ к важным для своей жизни услугам, ресурсам, информации и т.п. Массовый инструментальный интерфейс должен ориентироваться на «слабое» звено, то есть с ним должен успешно справляться человек с минимальными возможностями по вводу, восприятию и анализу информации [61].

Однако (как уже указывалось выше) методики проектирования и критерии оценки, применяемые для «интерфейсов потребления», не пригодны в случаях «инструментальных» интерфейсов. «Развлекательные» интерфейсы обычно не связаны с целенаправленной деятельностью. Их разработчики ставят перед собой задачи привлечь внимание пользователей к каким-либо стимулам для достижения нужной заказчиком реакции. Однако деятельностный подход применим и к проектированию информационно-рекламных сайтов, хотя их посещение пользователем и не является деятельностью. Пользователь выступает как объект деятельности владельца (заполнителя, заказчика) сайта. Именно для владельца мы можем выстроить и цель, и мотивацию и определить действия и операции. Последние, впрочем, возможно, будут определяться не очень четко, так как деятельность, направленная на человека, не поддается точному программированию.

Во многих случаях от пользователей современных интерфейсов, так или иначе, требуется деятельность по программированию. Отметим тот факт, что проектировщиками четко не описаны (а возможно даже ими и не осознаются) и правила и «язык программирования», и само «виртуальное устройство», которое необходимо «программировать».

Следующий, уже современный этап развития интерфейсов связан с интернетом. Важным источником идей разработки интерфейсов стали развлекательные сайты и сайты электронной торговли и электронного предоставления услуг. Оценки качества интерфейсов также во многом стали связываться с оценками эффективности электронной торговли и с эффективностью размещенной на сайтах рекламы. Эффективность последней может оцениваться, прежде всего, по количеству «кликов» на рекламный баннер. Представляется, что с этим связана немаловажная причина господства

при оценке качества интерфейсов (бихевиористской по своей сути) модели «стимул-реакция».

Оценка массовых интерфейсов может вестись по критериям usability. В этом случае используются субъективные методики оценки качества интерфейсов и сайтов на базе анкетирования небольшого количества пользователей, которые совмещаются с анализом качества дизайна и его эргономичности. К этому добавляются инструментальные методики отслеживания перемещения взгляда по экрану (eye tracking).

Традиционно разработчики много говорят о дружелюбных и интуитивно понятных интерфейсах. Система имеет интуитивно понятный интерфейс (интуитивно пригодна к использованию или пригодна к использованию на интуитивном уровне), если неосознанное применение пользователем уже имеющихся у него знаний приводит к эффективному взаимодействию с ней [62].

Следует обратить особое внимание на интерфейсы компьютерных игр. Такие игры появились ещё в семидесятых годах и поначалу пользовались текстовой подачей информации. Затем начали активно развиваться игры, использующие компьютерную графику. Правда, графика была достаточно примитивной, и игроки не обращали на это внимания, например, на трёхпиксельных человечков на заднем плане просто потому, что сосредотачивались на том, что происходило на переднем. Уже в восьмидесятых годах в компьютерные игры играли уже миллионы людей по всему миру. Интерфейсы игр оказали влияние на разработчиков систем визуализации, которые заимствовали отсюда ряд идей (метафор визуализации). Современные игры начали использовать средства виртуальной реальности. Одно время утверждалось, что такие игры вытеснят все остальные, но этого не произошло до сих пор и, наверное, не произойдёт никогда. Отметим, что в игровых интерфейсах могут использоваться устройства ввода, как геймпады и джойстики, а также клавиатуры, специально разработанные для «геймеров». Специалисты, связанные с разработкой игр, считают, что хороший игровой интерфейс – это такой интерфейс, который игрок не замечает. Правда, иногда сложности интерфейсов могут входить в состав геймплея (компонента игры, отвечающего за взаимодействие игры и игрока). Игры на мобильных устройствах (как и вообще использование мобильных устройств) также оказывают большое влияние на пользователей и, следовательно, влияют на разработку массовых интерфейсов. Привычные пользователям с детства приёмы организации взаимодействия должны учитываться при создании как массовых, так и специализированных интерфейсов.

23. Заключение

История современного компьютеринга насчитывает почти 80 лет. Постоянно возникают новые компьютерные технологии, например, облачные вычисления, дающие доступ к конфигурируемым вычислительным ресурсам, или 3D-принтинг, позволяющий создавать готовые детали сложных машин, а также «печатать» на клеточной основе фрагменты органов и тканей человека. Сейчас у всех на слуху квантовые вычисления, использующие квантово-механические явления для выполнения вычислений. В связи с развитием современных вычислений стала востребованной задача обработки и анализа «больших данных» (big data). Огромную популярность приобрели современные подходы к созданию искусственного интеллекта на базе нейронных сетей.

По-прежнему актуален вопрос о взаимодействии человека и компьютера. Это связано с тем, что если в первые два десятилетия его развития ЭВМ использовали тысячи программистов по всему миру, то сегодня в том или ином виде их используют миллиарды людей, среди которых миллионы специалистов, включая программистов различного уровня и IT-специалистов, ученых, медиков, инженеров, преподавателей, банковских служащих, офисных работников и др.

Одним из подходов к обеспечению более эффективного человеко-компьютерного взаимодействия является использование естественных языков. Причем в шестидеся-

тых-семидесятих годах даже велись разработки средств программирования на базе естественных языков. Однако, ввиду большого объема и неоднозначности словарей естественных языков и их сложного синтаксиса очень сложно обеспечить точное распознавание естественных текстов.

В 1978 году академик А.П. Ершов предложил использовать подмножество естественного русского языка – «канцелярский русский», язык отчетов, заявлений и анкет в качестве основы языка взаимодействия с ЭВМ. В первом докладе на эту тему Андрей Петрович употреблял придуманное К.И. Чуковским слово «канцелярит», а затем использовал более научный термин «язык деловой прозы». Этот язык характеризуется относительно небольшим словарем (около трех тысяч слов), достаточно точно определенными синтаксисом и семантикой. То есть у слов и понятий «канцелярита» значения точно определены, а фразы отчетов и заявлений пишутся по установленной схеме. Язык деловой прозы естественен для многих специалистов в сфере управления, и в этой сфере возможно достаточно эффективное взаимодействие человека с компьютером на его базе. При этом реализация распознавания текстов с ограниченным словарем и четко определенными синтаксисом и семантикой значительно упрощается [63, 64]. Работы в этом направлении велись вплоть до девяностых годов. Однако, развитие современных информационных технологий в области управления бизнеса понизило актуальность этих разработок.

Вместе с тем естественное человеко-компьютерное взаимодействие остается важным элементом современного компьютеринга. Например, большое внимание сейчас уделяется голосовому взаимодействию с компьютером с использованием элементов естественного языка.

Представляется, что естественность взаимодействия надо увязывать с основной деятельностью будущих пользователей, учитывая их опыт в использовании информационных технологий. Например, для инженеров-конструкторов и проектировщиков сложной техники было бы полезно объединить в рамках единой интерактивной системы средства автоматизации проектирования (САПР), пакеты математического моделирования с использованием ресурсов современных супер-ЭВМ (например, Логос [65]), а также средства визуализации на базе виртуальной реальности. Прототипы таких систем уже разрабатываются, хотя, создание полноценных сред проектирования является пока перспективной целью. При этом важной задачей становится разработка правильного интерфейса и методов визуального представления объектов проектирования.

Выработка критериев «правильности» требует дополнительного анализа. Опыт показывает, что оптимальность загрузки пользователей может оказаться негативным фактором, например, для интерфейсов рабочего места хирурга. В этом случае хирург в ходе операции может оказаться перегруженным управлением интерфейсом, в результате чего меньше внимания уделяется ходу операции и состоянию пациента.

И ещё про интерфейсы. Современные методы управления устройствами позволяют организовать управление автомобилями при помощи кнопок и сенсорных экранов, но откажутся ли сотни миллионов водителей от традиционных руля, педалей и ручки переключения передач. В случае интерфейсов для полноценного компьютерного управления сложными техническими объектами требуется учет функционирования этих объектов, а также задач их управления. Проектирование интерфейсов требует от разработчика серьезных знаний в разных областях.

Нужно сделать одно замечание. В плане основных сфер деятельности, связанных с компьютерингом, основное внимание уделяется информационным технологиям. Информационные технологии, конечно, важный компонент современного мира. Однако надо помнить, что в промышленности технологии производства вторичны по отношению к проектированию техники и уж точно по отношению к исследованиям и опытным разработкам. В области компьютеринга для нашей страны на первом месте должно стоять развитие компьютерных наук, а технологии должны быть результатом собственных исследований и опытных разработок в сфере программного обеспечения. В

этом случае мы сможем добиться технологической независимости, а не следовать за новыми западными технологиями. Для развития компьютерных наук необходима подготовка научных кадров и серьёзные усилия в поддержании уже имеющихся и создании новых научных школ. Образование в области компьютерных наук должно включать в себя математические дисциплины, относящиеся как к непрерывной, так и к дискретной математике, а также серьёзные знания в различных разделах программного обеспечения вместе с получением опыта реальных разработок. Такие же знания необходимы и разработчикам программного обеспечения человеко-компьютерного взаимодействия и визуализации. Например, для освоения элементов фотореалистичной графики необходимы знания в такой математической дисциплине, как уравнения математической физики, а для серьёзных разработок системного программирования необходимы знания в дискретной математике и математической логике. При разработке специализированных систем научной визуализации или визуализации для медицинских целей, необходимо прослушать краткие курсы лекций по соответствующей тематике. Также могут оказаться необходимыми знания по компьютерной психологии или семиотике, науке о знаковых системах. Чтобы быть готовым к пониманию новых идей, разработчики должны иметь разносторонние научные знания. Таким образом, не только учёные в сфере компьютерных наук, но и разработчики программных комплексов, предназначенных для научных вычислений, должны вместе с знаниями и навыками в информационных технологиях должны иметь очень университетскую научную подготовку, возможно, с дальнейшей специализацией в конкретных разделах.

И последнее. На заре компьютерной эры отец кибернетики Норберт Винер предупреждал о возможности выхода новой по тем временам техники из-под контроля её создателей. Винер рассказал о фантастическом рассказе Уильяма Джейкобса «Обезьянья лапа», в котором рассказывается о талисмানে – заколдованной обезьяньей лапке, которая могла исполнять желания, причём, они исполнялись таким образом, что приносили страшные несчастья обладателю. При этом формально желания были выполнены с точностью, хотя и совершенно не так, как мог себе представить «заказчик» [66]. Норберт Винер боялся, что компьютеры, не обладающие знаниями о человеческих ценностях и приоритетах, выйдут из-под контроля и принесут человечеству несчастья подобно обезьяньей лапе. Тогда было много разговоров о том, могут ли ЭВМ мыслить и не будут ли они пытаться покорить человечество. Но и тогда, и даже в семидесятых годах ЭВМ были под полным контролем программистов, поэтому, несмотря на развитие математической области «искусственный интеллект» предупреждения Винера казались избыточными. Сегодня вновь очень много говорят о искусственном интеллекте, который реализуется на базе нейронных сетей. Говорят, что компьютеры и роботы заменят человека чуть ли ни во всех областях деятельности. Впору вспомнить научно-фантастические книги пятидесятых годов. Прежде всего, три закона робототехники, которые придумал замечательный американский писатель Айзек Азимов [67]. Эти законы описывают этику взаимодействия человека и робота, обладающего искусственным интеллектом. Главная цель законов, закреплённых в самой конструкции электронного мозга роботов, – не допустить вреда для человека, в том числе вызванного приказом этого или другого человека. В современных описаниях искусственного интеллекта мало внимания уделяется этому вопросу. Использование искусственного интеллекта без анализа возможных опасностей может привести к появлению серьёзных проблем. Для преодоления эффекта обезьяньей лапы, то есть бездушного выполнения алгоритмов и команд, необходимо описание этики, а также знаний, касающихся человеческих ценностей и ценностных ориентиров. То есть для осуществления реального искусственного интеллекта требуются масштабные исследования и разработки.

Список литературы

1. Knuth D.E. Von Neumann's First Computer Program // Computer Surveys, Vol. 2, No, 4, 1970. p. 247-260.
2. Творцы первых советских ЭВМ. <https://odnarodyna.org/content/tvorcy-pervyh-sovetskih-evm>, Дата доступа – 20.08.20.
3. ЭВМ «Стрела». http://informat444.narod.ru/museum/1_17_4_strela.htm, Дата доступа – 20.08.20.
4. Трубка электронно-лучевая ЭВМ «Стрела». <https://polymus.ru/ru/museum/fonds/stock/trubka-elektronno-luchevaya-ln-4-evm-strela-122446/>, Дата доступа – 20.08.20.
5. Носители информации от древности до наших дней. http://www.computerhistory.narod.ru/nositeli_inf.htm, Дата доступа – 20.08.20.
6. Компьютеры в СССР. <https://visualhistory.livejournal.com/618431.html>, Дата доступа – 20.08.20.
7. Эволюция операционных систем. <https://ppt-online.org/94609>, Дата доступа – 20.08.20.
8. Виртуальный компьютерный музей. <https://www.computer-museum.ru/articles/personalnye-evm/897/>, Дата доступа – 20.08.20.
9. Teleprinter. <https://en.wikipedia.org/wiki/Teleprinter>, Дата доступа – 20.08.20.
10. Музей DataArt. Видеотерминал ADM-3A. <https://habr.com/ru/company/dataart/blog/453956/>, Дата доступа – 20.08.20.
11. Войскунский А. Е. Я говорю, мы говорим... Москва, Знание, 1982.
12. Будущее компьютерных моделей в индустрии моды и не только. <https://infuture.ru/article/1248>, Дата доступа – 20.08.20.
13. Графопостроитель. Первая половина семидесятых годов. <http://www.emohr.com/paris-montreal-1974/photos1974.html>, Дата доступа – 20.08.20.
14. Première phase d'avancement : la fonction vectorielle, http://jeanpierre.rousset.free.fr/Informatique/Noyau_Graphique/preambule_comments.html, Дата доступа – 20.08.20.
15. Интересные факты. История. <https://tunnel.ru/post-interesnye-fakty-istoriya>, Дата доступа – 20.08.20.
16. Математическое обеспечение графопостроителей. I уровень: СМОГ: Инструкция по программированию. / Под ред. Ю.А. Кузнецова. – Новосибирск: ВЦ СО АН СССР, 1976.
17. Баяковский Ю.М., Галактионов В.А., Михайлова Т.Н. Графор. Графическое расширение фортрана. М.: Наука. 1985.
18. Авербух В.Л., Каракина И.В., Подергина Н.В., Пономарева Л.С., Самофалов В.В., Соловьева Л.А. Реализация графической диалоговой системы ГРАДИС // Автометрия, 1978, № 5, с. 41-47.
19. The Emergence Of Software Engineering. <https://www.macmillanihe.com/blog/post/software-engineering-history-gerard-oregan/>, Дата доступа – 20.08.20.
20. Cray-1. <https://ru.wikipedia.org/wiki/Cray-1>, Дата доступа – 20.08.20.
21. Научное общество GraphiCon. <https://www.graphicon.ru/>, Дата доступа – 20.08.20.
22. Bondarev A.E., Galaktionov V.A., Chechetkin V. M. Analysis of the Development Concepts and Methods of Visual Data Representation in Computational Physics / Computational Mathematics and Mathematical Physics, 2011, Vol. 51, No. 4, pp. 624–636.
23. Cabral B., Hunter C.L. Visualization Tools at Lawrence Livermore National Laboratory // Computer, 1989. Vol. 22, No.8. Pp. 77-84.
24. Shu N.C. Visual Programming : Perspectives and Approaches // IBM System Journal. Vol. 22, No 4, 1989. pp. 525-547.

25. Brown M. Algorithm Animation. The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.
26. Stasko J.T. Tango: A Framework and System for Algorithm Animation // IEEE Computer. Vol. 23, No 9 (September 1990). Pp.27-39.
27. Price B.A., Small I.S., Baecker R.M. A Taxonomy of Software Visualization // IEEE Computer Society Press Reprint. 1992.
28. Авербух В.Л. Визуализация программного обеспечения. Конспект лекций. Екатеринбург. Математико-механический факультет. Уральский Государственный Университет. 1995.
29. Software Visualization. From Theory to Practice. Edited by Kang Zhang. KLUWER ACADEMIC PUBLISHERS. Boston, Dordrecht, London. 2003.
30. Diehl, S. Software visualization: visualizing the structure, behaviour, and evolution of software. Springer, 2007.
31. Диаграмма Ганта.
https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%93%D0%B0%D0%BD%D1%82%D0%B0 , Дата доступа – 20.08.20.
32. Radar chart. https://en.wikipedia.org/wiki/Radar_chart, (Accessed August 20, 2020).
33. Watch What I Do. Programming by Demonsration / Ed.- Allen Cypher / MIT Press. Cambridge, (Mass.), 1993.
34. Apple II. https://ru.wikipedia.org/wiki/Apple_II , (Accessed August 20, 2020).
35. IBM PC. https://ru.wikipedia.org/wiki/IBM_PC, (Accessed August 20, 2020).
36. Shneiderman B. The future of interactive systems and the emergence of direct manipulation // Behaviour & Information Technology. 1 (3). 1982. Pp. 237-256.
37. Забродин А.В. Супер ЭВМ МВС-100, МВС-1000 и опыт их использования при решении задач механики и физики // Матем. Моделирование, 12:5 (2000). Стр. 61-66.
38. Российский суперкомпьютер «Ломоносов-2» увеличит мощность с трех до пяти петафлопсов. <http://informaticslib.ru/news/item/foo/so6/no000622/index.shtml>, Дата доступа – 20.08.20.
39. Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual reality and parallel systems performance analysis // IEEE Computer, November 1995, vol. 28, no. 11. Pp. 57-67.
40. Современные очки виртуальной реальности, вид изнутри, <http://i.playground.ru/i/blog/111810/content/n6yrds wz.jpg>, (Accessed August 20, 2020).
41. Monash CAVE2.
<https://www.monash.edu/researchinfrastructure/mivp/access/facilities/cave2>, (Accessed August 20, 2020).
42. Bryson, S. Virtual Environments in Scientific Visualization. VRST '94 Proceedings of the conference on Virtual reality software and technology. Pp. 201-220. 1994
43. Авербух В.Л., Авербух Н.В., Бахтерев М.О., Васёв П.А., Зырянов А.В., Манаков Д.В., Стародубцев И.С., Щербинин А.А., Системные и визуализационные предпосылки создания виртуального испытательного стенда // Вопросы оборонной техники. Серия 14. 2012. Выпуск 2, стр. 20-26.
44. Васёв П.А., Вопросы выбора архитектуры интерактивного взаимодействия с параллельными программами // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа 29 марта - 2 апреля 2010 г.). [Электронный ресурс] Челябинск. Издательский центр ЮурГУ, 2010, с. 658-658.
45. Vincur J., Navrat P., Polasek I. VR City: Software Analysis in Virtual Reality Environment // 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, pp. 509 – 516.

46. Merino L., Ghafari M., Anslow C., Nierstrasz O. CityVR: Gameful Software Visualization // IEEE International Conference on Software Maintenance and Evolution (ICSME TD Track). 2017, pp. 633-637.
47. Авербух Н.В., Психологические аспекты феномена присутствия в виртуальной среде // Вопросы психологии. 2010. № 5. С. 105-113.
48. Brain-Computer Interface Detects Patient's Thoughts and Responses. <https://www.eletimes.com/brain-computer-interface-detects-patients-thoughts-responses>, (Accessed August 20, 2020).
49. DARPA's Brain-controlled Prosthetic Arm and a Bionic Hand That Can Touch. <https://singularityhub.com/2013/07/24/darpas-brain-controlled-prosthetic-arm-and-a-bionic-hand-that-can-touch/>, (Accessed August 20, 2020).
50. RealSpeaker читает по губам. <https://iz.ru/news/547160>, Дата доступа – 20.08.20.
51. What is eye tracking? <https://www.pantechsolutions.net/blog/what-is-eye-tracking/>, (Accessed August 20, 2020).
52. BMW HoloActive Touch: инновационный интерфейс для взаимодействия с автомобилем. <https://3dnews.ru/944494>, Дата доступа – 20.08.20.
53. Создано новое устройство, позволяющее имитировать тактильные ощущения прикосновения к мягким предметам. <http://ve-group.ru/sozdano-novoe-ustroystvo-rozvolyaushhee-imitirovat-taktilnyie-oshhushheniya-prikosnoveniya-k-myagkim-predmetam/>, Дата доступа – 20.08.20.
54. Foot Mouse & Foot Keyboard. <http://octopup.org/computer/foot-control>, (Accessed August 20, 2020).
55. I. Starodubtsev, V. Averbukh, N. Averbukh, D. Tobolin, Professional Natural Interfaces for Medicine Applications // Communications in Computer and Information Science / Ed. by C. Stephanidis. Springer International Publishing, 2014. Vol. 435. P. 435-439.
56. Видео жестового управления квадрокоптером. <https://www.youtube.com/watch?v=ybVV5ulGtos>, Дата доступа – 20.08.20.
57. Леонтьев А.Н. Деятельность. Сознание. Личность. М., Политиздат. 1975.
58. Рубинштейн С.Л. Основы общей психологии. - СПб. Питер, 2005.
59. Зинченко В.П. Эргономика и информатика // Вопросы философии. 1986. № 7. С. 53-64.
60. Zinchenko V. P. Activity theory: Retrospect and prospect // Proceedings "EAST-WEST" International Conference on Human-Computer Interaction: EWHCI'92. 4-8 Aug., 1992, St.-Petersburg, Russia. M. ICSTI, 1992. Pp. 1-5.
61. Авербух В.Л., Авербух Н.В., Наймушина А.В., Семенищев Д.В., Тоболин Д.Ю. Деятельностный подход при проектировании человеко-компьютерного взаимодействия: На примере медицинских интерфейсов. Изд. 2. М., URSS. 2019.
62. Blackler A.L., Hurtienne J. Towards a unified view of intuitive interaction: definitions, models and tools across the world // ММIIнтерактив, 13 (2007). Pp. 36-54.
63. Ершов А.П. К методологии построения диалоговых систем. Феномен деловой прозы. – Новосибирск, 1979. – 24 с. – (Препр./ АН СССР, Сиб. отд-ние; ВЦ; № 156).
64. Ершов А.П. Деловая проза как предмет общения с машиной на естественном языке // Человек и машина. Сб. публ. выступл. – М.: Знание, 1985. – С. 8–16. – (Новое в жизни, науке, технике. Сер. "Математика, кибернетика", № 4).
65. <http://logos.vniief.ru/products/>
66. Винер Н. Кибернетика, или Управление и связь в животном и машине. / Пер. с англ. И.В. Соловьева и Г.Н. Поварова; Под ред. Г.Н. Поварова. – 2-е издание. – М.: Наука; Главная редакция изданий для зарубежных стран, 1983. – 344 с.
67. Азимов А. Я, робот / М.: Эксмо, 2019.

Evolution of Human Computer Interaction

V.L. Averbukh^{1,A,B}

^A IMM UrB RAS

^B Ural Federal University

¹ ORCID: 0000-0002-4379-1450, averbukh@imm.uran.ru

Abstract

The work is devoted to the review of the development the human-computer interaction. In the first sections the history of computing in the "pre-computer" era is briefly described and then the early history of modern computing, methods of the first computers controlling and the tasks of programmers at this stage are described. It describes the methods of interaction with the first -generation computers using the remote control elements, punched cards and punched tapes. The section, devoted to the second generation computers, describes the emergence of high-level operating systems and programming languages. At this point, there are such means of interaction with the computer as the displays and, respectively, such programming tools as interactive languages and interactive debuggers. Research is also beginning on principles of human-computer interaction the infancy of the discipline "computer graphics", the development of computer graphics packages and the emergence of interactive computer graphics standards are considered. In the section "Revolutions in computer science" describes the appearance of a large number of the same series computers and the first super-computers in the context of human-computer interaction. Revolutionary changes are considered in computer graphics and emerging of the science discipline "computer visualization" with its parts "scientific visualization", "software visualization", "information visualization" and also "programming by demonstration". The information about the attempt to create a fifth generation computer based on logical programming is given. It is told about the initial period of teaching programming. The creation of computer networks and the emergence of personal computing as well as the creation the tools of modern parallel computing have become the important stages in the development of modern computing. The virtual reality becomes an important computer visualization tool .

The modern state of human-computer interfaces is characterized primarily by emerging of natural interfaces which can be attributed Brain-Computer Interface (Neurocomputer interface, Brain-Computer Interfaces), interfaces based on the direct use of nerve impulses, speech recognition, recognition of lip movement, mimic recognition and eye tracking (Eye Gaze or Eye Tracking), haptic interfaces and also interfaces giving tactile feedback (allowing you to feel the touch), motion capture interfaces the entire human body or individual organs (head, entire arm, hands, fingers, legs), motion capture toolkits ,in particular, interfaces based on leg movements (foot-operated computer interfaces), sign interfaces, sign languages. We briefly describe the activity approach to the design of interfaces and also some problems concerning the problem of mass interfaces. Finally, we discuss a number of problems arising from the increasing capabilities of modern computers. The work is in the nature of a popular science article and it largely reflects the subjective impressions of the author.

Keywords: history of human-computer interaction, computer graphics, computer visualization, computer networks, personal computing, natural interfaces.

References

1. Knuth D.E. Von Neumann's First Computer Program // Computer Surveys, Vol. 2, No, 4. 1970, p. 247-260.

2. Creators of the first Soviet computers. <https://odnarodyna.org/content/tvorcy-pervyh-sovetskih-evm>, (Accessed August 20, 2020).
3. Computer «Strela». http://informat444.narod.ru/museum/1_17_4_strela.htm, (Accessed August 20, 2020).
4. Cathode-ray tube of the computer "Strela", <https://polymus.ru/ru/museum/fonds/stock/trubka-elektronno-luchevaya-ln-4-evm-strela-122446/>, (Accessed August 20, 2020).
5. Information carriers from antiquity to the present day. http://www.computerhistory.narod.ru/nositeli_inf.htm, (Accessed August 20, 2020).
6. Computers in the USSR. <https://visualhistory.livejournal.com/618431.html>, (Accessed August 20, 2020).
7. Evolution of operating systems. <https://ppt-online.org/94609>, (Accessed August 20, 2020).
8. Virtual computer museum. <https://www.computer-museum.ru/articles/personalnye-evm/897/>, (Accessed August 20, 2020).
9. Teleprinter. <https://en.wikipedia.org/wiki/Teleprinter>, (Accessed August 20, 2020).
10. DataArt Museum. Video terminal ADM-3A. <https://habr.com/ru/company/dataart/blog/453956/>, (Accessed August 20, 2020).
11. Vojskunjij A. E. YA govoryu, my govorim... Moskva, Znanie, 1982.
12. The future of computer models in the fashion industry and beyond. <https://infuture.ru/article/1248>, (Accessed August 20, 2020).
13. Plotter. First half of the seventies. <https://images.app.goo.gl/vxHkXaKkw8cHYj507>, (Accessed August 20, 2020).
14. Première phase d'avancement : la fonction vectorielle, http://jeanpierre.rousset.free.fr/Informatique/Noyau_Graphique/preambule_comment_s.html, (Accessed August 20, 2020).
15. Interesting Facts. History. <https://tunnel.ru/post-interesnye-fakty-istoriya>, (Accessed August 20, 2020).
16. Matematicheskoe obespechenie grafopostroitelej. I uroven': SMOG: Instrukciya po programirovaniyu. / Pod red. YU.A. Kuznecova. – Novosibirsk: VC SO AN SSSR, 1976.
17. Bayakovskij YU.M., Galaktionov V.A., Mihajlova T.N. Grafor. Graficheskoe rasshirenie fortrana. M.: Nauka. 1985.
18. Averbuh V.L., Karakina I.V., Podergina N.V., Ponomareva L.S., Samofalov V.V., Solov'eva L.A. Realizaciya graficheskoy dialogovoj sistemy GRADIS // Avtometriya, 1978, № 5, s. 41-47.
19. The Emergence Of Software Engineering. <https://www.macmillanihe.com/blog/post/software-engineering-history-gerard-oregan/>, (Accessed August 20, 2020).
20. Cray-1. <https://ru.wikipedia.org/wiki/Cray-1>, (Accessed August 20, 2020).
21. Scientific Society GraphiCon. <https://www.graphicon.ru/>, (Accessed August 20, 2020).
22. Bondarev A.E., Galaktionov V.A., Chechetkin V. M. Analysis of the Development Concepts and Methods of Visual Data Representation in Computational Physics / Computational Mathematics and Mathematical Physics, 2011, Vol. 51, No. 4, pp. 624–636.
23. Cabral B., Hunter C.L. Visualization Tools at Lawrence Livermore National Laboratory // Computer, 1989. Vol. 22, No.8. Pp. 77-84.
24. Shu N.C. Visual Programming : Perspectives and Approaches // IBM System Journal. Vol. 22, No 4, 1989. pp. 525-547.
25. Brown M. Algorithm Animation. The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.
26. Stasko J.T. Tango: A Framework and System for Algorithm Animation // IEEE Computer. Vol. 23, No 9 (September 1990). Pp.27-39.
27. Price B.A., Small I.S., Baecker R.M. A Taxonomy of Software Visualization // IEEE Computer Society Press Reprint. 1992.

28. Averbuh V.L. Vizualizaciya programmnoho obespecheniya. Konspekt lekcij. Ekaterinburg. Matematiko-mekhanicheskij fakul'tet. Ural'skij Gosudarstvennyj Universitet. 1995.
29. Software Visualization. From Theory to Practice. Edited by Kang Zhang. KLUWER ACADEMIC PUBLISHERS. Boston, Dordrecht, London. 2003.
30. Diehl, S. Software visualization: visualizing the structure, behaviour, and evolution of software. Springer, 2007.
31. Gantt chart.
https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%93%D0%B0%D0%BD%D1%82%D0%B0 , (Accessed August 20, 2020).
32. Radar chart. https://en.wikipedia.org/wiki/Radar_chart, (Accessed August 20, 2020).
33. Watch What I Do. Programming by Demonsration / Ed.- Allen Cypher / MIT Press. Cambridge, (Mass.), 1993.
34. Apple II. https://ru.wikipedia.org/wiki/Apple_II , (Accessed August 20, 2020).
35. IBM PC. https://ru.wikipedia.org/wiki/IBM_PC, (Accessed August 20, 2020).
36. Shneiderman B. The future of interactive systems and the emergence of direct manipulation // Behaviour & Information Technology. 1 (3). 1982. Pp. 237-256.
37. Zabrodin A.V. Super EVM MVS-100, MVS-1000 i opyt ih ispol'zovaniya pri reshenii zadach mekhaniki i fiziki // Matem. Modelirovanie, 12:5 (2000). Str. 61-66.
38. The Russian supercomputer Lomonosov-2 will increase its capacity from three to five petaflops. <http://informaticslib.ru/news/item/foo/so6/n0000622/index.shtml>, (Accessed August 20, 2020).
39. Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual reality and parallel systems performance analysis // IEEE Computer, November 1995, vol. 28, no. 11. Pp. 57-67.
40. Modern virtual reality glasses, inside view.
<http://i.playground.ru/i/blog/111810/content/n6yrds wz.jpg>, (Accessed August 20, 2020).
41. Monash CAVE2.
<https://www.monash.edu/researchinfrastructure/mivp/access/facilities/cave2>, (Accessed August 20, 2020).
42. Bryson, S. Virtual Environments in Scientific Visualization. VRST '94 Proceedings of the conference on Virtual reality software and technology. Pp. 201-220. 1994
43. Averbuh V.L., Averbuh N.V., Bahterev M.O., Vasyov P.A., Zyryanov A.V., Manakov D.V., Starodubcev I.S., SHCHerbini A.A., Sistemnye i vizualizacionnye predposylki sozdaniya virtual'nogo ispytatel'nogo stenda // Voprosy oboronnoj tekhniki. Seriya 14. 2012. Vypusk 2, str. 20-26.
44. Vasyov P.A., Voprosy vybora arhitektury interaktivnogo vzaimodejstviya s parallel'nymi programmami // Parallel'nye vychislitel'nye tekhnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferencii (Ufa 29 marta - 2 aprelya 2010 g.). [Elektronnyj resurs] CHelyabinsk. Izdatel'skij centr YUurGU, 2010, s. 658-658.
45. Vincur J., Navrat P., Polasek I. VR City: Software Analysis in Virtual Reality Environment // 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, pp. 509 – 516.
46. Merino L., Ghafari M., Anslow C., Nierstrasz O. CityVR: Gameful Software Visualization // IEEE International Conference on Software Maintenance and Evolution (ICSME TD Track). 2017, pp. 633-637.
47. Averbuh N.V., Psihologicheskie aspekty fenomena prisutstviya v virtual'noj srede // Voprosy psihologii. 2010. № 5. S. 105-113.
48. Brain-Computer Interface Detects Patient's Thoughts and Responses. <https://www.eletimes.com/brain-computer-interface-detects-patients-thoughts-responses>, (Accessed August 20, 2020).

49. DARPA's Brain-controlled Prosthetic Arm and a Bionic Hand That Can Touch. <https://singularityhub.com/2013/07/24/darpas-brain-controlled-prosthetic-arm-and-a-bionic-hand-that-can-touch/>, (Accessed August 20, 2020).
50. RealSpeaker reads lips. <https://iz.ru/news/547160>, (Accessed August 20, 2020).
51. What is eye tracking? <https://www.pantechsolutions.net/blog/what-is-eye-tracking/>, (Accessed August 20, 2020).
52. BMW HoloActive Touch: innovative interface for interacting with the vehicle. <https://3dnews.ru/944494>, (Accessed August 20, 2020).
53. A new device has been created that allows you to simulate the tactile sensations of touching soft objects. <http://ve-group.ru/sozdano-novoe-ustroystvo-pozvolayushchee-imitirovat-taktilnyie-oshhushheniya-prikosnoveniya-k-mygakim-predmetam/>, (Accessed August 20, 2020).
54. Foot Mouse & Foot Keyboard. <http://octopup.org/computer/foot-control>, (Accessed August 20, 2020).
55. I. Starodubtsev, V. Averbukh, N. Averbukh, D. Tobolin, Professional Natural Interfaces for Medicine Applications // Communications in Computer and Information Science / Ed. by C. Stephanidis. Springer International Publishing, 2014. Vol. 435. P. 435-439.
56. Video of gesture control of a quadcopter. <https://www.youtube.com/watch?v=ybVV5ulGtos>, (Accessed August 20, 2020).
57. Leont'ev A.N. Deyatel'nost'. Soznanie. Lichnost'. M., Politizdat. 1975.
58. Rubinshtejn S.L. Osnovy obshchej psihologii. - SPb. Piter, 2005.
59. Zinchenko V.P. Ergonomika i informatika // Voprosy filosofii. 1986. № 7. S. 53-64.
60. Zinchenko V. P. Activity theory: Retrospect and prospect // Proceedings "EAST-WEST" International Conference on Human-Computer Interaction: EWHCI'92. 4-8 Aug., 1992, St.-Petersburg, Russia. M. ICSTI, 1992. Pp. 1-5.
61. Averbuh V.L., Averbuh N.V., Najmushina A.V., Semenishchev D.V., Tobolin D.YU. Deyatel'nostnyj podhod pri proektirovanii cheloveko-komp'yuternogo vzaimodejstviya: Na primere medicinskih interfejsov. Izd. 2. M., URSS. 2019.
62. Blackler A.L., Hurtienne J. Towards a unified view of intuitive interaction: definitions, models and tools across the world // MMIIInteraktiv, 13 (2007). Pp. 36-54.
63. Ershov A.P. K metodologii postroeniya dialogovyh sistem. Fenomen delovoj prozy. – Novosibirsk, 1979. – 24 s. – (Prepr./ AN SSSR, Sib. otd-nie; VC; № 156).
64. Ershov A.P. Delovaya proza kak predmet obshcheniya s mashinoy na estestvennom yazyke// Chelovek i mashina. Sb. publ. vystupl. – M.: Znanie, 1985. – S. 8–16. – (Novoe v zhizni, nauke, tekhnike. Ser. "Matematika, kibernetika", № 4).
65. <http://logos.vniief.ru/products/>
66. Viner N. Kibernetika, ili Upravlenie i svyaz' v zhivotnom i mashine. / Per. s angl. I.V. Solov'eva i G.N. Povarova; Pod red. G.N. Povarova. – 2-e izdanie. – M.: Nauka; Glavnaya redakciya izdanij dlya zarubezhnyh stran, 1983. – 344 s.
67. Azimov A. YA, robot / M.: Eksmo, 2019.