Scientific Visualization, 2020, volume 12, number 5, pages 130 - 164, DOI: 10.26583/sv.12.5.11

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.

1. Introduction

This work is devoted to an overview of the evolution of human-computer interaction. It describes the early history of computing and the methods of controlling the first computers.

Then the emergence of interactive tools and computer graphics, the emergence of computer visualization systems is considered. In the context of human-computer interaction, revolutions in computing, such as the emergence of computer networks and personal computers, are briefly described. More modern means of interaction are considered, for example, natural interfaces. It also discusses some of the challenges arising from the increasing capabilities of modern computers. The work has in the nature of a popular science article; it largely reflects the subjective impressions of the author.

2. Calculator - who is he?

Entire books can be devoted to the history of human-machine interaction as well as the history of computation. We will only very briefly touch on these issues. Although both machines and calculations appeared in ancient times, we will start with more recent moments, namely, from the era of industrialization, when modern methods of equipment management appeared, and also the problems of complex calculations related to the design of machines for various purposes arose.

The machines were operated by workers, but the calculations were carried out by highly qualified specialists in the field of computing such as scientists and engineers. Although computing tools such as the logarithmic ruler and the adding machine have been invented long time ago, as a rule, calculations in the field of technology, banking and insurance, were carried out manually for four- handed reliability .It means that the calculations were carried out by two specialists and compared at each step. If the results did not match, then both calculators recalculated the last stage.

To speed up and reduce the cost of solving problems, new computational methods were created. It was possible with the use of which to organize computer bureaus with less skilled employees. It is interesting that already in the XX century a computing bureau was created in the United States where women worked without special training, who were called computers (calculators). At the same time, such computing devices as electro-mechanical adding machines and perforating machines were created. Perforating machines manufactured by IBM were used to calculate the results of regular population censuses in the United States. In our country this technique was used in machine counting stations (machine counting bureaus) for accounting and other calculations.

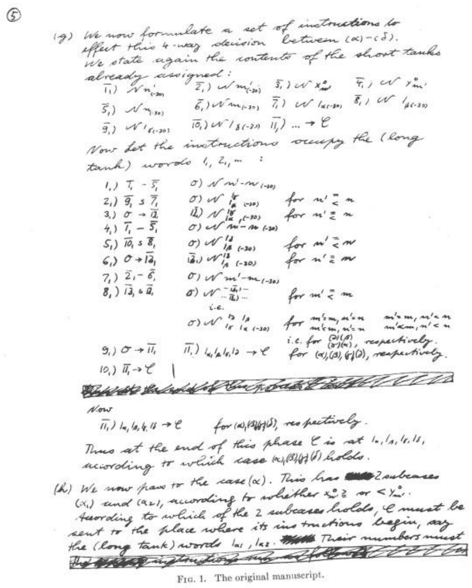
In the thirties of the XX century, the design and development of the first electronic calculating machines began around the world. Atomic projects in the USA and the USSR played an important role in their introduction into practice. However, the calculations of the first atomic bombs were carried out mainly manually and by outstanding theoretical physicists, mathematicians specializing in computational methods. Proto-programmatic methods were invented to facilitate and speed up the calculations (similar to those ones used in computers later).

For example, in the USA, physicists performed calculations according to a scheme similar to calculations in modern computers. The scheme looked something like this: a computing unit was extracted, which was calculated by one person, then the selection and verification of a certain value and its analysis followed, carried out by another person who transferred further calculations either to one or to another “calculator”. The Soviet atomic project used methods similar to parallel and distributed computing. Specially trained girls-calculators worked on electro-mechanical computers.

The calculations were divided into separate blocks, which were simultaneously calculated "in four hands". Then the individual results were and then collected into the overall score of the task.

Naturally, scientists, participants in atomic projects paid attention to the created electronic computers. There are examples of the first programs for the first computers, written by such prominent scientists as the American physicist and mathematician J. von Neumann and the

Soviet physicist Ya. B. Zeldovich. Figure 1 shows a fragment of the manuscript of John von Neumann's program [1].


⑤

(g) We now formulate a set of instructions to effect this 4-way decision between (α)–(δ). We state again the contents of the short tanks already assigned:

$\bar{1}_1$) $N\,n'_{(-30)}$ $\bar{2}_1$) $N\,m'_{(-30)}$ $\bar{3}_1$) $N\,x^0_{m'}$ $\bar{4}_1$) $N\,y^0_{m'}$
$\bar{5}_1$) $N\,n_{(-30)}$ $\bar{6}_1$) $N\,m_{(-30)}$ $\bar{7}_1$) $N\,l_{\alpha(-30)}$ $\bar{8}_1$) $N\,l_{\beta(-30)}$
$\bar{9}_1$) $N\,l_{\gamma(-30)}$ $\bar{10}_1$) $N\,l_{\delta(-30)}$ $\bar{11}_1$) $\ldots \to C$

Now let the instructions occupy the (long tank) words $1_1, 2_1, \ldots$:

```
1,) 1̄, - 5̄,        σ) N n'-n(-30)
2,) 9̄, s 7̄,        σ) N lγ/lα (-30)     for n' ≥/< n
3,) σ → 1̄2̄,        1̄2̄,) N lγ/lα (-30)  for n' ≥/< n
4,) 1̄, - 5̄,        σ) N n'-n (-30)
5,) 1̄0̄, s 8̄,       σ) N lδ/lβ (-30)     for n' ≥/< n
6,) σ → 1̄3̄,        1̄3̄,) N lδ/lβ (-30)  for n' ≥/< n
7,) 2̄, - 6̄,        σ) N m'-m (-30)
8,) 1̄3̄, s 1̄2̄,      σ) N ...1̄3̄,... / ...1̄2̄,...   for m' ≥/< m
                        i.e.
                   σ) N lδ lβ / lγ lα (-30)   for m'≥m, n'≥n   m'≥m, n'<n
                                                  m'<m, n'≥n   m'<m, n'<n
                                              i.e. for (δ)(β) / (γ)(α), respectively.
9,) σ → 1̄1̄,        1̄1̄,) lα, lβ, lγ, lδ → C   for (α),(β),(γ),(δ), respectively.
10,) 1̄1̄, → C  |
```

~~[illegible]~~

Now

$\bar{11}_1$) $l_\alpha, l_\beta, l_\gamma, l_\delta \to C$ for (α),(β),(γ),(δ), respectively.

Thus at the end of this phase C is at $l_\alpha, l_\beta, l_\gamma, l_\delta$, according to which case (α),(β),(γ),(δ) holds.

(h) We now pass to the case (α). This has ~~[illegible]~~ 2 subcases (α₁) and (α₂), according to whether $x^0_{m'} \geq$ or $< y^0_{m'}$. According to which of the 2 subcases holds, C must be sent to the place where its instructions begin, say the (long tank) words $l_{\alpha 1}, l_{\alpha 2}$. ~~[illegible]~~ Their numbers must ~~[illegible]~~

FIG. 1. The original manuscript.

Fig.1. Manuscript of John von Neumann's program [1].

3. The first computers

In the fifties of the XX century, the creation of a computer began. In the United States, IBM became the leading computer manufacturer. However, as they said in the late forties IBM considered the production of computers unprofitable, since, according to analysts, no more than five such machines were needed around the world. There were two in the United States, one each in England, Germany and the USSR. Germany was defeated and the USSR became the enemy of the United States in the Cold War, so that number was reduced to three. IBM was probably influenced by the American government to get the company into what seemed like a dubious business.
In our country the first electronic computer MESM was made in Kiev under the leadership of the future academician S.A. Lebedev. Figure 2 represents the team of creators near the MESM [2].

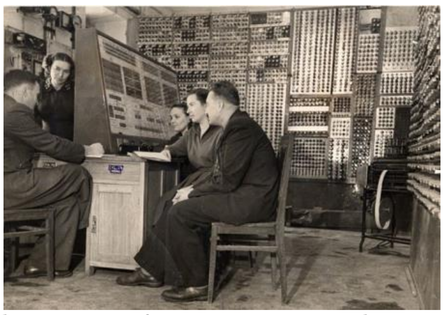


Fig. 2. The team of creators near the MESM [2].

By the end of the fifties the production of first-generation computers (based on electronic tubes) had expanded all over the world. Various models of computers were developed with different command systems. The speed of these machines was from several hundred to thousands of operations per second. According to today's scale, the memory of such machines was very small. At first, programming was carried out exclusively in codes. One of the first Soviet machines, Strela (Fig. 3), was implemented on the basis of cathode ray tubes (Fig. 4), which was less efficient compared to tube machines, but provided an interesting opportunity to interact with programmers.

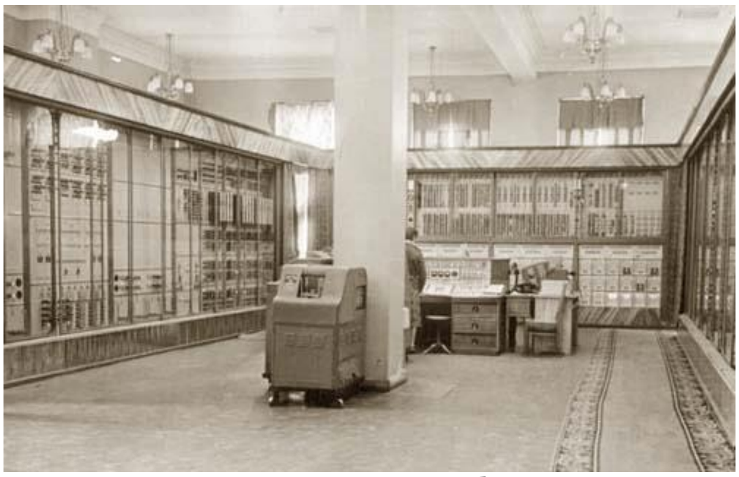


Fig. 3. Computer Strela [3].

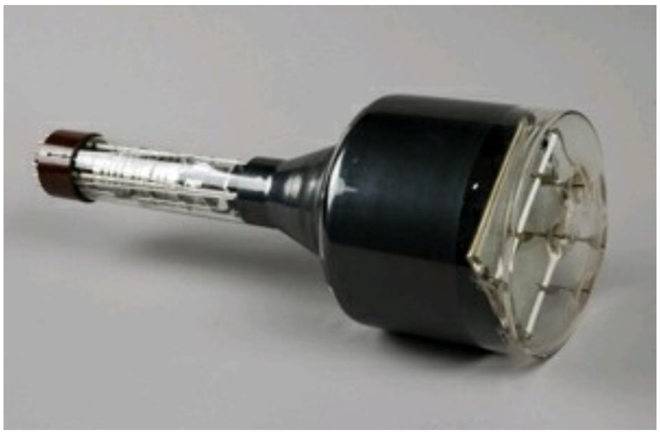


Fig. 4. Electron-beam tube LN-4 computer "Strela" [4].

Computers at this time were used exclusively for scientific computing, for solving complex problems related to physics, chemistry and engineering calculations. Complex computational methods were developed and / or used by mathematicians and physicists. There were these specialists who developed new programs based on these methods.
In our country teaching programming in the fifties began at the mathematics and physics departments of universities and other institutes. At Moscow Power Engineering Institute (MPEI), a faculty that trained specialists in the field of computers was created. A new profession a programmer was taken shape. As a rule, these specialists had good training in the field of numerical methods. They were engaged in the solution and computer implementation of applied problems. A number of technical universities began training electronic engineers to work with computer facilities

4. Interaction with computers of the first generation

Now let's consider what way the programmers interacted with computers of the first generation. The remarkable M-20 machine could be singled out among these machines in our country with a speed of twenty thousand operations per second. This computer was developed at the Institute of Precision Mechanics and Computer Engineering (ITMiVT) of the USSR Academy of Sciences under the leadership of S.A. Lebedev.
To work on a first-generation computer “computer time” was allocated , that is tens of minutes or hours during which the programmer was the full owner of the computer.
Programming was carried out in machine codes. The program was recorded using command numbers and cells. In the fifties programs were already introduced using punched tapes (Fig. 5) and punched cards (Fig. 6), borrowed from the technology of the past (for example, tele-

graph and perforated computing devices). The printing was carried out on a relatively narrow (less than twenty centimeters) paper tape. Although the internal representation of numbers and commands was binary (but there were also ternary machines, for example, the Soviet machine Setun), the programmer mainly worked with octal and decimal numbers. It is interesting when teaching programming, the newly-minted programmers were afraid of they would have to both multiply and divide numbers in the binary system in the future. But at the next step it turned out that they could do with the ordinary decimal numbers.

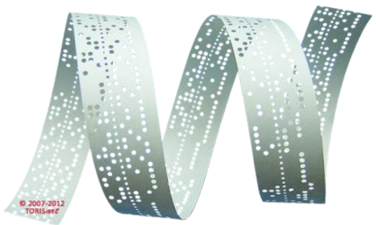


Fig. 5. Punched tape [5].

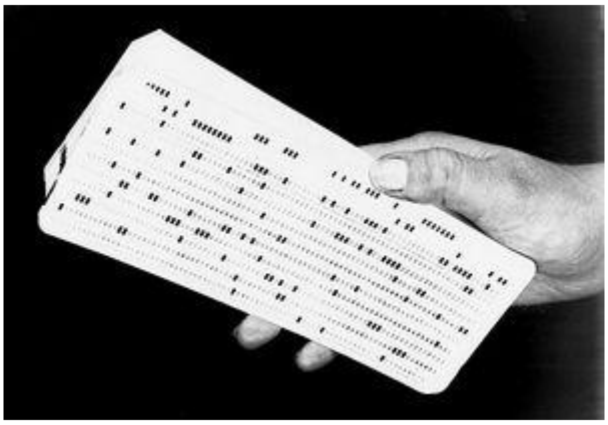


Fig. 6. Deck of punched cards [6].

All interaction with the computer went through the console (Fig. 7). The console had a set of buttons and toggle switches as well as indicators for displaying registers and cells. The programmer entered the program and launched it himself using the buttons on the computer control panel. Almost no program was launched the first time. An often long and arduous debugging process (debugging) was required. During debugging, the programmer inserted "stop" commands in key places of the program and could see to which of the nodes control was transferred and what was the state of the cells and registers of the computer when executing this section of the program.The "stop" commands were removed after debugging.
The reliability of the first generation computers was low and the machine broke down almost every hour. It was necessary to organize a round-the-clock duty of electronic engineers and technicians since the work was carried out day and night.

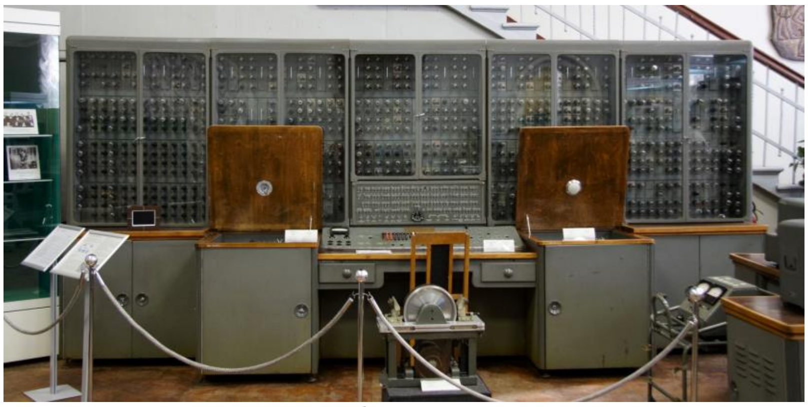


Fig. 7. Ural-1 computer [7].

5. The development of computing technology

The situation changed with the creation of computers based on semiconductors (second generation computers) in the sixties, which reliability and productivity dramatically increased. For example, the Soviet BESM-6 in 1965 was the fastest computer in the world (Fig. 8). This computer was also developed at the ITMiVT of the USSR Academy of Sciences under the leadership of S.A. Lebedev. Its speed was one million operations per second. It's true that the RAM was tiny by today's standards, a little more than thirty-two thousand six-byte cells. This machine worked smoothly almost around the clock with a short break for maintenance. On the second generation machines such external memory devices as magnetic tapes and magnetic drums were actively used.
It was on the second generation machines (primarily FORTRAN and ALGOL) that high-level operating systems and programming languages were introduced. In our country to describe the basic concepts, a language different from Western one was used and it is used now. For example, they said "machine", not "computer", "autocode", not "assembler", "translator", not "compiler".
Operating systems (OS) provided the execution of application programs and their interaction with external input and output devices as well as with external memory such as magnetic tapes, drums and disks. A batch, multitasking mode of computer operation, was implemented in which the operating system processed a package of programs that were allocated a certain time slice for execution.
At the same time the application programmer was completely removed from communication with the computer. The new specialties such as computer operator, system administrator and system programmer appeared. There were the specialists who continued to interact with computers through an increasingly complex console. The division between application and system programmers was quite tough.
For example, in the early instructions for BESM-6 two modes of command execution were distinguished: the programmer mode (system programmer with great capabilities of some system commands) and the mathematician (applied programmer) mode.

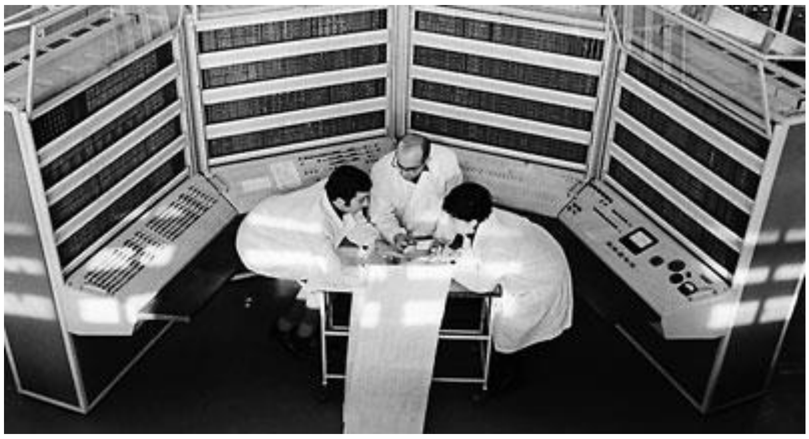


Fig. 8. Computer BESM-6 [8].

In addition to using the remote control, interaction with the operating system could be organized through videoterminals connected to the computer (Fig. 10), as well as in cases there were practically no displays (as in our country), via teletypes (Fig. 9).
The operator and the "system engineer" (the system administrator and the system programmer who had access to control the operating system) gave OS commands related to managing the general computation's progress and received information from the operating system about the progress of individual tasks.

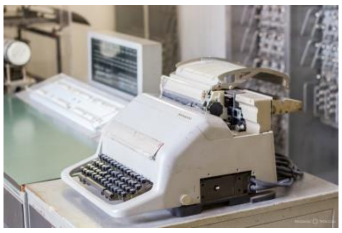


Fig. 9. Teletype [9].

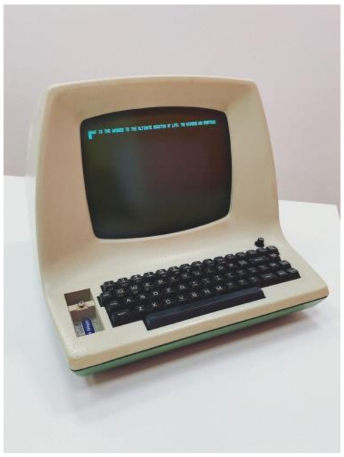


Fig.10. Alphanumeric display, [10].

Applied programmer usually transferred the text of the program to special staff (punch-girls) who typed on punched cards. Then the programmer assembled the punched cards into a package and handed it over to the computer operators, who entered the programs through input device . The operating system determined when the program would be launched and the result of the count was issued for "wide printing", that is on an alphanumeric printing device (APD). For novice programmers, most of the launches were wasted due to numerous compilation errors associated with ones when entering punched cards and other small beginner errors. Debugging the program took quite a lot of time.
The introduction of such means of interaction as teletypes and displays made it possible for the emergence of dialogue languages, such as the predecessor of Basic, JOSS language (the domestic version was Dialogue BESM-6). The programmer working with language entered the text of the program and started the execution, having the opportunity to check the results of calculations on each operator. The development of dialog debuggers for autocodes (assemblers) and high-level languages such as FORTRAN and ALGOL began. In the sixties and seventies there were attempts to create Russian-language autocodes (assemblers) and programming languages. For example, the BESM-6 autocodes were russian-speaking Autocode SOMI and BEMSH (whereas the Madlen autocode was in English). A Russian-language version of ALGOL was created.
When debugging a program, a programmer could switch to a step-by-step (operator-by-operator) execution mode or set a "stop" command on a certain operator and display the results of calculations at each step. Dialogue debuggers dramatically reduced debugging time for complex programs. Subsequently, the widespread introduction of alphanumeric displays and the development of appropriate software allowed the computer programmer with "batch" operating systems to enter program texts without using punched cards and ran programs from their workplace.
In the first half of the seventies recommendations for the design of interfaces already appeared. In particular, it was stated as important information on the display screen had to be located diagonally from the upper left corner of the screen and the most important piece of information was located in the center of the screen. This follows the usual order of reading and writing texts (from left to right and from top to bottom). It is interesting that for foreign

users, who had the order of reading and writing texts from right to left, were recommended to arrange information on a diagonal starting from the upper right corner. It was also recommended to design the dialogue systems so that the response time fits within three seconds. In some cases users were not ready to wait even one second for a response and became nervous because of waiting. In other cases if the system displayed the progress of the task (for example, the loading process), the users could wait several tens of seconds. The recommendations for choosing the color scheme of the displayed information appeared after the introduction of color displays. In many respects they were similar to the recommendations for advertising design.

The problems studying of interaction with computers has become one of the directions of computing development. In the early eighties one of the founders of Russian computer psychology A.E. Voiskunsky described the important aspects of human-computer dialogue [11].

6. Formation of computer graphics

Let's talk about the formation of mashine (computer) graphics . The creator of the first computer graphics system Sketchpad (Fig. 11) was Aiven Sazerland (Ivan Sutherland). His scientific adviser was the information theory creator K. Shannon . However, according to the information of the "zero" generation programmers, engineers of the very first machines began to use recorders and oscillographs to output information about the counting results. On the Strela computer, building on the basis of cathode-ray tubes, it was also possible to output an image on the tubes that gave information about the stability of the program execution . The image was stable when everything was in order with the program. When something went wrong, the image blurred or even disappeared. It was the prototype for software visualization, a discipline that emerged in the 1980s.

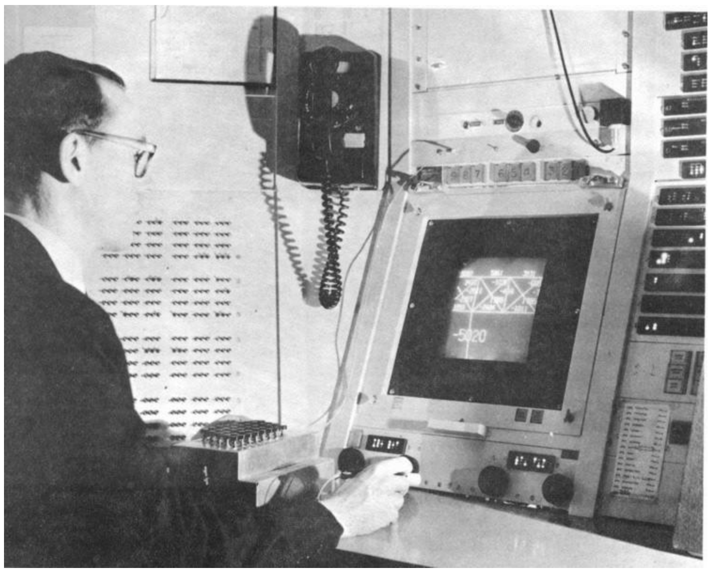


Fig. 11.Aiven Sazerland working in the Sketchpad system [12].

Two types of graphics devices were developed. There were devices of "hard copy" (plotters, graphplotters) and devices of "soft copy" (graphic displays), hard copy and soft copy in English.

The first graphic displays resembled the radar screens and may have been based on them. It seemed natural to make displays based on television receivers, but for a long time bitmap displays could not be used, since to store the image it was necessary to remember the state of all the pixels (picture elements) of the screen. Even the most primitive screen of 128x128 lines required storage of 16384 bits of information. Vector black-and-white displays were widely used in which the coordinates of the vector’s origin and the shifts in the X and Y coordinates were transmitted to the screen. It was necessary to update it continuously to obtain a stable

image. The display memory was relatively small and the display began to blink to update the image with a large amount of information due to the fact that there was not enough time. Because of this, the display capabilities of such devices were small. It was almost impossible to display a complex 3D picture.
An alphanumeric keyboard and telephone discs were used as input devices. Later, in addition to the alphanumeric and keypad, a device such as a light pen was used (Fig. 13). The light pen could be used in three modes as pointing to a graphic object (the program received the name of the object; on the basis of this, it was possible to implement "light buttons"), pointing to a point on the screen (the program received the coordinates of the point) and drawing (the program received a set of coordinates entered lines). Graphic displays could have their own memory or they could use the memory of the computer to which they were connected. In the latter case, it was easy to animate 2D images. It was necessary to move the image by two arc minutes (in the case of a small screen about 2 mm) and do it a little faster than half a second. For example, various input devices were introduced with a mouse (at first it was only called a bug) (Fig. 14) and a joystick (wand of joy), the trackballs, the touch-screens. Such interfaces, implemented through operations with any devices, could be called Device Interfaces.
In devices of "hard copy" (plotter) (Fig. 12) it was possible to produce multi-colored graphics by switching pens (later - markers). Color displays in the 70s were too complex and expensive. According to some sources in the USA a color display cost $ 100,000. In our country color displays were produced in single copies for special tasks. By the way, in the mid-70s a foreign black-and-white display (incomplete configuration) cost 10,000 gold rubles for Soviet scientific institutes, that was more than 11,000 dollars of that time.

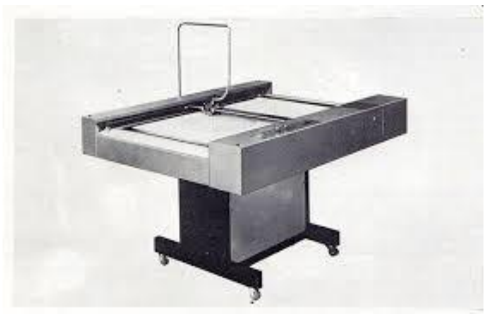


Fig. 12. Plotter. First half of the seventies [13].

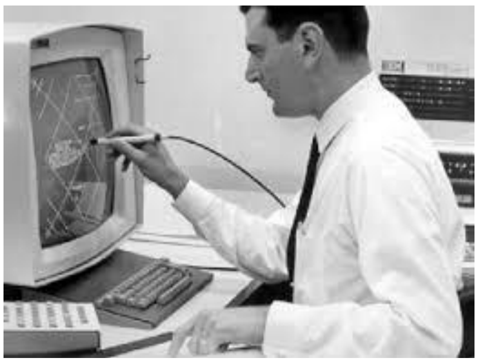


Fig. 13. Light pen [14].

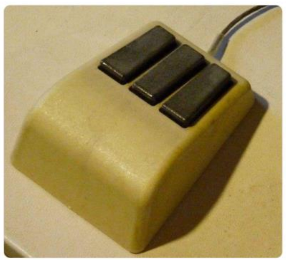


Fig.14. The computer mouse of the early seventies [15].

Machine (computer) graphics were used to visualize the results of scientific computing and in design automation systems (CAD - in English it is Computer-aided design, CAD systems). However, the output of a more or less serious drawing in A1 format could require several hours of work of the plotter.
Software was developed actively (at first computer graphics packages). In the early seventies a number of graphic packages were developed in our country. At first, it is necessary to mention such packages as SMOG (Computing Center of the Siberian Branch of the USSR Academy of Sciences, Novosibirsk), [16] and GRAFOR (Institute of Applied Mathematics USSR Academy of Sciences, Moscow) [17]. For example, packages of dialog (interactive) computer graphics were also developed [18]. In the West such packages appeared in the sixties. In our country, the development of it was carried out in the mid-seventies. Foreign plotters and displays were often used, although domestic devices were also developed.
Interactive computer graphics gave a great effect in the computer's problems of various modeling processes. It was possible to quickly view a significant number of frames with a graphical display of various model's elements. As a result, the applied mathematician found himself "inside" his model and could observe its changes due to the input of parameters and interaction with the program. The presence of graphical input was particularly important as it was entering the coordinates of a point or entering a curve using a light pen (as a result, the program received two arrays with the coordinates of the entered points). Sometimes working with the program in an interactive mode and displaying graphs made it possible to solve a complex problem of computer modeling in one session, although before that specialists spent many weeks on the solution, conducting hundreds of launches.
In the second half of the seventies, the development of standards for interactive computer graphics began. A draft ,developed in the USA ,of a standard as the Core Graphics System was published. In our country, this draft standard was implemented in several organizations. A little later, another project the Graphical Kernel System (GKS) was developed in Germany which was adopted as an international standard. Despite the very interesting ideas inherented in these standards, they did not give the expected results, since already in the eighties another revolution in computing technology and, accordingly, in computer graphics began.

7. Revolution in computing

In general, the development of computing technology is a series of successive revolutions, which often almost cancel the achievements of previous years. Thus, the development of relatively powerful and reliable second-generation computers based on semiconductors made it possible to introduce operating systems and programming languages into practice. The art of programming in codes had been developed by that time and the ability to squeeze a serious program into the limited resources of tube computers turned out to be superfluous. In the sixties, IBM developed the IBM System / 360 series of computers (figure 15) based on semiconductor technology (Fig. 15) with a common command system for both small and powerful (at that time) machines.
The computers of this series had the same software (operating system and programming system) from the point of view of 02 - 5 users . Later, the IBM System / 360 series was supplemented with the IBM System / 370 series, after the appearance of third-generation computers based on integrated circuits. In many countries around the world, analogs of the IBM System / 360 were created. In our country, this analogue was called ES EVM. The presence of computers with common programming systems made it possible to ensure easy implementation of the results obtained in one organization into another, even located in another country. The number of computers in the eighties around the world was already tens of thousands of copies. These computers were widely used in commercial and scientific computing.

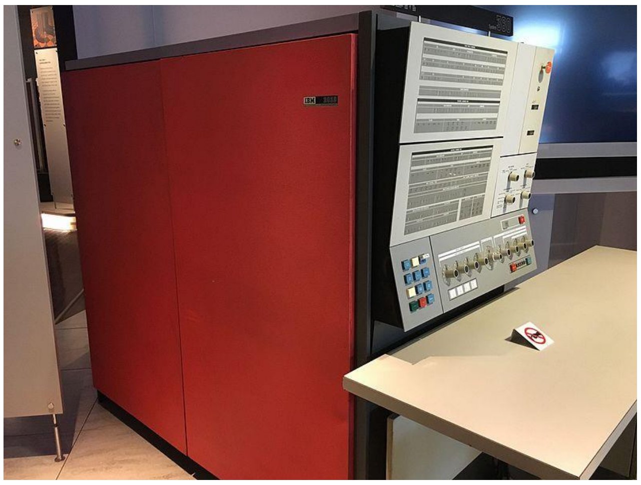


Fig. 15. Computer IBM-360/30 [19]

In addition to computers of "average" power, the development of a supercomputer for scientific computing tasks began. In particular, in the USA the supercomputers (by the standards of that time) Cray-1 (Fig. 16) and Cray-2 were developed.

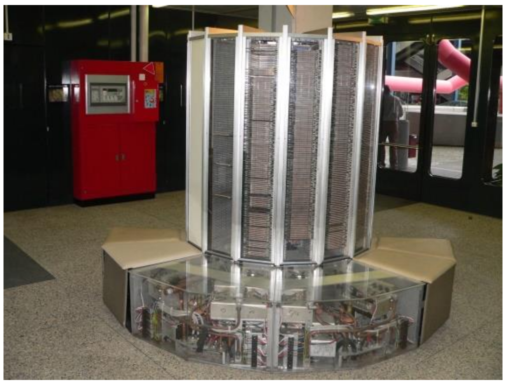


Fig. 16. Computer Cray-1 [20].

Back in the sixties, the so-called mini-computers appeared. They were designed to control technological processes and complex objects, for example, sea vessels. Mini-computers were also used in interactive computer graphics systems. They were connected to the "main" computer, on which the main computation and the formation of data for visualization were carried out. Mini-computers carried out direct output to graphic displays and support for interaction with users by inputting information on display input devices [18].
The development of aviation and rocket technology required the creation of a micro-computer that could be used to control aircraft and missiles. Precisely micro-computers became the basis for the next revolution in the world of computers.
Computers of high and medium power were occupied in computer rooms .They were large, specially equipped rooms entirely filled with metal cabinets with main and peripheral devices, including random access memory, external memory devices (magnetic tapes, magnetic disks), input and output devices, current rectifiers and transformers. The minicomputers were much more compact and were the size of one or two double-leaf wardrobe. The transition to integrated circuits made it possible to reduce the computer size, but the Cray still occupied an extensive room.
In the seventies and eighties the development of a number of areas in computer technology began. There were such changes in the life of all people on Earth as computer networks, personal computing and cellular communications. But now we will consider changes in computer graphics that took place in the eighties.

8. Changes in computer graphics

Developments in the field of computing circuitry and memory made it possible to create reliable, compact and inexpensive color output devices based on raster graphics. Graphics stations were developed as specialized computers that were connected to the "big" computer. In our country a series of graphic stations Gamma (Novosibirsk, Institute of Applied Physics) was developed. With the help of the graphic Gamma stations, the data obtained during the study of the planet Venus were visualized. Gamma-5 graphics cards turned alphanumeric displays into graphic displays.
The creation of high-quality graphics devices served as a prerequisite for the algorithmic support development. Algorithms such as ray tracing and radiosity were developed by American undergraduates and graduate students in the 1980s.
Ray tracing was based on the ancient idea of Aristotle that rays from a person's eyes fall on objects. However, the idea was used in the opposite way. It is assumed that the emission of a ray from an object located on the scene constructed by the algorithm on the screen (more precisely, a separate pixel) onto the human eye or the camera that determines the point of view. If the ray emanating from the first object collides with another object in front of the first, then the elements (pixels) of the second object are displayed on the screen and the corresponding elements (pixels) of the first one are not displayed. Pay attention that the algorithm is not considered efficient enough, but it parallelizes perfectly.
The radiosity algorithm was based on the assumption that all objects in the scene constructed by the algorithm either reflect or absorb light. With the software implementation of the algorithm, the corresponding equation of mathematical physics was solved. Other algorithms for photorealistic graphics were developed, which served as the basis for modern computer graphics. Later, the implementation of the algorithms was "wired" into specialized graphics processing units (GPUs), which became one of the foundations of modern computing.
It is interesting that some developers of photorealistic graphics algorithms came to the first conferences of GraphiCon [21] in Moscow.
Nowadays computer graphics have become not only a scientific discipline, but also an important branch of the modern computer industry.

9. Computer visualization

The development of modern computer graphics served as a prerequisite for the design of computer visualization as an independent discipline.
Visualization is described as a tool or method for interpreting graphic data entered into a computer and generating images from complex multidimensional datasets. It is obvious that visualization, that can be understood as the visible representation of mental models, existed long before the advent of modern computing. Moreover, visualization, that is the translation of data and information into some graphic images, can be considered as an integral part of our daily life. Initially, immediately after the creation of the first computers, visualization of counting results was understood as any output of numbers or symbols on a tape of a primitive printing device, an ATPU sheet or a display screen. Gradually, visualization began to be understood only as graphical output, for example, drawing two-dimensional graphs or three-dimensional surfaces. The publication of the report "Visualization in Scientific Computing" in the November 1987 issue of ACM SIGGRAPH Computer Graphics magazine marked the beginning of a new era in the history of computer visualization. It's important to say that the report was created under the auspices of the US National Science Foundation.
Let us first give the basic definitions and then talk about the history of the discipline's development.
Computer visualization refers to the technique of translating abstract representations of objects into geometric images, which enables the researcher to observe the computer modeling results of phenomena and processes.
The following sub-areas of computer visualization are traditionally distinguished:

- scientific visualization;
- software visualization;
- information visualization.

Scientific visualization refers to the use of computer graphics and human-machine interaction to represent data about objects, processes and phenomena simulated in scientific calculations.

Software visualization is understood as a set of techniques for using graphics and human-machine interaction tools used for a better understanding of concepts and effective operation of software as well as for the specification and presentation of software objects in the process of creating programs.

The term information visualization refers to the visual description and presentation of abstract information obtained as a result of the collecting process and processing data of various types and purposes. As a rule, this data does not have a natural and obvious graphical presentation. Information visualization combines scientific visualization and human-machine interaction methods. Information visualization techniques are largely associated with such disciplines as obtaining new knowledge from databases (data mining or knowledge discovery) and visual analytics.

Note also that visualization is often simplistically understood only as a direct mapping of three-dimensional images (rendering) onto a certain output plane or even as a simple set of visual and iconic interactive methods. By the way, these are very important questions that should rather be attributed to the problems of computer graphics and human-machine interaction.

Despite the different areas of visualization application, there is a deep unity of all its subsections, both in the methods of constructing display types (up to rendering techniques) and in terms of the ultimate goals and objectives to ensure the interpretation and analysis of computer modeling results. All this allows us to single out computer visualization as an independent discipline with its own subject and research method.

10. Scientific visualization

It is clear that the use of graphics to represent the results of scientific computation dates back to the very beginning of the computer era. In the 1980s, developments in graphics hardware and software enabled American researchers to rapidly deploy scientific visualization to a new level. For example, a special issue of Computer magazine was published in August 1989 , devoted to scientific imaging and containing the work of researchers from NASA, the laboratory of the Department of the Navy and other important research centers. The visualization quality was very good even by today's standards. Soon, other special issues on this topic were published. The history of the concepts' development and methods of scientific visualization is presented in sufficient detail in work [22].

Later, developments in the field of scientific visualization were implemented on powerful graphics devices, including virtual reality environments. It will be discussed below.

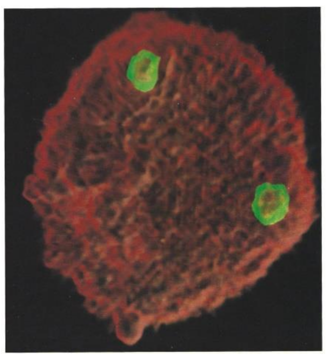


Fig. 17. An example of three-dimensional visualization of the human cell's nucleus, late eighties [23].

At present, a whole series of annual international conferences around the world are devoted to scientific visualization. A typical example of a three-dimensional human cell’s visualization of the nuclei, implemented in the late eighties, is shown in Figure 17.

11. Visualization software

In the eighties, a large number of scientific journals devoted to computer topics were published. The materials of these journals were available to domestic researchers either through scientific libraries or through abstract journals, in which reports of almost all scientific articles were published. The full texts of these articles were available on paper or microfilm. The analysis of publications related to computer graphics allowed to identify the emergence of two new directions the visual programming and the program visualization. It is interesting that at first it was difficult to separate these concepts due to the similarity of the names.

Visual programming implied the use of graphics, in particular schemes, diagrams, iconic images (icons) in the process of developing programs. Visualization of programming assumed that already developed programs were presented in the form of the same graphic elements. Within the framework of programming visualization, one could single out such sections as animation of algorithms, visual debugging etc.

The ideas of visual programming languages appeared in the seventies when computer graphics systems began to work quite steadily. Visual languages of the seventies were built on the basis of flowcharts or diagrams of Nassi-Shneiderman, which served to describe structured programming. This direction had developed sufficiently by the end of the eighties. Diagrammatic and iconic programming languages were developed, compilers of visual languages were created, [24].

The first animation system for algorithms was developed back in the mid-sixties. Individual frames were shot on film, and the result was a movie describing the operation of the algorithm. In the eighties, animation systems of algorithms were implemented onthe basis of modern computer graphics at that time [25]. An interesting idea of algorithmic operations was proposed which served as the basis for the animation "script" [26].

In general,visual debuggers were based on ideas for interactive debugging. The user had the opportunity to view, for example, the program trace to see a graphical displaying the data that the user was interested in. The development of visual debugging systems for parallel computing began. For this purpose both natural graphics for the application were being debugged and traditional schemes and diagrams could be used.

The idea of creating software visual complexes was put forward, which consisted in the fact that both the development of programs, their debugging and their maintenance had to be carried out within a single system with the same graphical representation of software entities.

By the late 1980s, it became clear that a new discipline Software Visualization had taken shape. In the early nineties, the first publications with its description appeared [27]. In our country, in 1995 a textbook on this discipline was published [28]. Later the monographs on software visualization were published in the USA [29, 30]. Visualization of software received a great development in connection with the development of supercomputing in the nineties. Figure 18 shows an example the architecture of three-dimensional visualization of the software package made in the Vizz3D system [30].

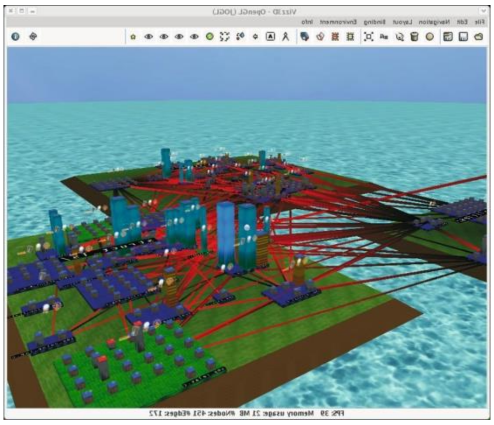
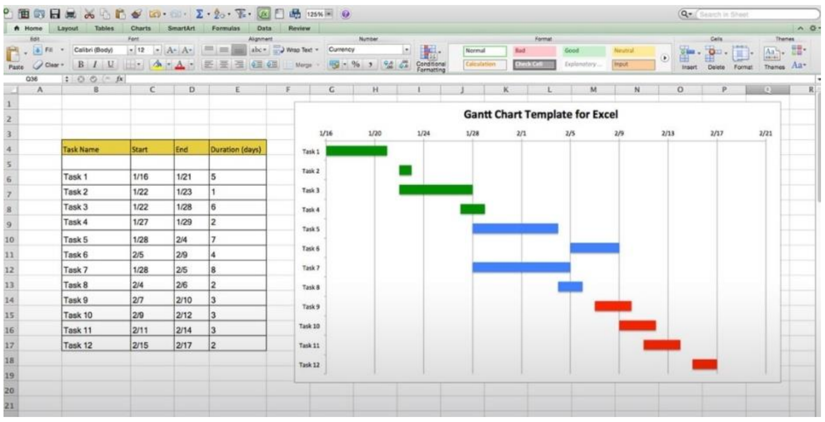


Fig. 18. An example the architecture of three-dimensional visualization of a software package made in the Vizz3D system [30].

12. Information visualization

At first, information visualization was based on "pre-computer" statistical graphics, that was on graphical methods of presenting statistical information. Various types of graphs, charts and diagrams were used, for example, Gantt charts (Fig. 19) and Kiviat charts (Fig. 20).



Task Name	Start	End	Duration (days)
Task 1	1/16	1/21	5
Task 2	1/22	1/23	1
Task 3	1/22	1/28	6
Task 4	1/27	1/29	2
Task 5	1/28	2/4	7
Task 6	2/5	2/9	4
Task 7	1/28	2/5	8
Task 8	2/4	2/6	2
Task 9	2/7	2/10	3
Task 10	2/9	2/12	3
Task 11	2/11	2/14	3
Task 12	2/15	2/17	2

Fig. 19. The example of Gantt chart [31].

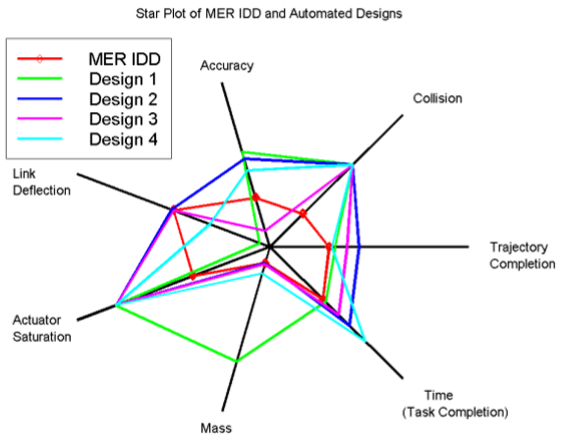


Fig. 20. The example of Kiviat chart [32].

Even simple visualization techniques could be very effective. So the use of moving graphs displaying the patient's condition (temperature, blood composition, test results for infections, etc.) and a set of medical procedures (use of medicines, physiotherapy, etc.) made it possible for a specialist to identify effective procedures quickly for specific patients. Note that all the initial information was at the disposal of doctors , but it was the simultaneous movement of

the graphs that helped to see the connection between the patient's condition and the treatment being carried out (for a certain contingent of patients).
Later, in view of the need to visualize big data, for example, data on activity in social networks, the researchers started using of three-dimensional graphics, drawing complex graphs, etc. It’s important that information visualization methods are used in systems for debugging the efficiency of parallel programs.

13. Programming by patterns (programming by demonstration)

The "Pattern Programming" section was associated with the problem of software visualization. It was popular in the eighties and nineties.
The term "Pattern Programming" was defined for systems that allow the programmer to use patterns of input and output information during the programming process. There were two main aspects to programming by patterns (examples) of input, output and the process of logical inference (or guessing) of the program from these patterns.
In literature the terms "programming by demonstration", "visual learning", "programming by rehearsals" and others were also used. Although some authors contrast these terms, but we will use "programming by patterns" and "programming by demonstration" as equivalent.
There was no equivalent in textual sequential programming for many visual aspects of programming by demonstration. Programming by demonstration was carried out by manipulating data on a screen that demonstrated to the computing system what the program had to do. There were the advantages of this approach because it was easier for the programmer to accomplish something than to describe it textually. The user had to be able to instruct the computer, “Watch what I do” and the computer would create a program corresponding to the user's actions. For many systems, programming by demonstration consisted in constructing a program continuously from the execution track of examples demonstrated by the user.
The principles of programming by demonstration were largely close to the methods of teaching programming, in which ready-made examples’ demonstration of correct programs played a significant role. Then the student generalized the experience gained to solve new problems similar to the demonstrated problems. If the demonstration of ready-made sample programs was a powerful method of teaching a novice programmer, then programming through demonstrations was a technique for teaching a computing system, a method of demonstrating to a computer samples of input and output information necessary for the user, so that programs interacting with the user were built on their basis.
The main goal of programming by demonstrations was to provide the end user with tools for creating ready-made programs, while at the same time, if it was possible without burdening it with the need to program (in the usual sense) in languages like C, Pascal or Prolog.
The main aspects of programming by demonstrations were the formation of patterns (examples) of input, output and the process of inference (or guessing) a program from these patterns. It is clear that an essential part of programming systems through demonstrations was the means of providing input and output the necessary information samples. Naturally, these systems had to contain knowledge about the application areas in which the resulting programs would operate as well as a set of rules for inference of programs based on knowledge.
Therefore, it was possible to characterize programming systems through demonstrations by the following parameters:
1. Application area and type of users for programs received by the system;
2. Methods of interaction with users of the system;
3. A set of inference rules for programs;
4. Knowledge of the application area.
Programming systems through demonstrations was going from unpretentious systems in the early seventies to quite sophisticated, equipped with artificial intelligence systems, developed in the nineties [33]. Many of the works were funded both by the government departments of

the United States and Canada (mainly by the military) and by the largest companies as IBM, Apple, Xerox etc. However, further interest in this topic began to wane. Perhaps this was due to the fact that the developed systems ceased to be relevant due to the rapid development of computer technologies. At the same time, the approaches proposed in this area can be used, for example, in robot training systems. It seems that interesting programming ideas through demonstration may come in handy further.

14. Fifth generation of computers

In the eighties, another attempt was made to create a new and important direction in the development of computer technology. The idea of a fifth generation computer was put forward. At that time, three generations of computers were known as on the basis of lamp, semiconductor and computers on integrated circuits. The fourth generation of computers was not considered. The fifth generation assumed a breakthrough in the field of computing, programming and human-computer interaction. Ideas related to artificial intelligence were put forward, attempts were made to create a new type of supercomputer and programming approaches based on inference. These programming approaches were supposed to provide a fundamentally new work with a computer, which had to understand input in natural languages, speech recognition, language to language translation. Fifth generation computers were programmed in the Prolog language. Knowledge bases and expert systems were actively developed within the framework of the project.

Knowledge bases are a collection of facts and rules of inference in the chosen subject area of activity. For example, you can provide the kinship knowledge base with rules such as "if X is the father of Y, then Y is the son of X". And then ask information that Sergey is Vasily's father. Then the system will be able to find Sergei by the query “find Vasily's son”. Knowledge bases (databases), supplied with predicates, can be an effective tool in more complex versions.

Expert systems had to include both the knowledge of a specialist and the rules of inference received from him that are relevant for a specific area.

The fifth generation computer project was proposed by Japanese researchers and developers. It seemed that it was Japan that would make a breakthrough into the future of computing and programming, overtaking the whole world. However, the development of computer technology for a number of reasons took a different path. Many factors played a role in the insufficient power of computers of the eighties, such as the difficulties of logical programming, implementing full-fledged artificial intelligence, and, most importantly, the computing revolution associated with personal computers and the Internet. At the same time, it is necessary to emphasize the importance of developing knowledge bases and expert systems, which are currently receiving insufficient attention.

Let’s consider some issues related to the initial training in programming before proceeding to a description of the emergence of personal computing and computer networks.

15. School programming

In our country, teaching programming to schoolchildren began in the early sixties. Students of specialized (mathematical) classes were taught basic information on number systems, mathematical logic, computer devices, command systems and calculation methods. During the training, a summer practice was carried out during which the students wrote and debugged programs on real computers. Until the mid-seventies teaching programming was carried out mainly in machine codes, as many thought that mastering programming languages would make learning difficult. By this time, languages such as Algol and Fortran were already widely used in the practice of applied programming. The theory of algorithmic languages and the practice of their application led to the fact that many hundreds of languages and compilers were implemented in the world. Note that these languages used English vocabulary as a basis for describing operators and basic concepts. It’s true that there was a point of view that the use of national languages in this capacity would facilitate the initial training and master-

ing of programming. By the way, there was a variant of the implementation of Algol with a full translation into Russian.
Domestic scientists participated widely in teaching programming to schoolchildren in the framework of school and out-of-school education. It had to be noted the results of researchers led by Academician A.P. Ershov from the Computing Center of the Siberian Branch of the USSR Academy of Sciences. Such languages of elementary teaching programming based on the Russian language, such as Robik and Rapira, were developed, as well as a programming system for these languages.
At the turn of the seventies and eighties, programming olympiads for schoolchildren began. However, due to the limited access to computers loaded with real problems, the Olympiads were held "dry" and the jury checked only the texts of programs written on paper as well as solutions of logical problems.
In the second half of the eighties the mass training in programming of teachers, university professors and, most importantly, all schoolchildren began in our country. Scientists of the USSR Academy of Sciences took an active part in the work on the computerization of education. For example, in Sverdlovsk region, this process was led by Academician N.N. Krasovsky. At this time, personal computers were already widely used abroad, which were used as a training base. A significant number of personal computers were purchased, which were placed in classrooms, to which high school students from all over the region were brought. Mobile classrooms housed in buses came to remote areas. Programming was taught in BASIC language.
As a result of work on computerization, all-union olympiads in programming began to be held.
In the nineties, the subject "computer science" entered the secondary school curriculum. Teaching programming on the basis of personal computers was carried out, as a rule, using Pascal language. However, talented students mastered new computer technologies on their own and often outstripped their teachers.

16. Personal computers

Except for mainframes, various specialized computers were created to solve specific problems. For example, back in the sixties in our country in the Cybernetics Institute of the Academy of Sciences (Ukrainian SSR) under the leadership of academician V.M. Glushkov, specialized computers of the MIR series (machine for engineering calculations) designed to solve various engineering problems were developed. For the MIR-2 computer, the ANALITIK high-level language was developed as an input language. This language made it possible to formulate tasks with analytical transformations of formulas, to directly carry out operations of differentiation and integration. A display was used to input and output information to the MIR-2.
It seemed that the Apple II machine that appeared in the seventies was also a specialized computer.

Fig. 21. Apple II [34].

The fact is that in domestic programmer circles there were rumors about two American students Steve Wozniak and Steve Jobs. They were said to have been kicked out of the company where they worked part-time for playing computer games during working hours.
As a result, both students assembled a special computer for games in the garage, which they called the Apple II. The display of this computer worked in alphanumeric and graphic modes, which was its advantage. It was the Apple II (Fig. 21) that laid the foundation for modern personal computing, although there were previously computers like this one in a number of their parameters. It’s important that the Apple II was actually a versatile and easy-to-use computer. For example, it was used for serious imaging programs designed to manage the supply of drugs in large hospitals.
Later, there was talk about what the future would belong to personal computers. This already seemed incredible because it seemed that the future was in large computers like the Cray., This is probably the reasoning of IBM executives, who, according to rumors, refused to produce personal computers until offices for selling personal computers, appeared in the Corporation's buildings, and employees began to buy and use them for work. By the way, many people were later convinced that Microsoft was a subsidiary of IBM.
However, it was personal computers that became the basis of the new computer world. Their users became not only and not so much programmers, but also researchers of various categories, engineers, office workers. Although personal computers were produced all over the world, the largest manufacturers in the eighties and the first half of the nineties were Apple (Macintosh computers) and IBM (IBM PC computers (Fig. 22)).

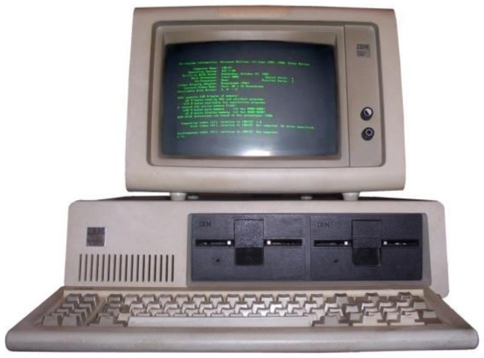


Fig. 22. IBM PC [35].

Millions of computers were produced. The main qualities of personal computers allowed them to conquer the world. These were compactness (a computer could be installed at every workplace), comparative cheapness, reliability and ease of use (maintenance of personal computers did not require a staff of electronics engineers, system programmers and operators), ease of software development, updating programs and installing new software, convenient means of interaction with a computer and software.
The introduction of mass personal computers simply could not take place without the appearance of visual means of interaction, new devices and fundamentally new concepts of organizing the interface with the user, for example, such as the concept of “direct manipulation”.
The concept of "direct action" was proposed in the early eighties by the well-known specialist in the field of computational sciences, professor B. Shneiderman, who brought together and analyzed new trends in the organization of the interface. Currently, this concept dominated in the interface design [36].
B. Shneiderman defined the following characteristics of the interface, created on the basis of the concept of "direct action":
1) Permanent display of the object of interest;
2) Physical actions (working with a mouse, joystick, touch screen, etc.,or using a functional keyboard instead of commands with complex syntax);

3) Fast, step-by-step, returnable operations whose impact on the object of interest was immediately visible.
The essence of this approach to creating an interface was to create the impression that the user directly affected the objects presented on the screen, and did not conduct a dialogue with the computer about these objects. Instead of using a command language to describe operations on objects, the user manipulated the visible representations of these objects on the display screen [36].
The use of personal computers by office workers, primarily managers of various levels and their secretaries served as the basis for the emergence of such a concept as a metaphor of the interface. The interface metaphor was considered as the basic idea of convergence and analogy between application domain model objects and interactive objects. First of all, the interface metaphor began to be used in the Desk Top metaphor variant.
This metaphor united real-world objects on the surface of an office worker's desk (for example, folders with documents), as well as the iconic representation of programs and, most importantly, "magic" operations, such as double-clicking the mouse to open folders and launch programs. Interestingly, attempts to expand the desktop metaphor, to make the desk three-dimensional with drawers and openable drawers, and also to develop a metaphor for the office room were not successful.
It is possible that the desktop metaphor was the basis for the powerful advancement of personal computing. All new versions of personal computers were released, and the old ones were treated in a very peculiar way. At one time, a secretary competition was held in the United States, during which girls had to run to the fourth floor in heels with system units on their hands and throw the system unit out of an open window. The winner was the girl whose system unit fell first. There were also more useful ideas for using old computers. Their processors were used to compose distributed computing systems with zero cost.
Currently, millions and millions of both traditional personal computers and laptops, tablets and smartphones are annually produced.

17. Computing networks

It is believed that the basis of the modern Internet was a network developed in the late sixties and early seventies by order of the US Department of Defense Advanced Research Projects Agency (DARPA) and designed to manage military facilities in the event of a nuclear war. It should be noted that at the beginning of the seventies, networks were already actively used connecting various computers. In our country, academician V.M. Glushkov proposed to use computer networks for effective management of the national economy. Somewhat later, work began on projects for global (nationwide) computer networks. However, due to the development of networks, the idea was put forward that networks had to be used to transfer tasks from a computer to a computer in approximately the same way as electricity is pumped. That is at a time when it was already night in Vladivostok and computers were not loaded, they could download programs from overloaded computing centers in Moscow. This seemed ridiculous, since in the seventies and eighties computers were idle only during preventive maintenance or failures. Therefore, although the projects of such networks were partially implemented, they were not introduced into serious practice.
"Normal" computer networks began to work in our country at the end of the eighties and were included in the world Internet. Ubiquitous email access was gradually implemented. Already in the nineties, together with search engines, it became possible to search, read and download scientific publications posted on foreign sites. This was especially important, since literally from September 1991 the foreign scientific journals ceased to arrive in scientific libraries and researchers in the first half of the nineties could only access them through their Western colleagues. This was especially important, since literally from September 1991 the foreign scientific journals were no longer available in scientific libraries.

The power of computer networks was low. So in order to download an article of 500-600 kilobytes in size, it took several hours, and sometimes you had to leave the download overnight. The urgent transfer of files with visualization of important mathematical models from Yekaterinburg to Moscow demanded that all academic institutes and universities of the city were disconnected from the Internet for half an hour. The fact is that the visualization was received literally an hour before the opening of the academic exhibition in Moscow. The opening was attended by the leaders of the Russian government and it was necessary to show them the latest research results of the institute.

Later, the speed and quality of the Internet increased, social networks and online computer games appeared. Around the same time, a cellular network based on different principles began to spread, but later, when “advanced” (smart) phones appeared, it became possible to access the Internet from cell phones. Billions of people from all over the world became the Internet users who had access to the Internet through personal computers, laptops, tablets and phones. In scientific and non-scientific literature, the term "Internet addiction" appeared, which described the fact that many users spend almost all their free (and not free) time on the Internet.

18. Parallel computing

Already in the sixties, the use of parallel computing began for various applications. At this time this area did not attract much attention from a wide range of developers and users. However, by the second half of the eighties, new solutions appeared that made parallel computing the most important direction of modern computing. Transputers, which included a central processor and four communication channels for two-way exchange with other devices, became effective devices for organizing parallel computing. Creation of parallel computers based on transputers prompted the development of parallel programming tools, for example, the Occam programming language.

In the mid-nineties, a number of programs were adopted in the United States, within the framework of which government support was provided for the creation of supercomputing based on parallel computing. First of all, the nuclear national laboratories (Los Alamos, Livermore, Sandia), NASA research centers and the Ministry of Defense took part in the development of parallel supercomputers. In these research organizations, a lot of attention was paid to parallel supercomputing. The computer base was regularly updated. It was in these centers that the use of graphics processing units (GPUs) and game console processors began in the development of parallel computers.

In general, parallel programming depended on the used parallel architectures. MPI, a software toolkit for providing communication between branches of a parallel application, could be used for systems with distributed memory. MPI standed for Message passing interface. The OpenMP parallel programming library could be used for shared memory systems.

In our country in the early nineties, despite difficult conditions, the development of parallel computers and related software began. In the first half of the nineties, the academic and industry institutes and organizations of our country developed a parallel computer MVS 100. Such computers worked in several organizations. Later, a series of MVS supercomputers was developed, in particular, MVS 1000, installed in the Interdepartmental Supercomputer Center and accessibled via remote access for a wide range of users [37]. Supercomputers were also developed in a number of other organizations in the country. Domestic basic software was created for domestic supercomputers, in particular, an operating system, a file management system and the computer graphics tools.

Parallel supercomputers are now widely used all over the world. The rating of the most powerful computers in the world is regularly published. Dozens of supercomputer centers have also been created in our country. Supercomputing conferences are regularly held. Figure 23 shows the Russian supercomputer «Lomonosov» located at Moscow State University.

Fig. 23. Supercomputer "Lomonosov" in Moscow State University [38]

The use of such powerful computing technology led to problems due to the cost of electricity and water for cooling. Californians were said to be protesting during the construction of new centers, fearing a shortage of electricity and water. New methods of cooling computers were being developed.
The widespread use of parallel supercomputers posed new challenges for visualizing the data obtained to ensure the analysis and interpretation of the results. The huge volume of the resulting files and the complexity of the parallel architectures set the task of organizing data output for computer graphics systems. In the future, it was a need to develop tools for remote and online visualization of parallel computing.
Since the end of the eighties, active development of software visualization tools has been carried out visual parallel programming languages, visual correctness debuggers and efficiency debuggers for parallel programs in connection with parallel computing. The latter were used to predict, find and avoid possible inefficiency in the execution of parallel programs.
It is interesting that in the early nineties visualization system for the Avatar software was developed [39], actively using virtual reality tools and metaphors of a room and a building in a three-dimensional version and operating on the basis of a virtual reality environment such as CAVE. The Avatar system was designed for debugging the performance of parallel programs and allowed presenting large amounts of data on the performance of parallel processes obtained during the operation of a supercomputer. In the course of work, the user seemed to be inside a three-dimensional room, on the walls of which the video image was projected. Curves describing the performance metrics of parallel programs in the form of two-dimensional graphs were displayed on the floor and walls of this room. The view was similar to a glass skyscraper, consisting of rooms, each of which contains graphical output characterizing various aspects of the described parallel program behavior. A transparency mode was provided for the ceiling and floor, which made it possible to see adjacent data in adjacent "rooms". A visual display was defined a "history tape", to represent the sequence of the supercomputer processors. A "virtual flight" over a skyscraper based on this tape made it possible to explore aggregate data on the performance of an applied parallel program.
More details about virtual reality will be discussed in the next section.

19. Virtual reality systems

Virtual reality is a historical term that refers to a computer-generated environment using special devices such as special helmets (glasses) Fig. 24), screens with the illusion of three-dimensionality, CAVE systems (Fig. 25) (virtual reality environments in which images are projected on the walls, floor and ceiling of the room). These devices allow to create a special environment, that is perceived by the user as the real world, in which he really is (but does not observe from the outside) and with which he interacts directly, as well as with the ordinary world.

Fig. 24. Modern virtual reality glasses, inside view [40].

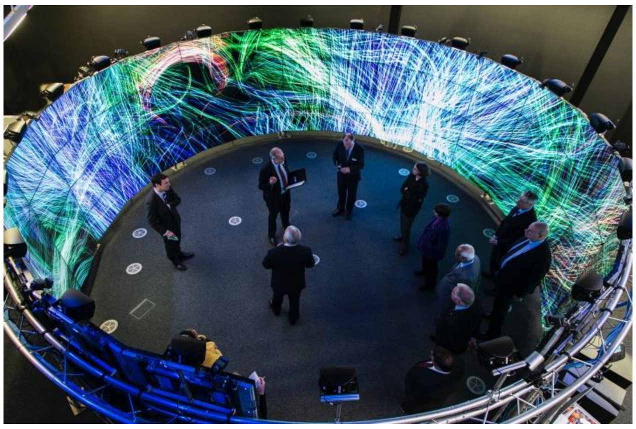


Fig. 25. Virtual reality system CAVE-2 [41].

In the literature you can find information that the first virtual reality system was developed by Ivan Sutherland, one of the founders of computer graphics. However, it seems that virtual reality traced its history back to aviation simulators, in which elements of flight simulation and even air combat were displayed on large screens (in our country) or on special helmets (in the USA). It was these helmets that served as the basis for the first virtual reality systems in the eighties.

In the late eighties and early nineties virtual reality environments were used both for computer visualization tasks and in simulators. The use of these media for games and entertainment also began. The project of a virtual wind tunnel implemented at NASA for the design and testing of the Shuttle spacecraft is the great interest to us. Since it was impossible to conduct tests in a conventional wind tunnel for such an apparatus as the Shuttle, a full-fledged model was created and displayed in a virtual environment. The researchers could change the parameters of the model and see the results [42]. By the way, in 1994 in Nizhniy Novgorod a very interesting report on a virtual wind tunnel for Shuttles was made by S. Bryson ,the lead developer of the system, at the Graphikon conference.

A generalization of the idea of a virtual wind tunnel was the idea of a virtual test bench for those studies in which a full-scale experiment was either difficult to organize or impossible. In these interactive systems, it was necessary to provide the ability to repeatedly run a program that simulated processes with different parameters. In systems of a virtual test bench, it was supposed to carry out computational experiments to simulate such complex technical objects as spaceships, rockets or airplanes with visualization of the results by means of virtual reality. Computer simulation usually required the use of parallel and distributed computing. There was a need to use virtual reality in online visualization mode [43, 44].

Virtual reality was actively used in software visualization systems. These systems often used visualization metaphors, for example, the metaphor of the city [45, 46]. The visualization metaphor was understood as the main idea of object representation, which helped the perception of visual images.

Also in the nineties, the development of augmented reality tools began, overlaying computer-generated images on real objects. It became possible to use augmented reality tools to organize technological processes for assembling complex devices. The assembly technician could receive visual cues, which made his work easier and more efficient.

Currently, virtual reality environments are used for entertainment, educational, scientific and therapeutic purposes. For example, virtual reality systems are being developed that make it possible to treat phobias by gradually bringing a person into contact with a frightening object. The most important phenomenon associated with virtual reality is the phenomenon of presence. Presence refers to the feeling of being in a virtual environment as if it were real. This experience is described as "being there" [47].
Already in the nineties, predictions appeared that humanity would soon move to the virtual world. At the same time, virtual reality environments and the virtual world of the Internet, in which millions of users "sit", are often confused. These predictions are actively disseminated to this day. It seems that the authors do not fully understand the essence of virtual reality as a tool for human-computer interaction.
We also note some limitations when using virtual reality associated with possible unpleasant and even painful sensations that may be caused by the user. This may be due to manifestations of cyberbullying (cybersickness is a health disorder similar to seasickness). Cyber disease is often associated with the inability to manage virtual reality events actively [47].

20. Natural interfaces

There are several definitions of natural interfaces. In some cases, it is assumed that user operations are intuitive and based on natural everyday behavior. Others talk about a virtually invisible interface based on natural elements (or becoming one after the user has mastered it).
In this article, natural interfaces are understood as interfaces based on fixation and recognition of any human movements' combination or the activity of his organs.
Natural interfaces provide "head-to-toe" interaction. In particular, the following interfaces can be distinguished:

- Brain-computer interface;
- Interfaces based on the direct use of nerve impulses;
- Interfaces based on speech recognition;
- Interfaces based on lip recognition;
- Interfaces based on recognition of facial expressions;
- Interfaces based on recognition of gaze movement (Eye Gaze or Eye Tracking);
- Tactile interfaces as well as interfaces that give tactile feedback (allowing you to feel touch);
- Interfaces based on motion capture of the entire human body or individual organs (head, entire arm, hands, fingers, legs);
- Interfaces based on motion capture tools, in particular, foot-operated computer interfaces;
- Sign interfaces, sign languages.

In this case, it is necessary to take into account the possibility of combining several natural human activities within the framework of one implementation.
Brain-computer interfaces are based on the recognition of the brain's own electrical activity, associated, in particular, with the movements of the arms and legs and the formation of commands to move real or virtual objects (Fig. 26). This direction has been actively developing over the past decades both abroad and in our country. There are interesting results on the use of interfaces of this type in virtual and augmented reality systems both for manipulating objects and for navigating in virtual space.

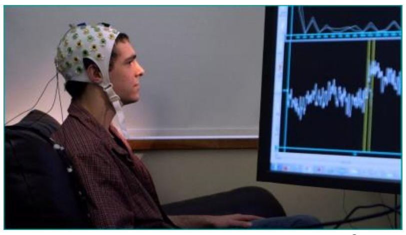


Fig. 26. Brain-computer interface [48].

Interfaces based on the direct use of nerve impulses can be used to develop new types of prostheses (Fig. 27). There are also known examples of hand prostheses built on the basis of neurocomputer interfaces.

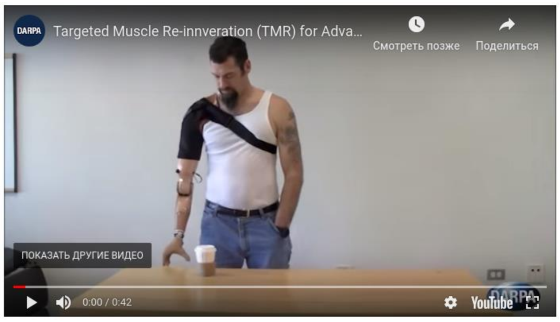
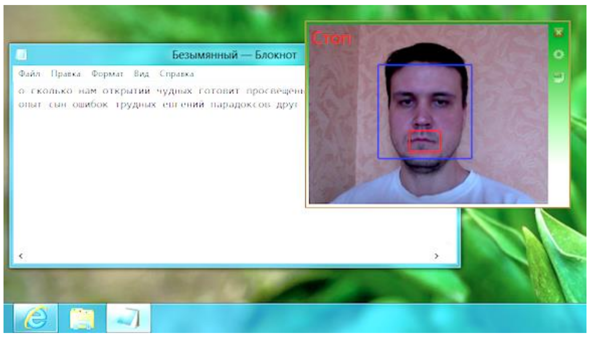


Fig. 27. Video frame of manipulation of a hand prosthesis controlled by neurocomputer interface [49].

Speech recognition is one of the most popular applications of artificial intelligence ideas. The real results have been obtained in this direction. Existing applications are already used in everyday life (for example, requests to mobile devices, elevator control, etc.). Interfaces based on lip recognition can be considered as auxiliary, allowing toimprove the accuracy of speech recognition systems (Fig. 28).



о сколько нам открытий чудных готовит просвещен
опыт сын ошибок трудных енгений парадоксов друг

Fig. 28. Interface based on speech recognition [50].

Recognition of the direction of gaze and human facial expressions can be used to organize human-computer interaction in visualization systems (Fig. 29).

Fig. 29. Interface based on eye tracking [51].

On the helmets used in flight simulators, which served as the basis for virtual reality systems, the image was adjusted depending on the direction of the pilot's gaze. Interfaces based on recognition of facial expressions and direction of gaze are also being developed to provide communication for people who have lost the ability to move or even speak. Recently, the use of gaze recognition-based interfaces in games has become popular. These interfaces are often used in conjunction with other types of natural interfaces.
Tactile interfaces can be interesting in connection with the creation of gesture interfaces and providing feedback when working in virtual reality environments and with "large" screens (Fig. 30, 31).

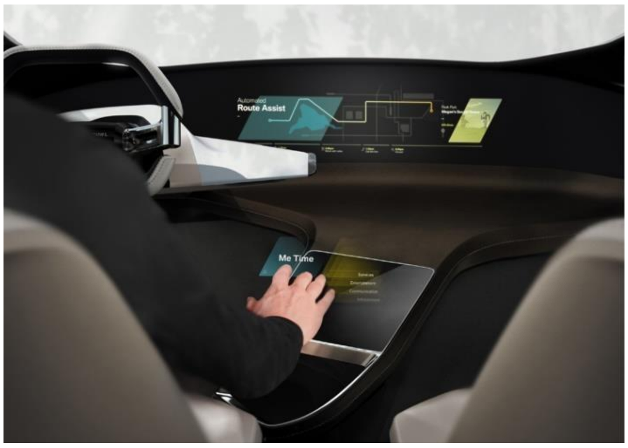


Fig. 30. The example of tactile interface for driving a car [52].

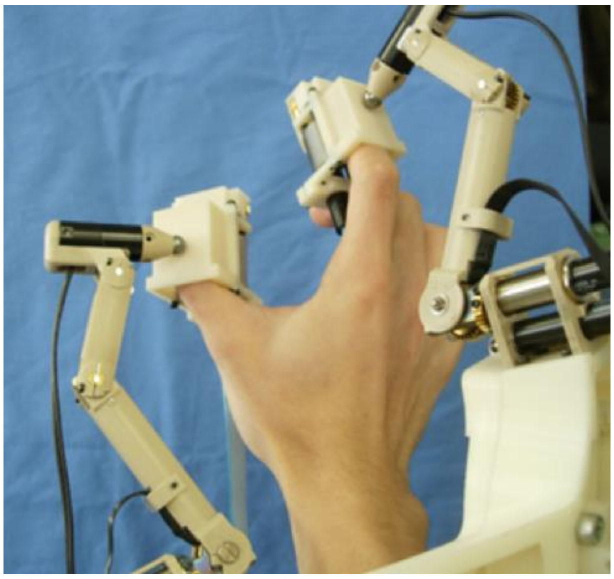


Fig. 31. Multi-finger tactile interface. A device that allows to simulate the tactile sensations of touching soft objects [53].

When organizing movement in virtual space, natural interfaces are widely used, based on fixing and recognizing movements of the entire human body or individual organs.
In the first period of the development of virtual reality environments, special suits were used to fix the movements of the legs. Special panels and platforms were actively used, steps and movements along which were associated with movements in virtual space. Real walking was proposed as a way to organize movement in a virtual environment. On the other hand, it is possible that movement in abstract virtual spaces is easier to organize through virtual flight. Recall that moving in a virtual space that is beyond the user's control can cause him unpleasant sensations, described by the concept of cyber-disease. Natural interfaces were developed based on the use of leg movements (Fig. 32, 33). In this case, the hands remained free.

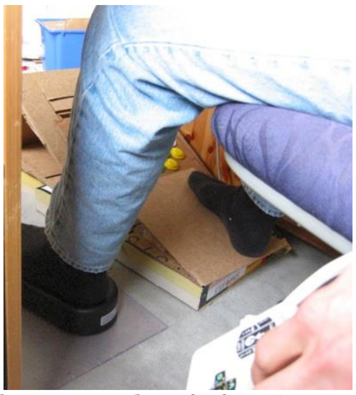


Fig. 32. The example of "foot" interface [54].

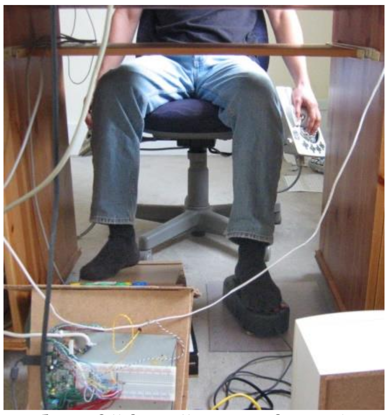


Fig. 33. The example of "foot" interface (another view) [54].

Multimodal natural interfaces were also developed, in which several methods of human-computer interaction are used at once as hand gestures, leg movements, gaze fixation, tactile interfaces.
Gesture interfaces can be divided into two types such as static, when a certain set of fixed hand positions is presented, and dynamic, when the system perceives and analyzes movements of the hands or other parts of the body. The analysis is based on motion capture, which is based on the methods used in the theory of image processing and belongs to the field of technical vision. Gesture interfaces can be used in medical applications, for example, to ensure the safety of physicians when examining a patient. Gesture interfaces are also used for control in virtual and augmented reality environments (Fig. 34).
When using sign interfaces, it is important to develop sign languages to assist users in their activities [55].

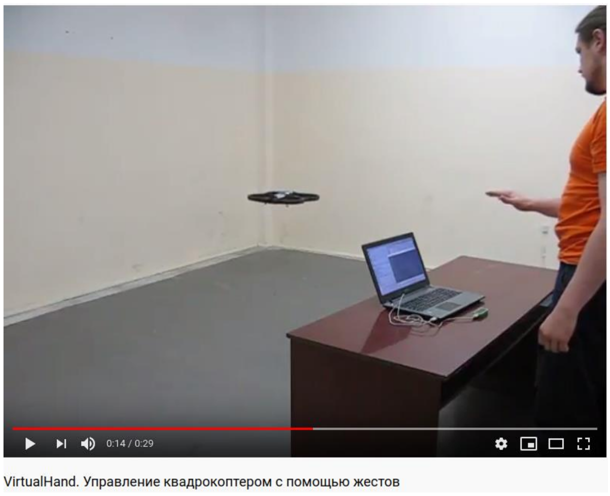


Fig. 34. Video frame of control of a quadcopter [56].

21. Activity-based approach to the interface design

When designing interfaces, a serious analysis of business aspects of future users is required. At the stage of "maintenance" and revision of the system, it is necessary to understand how the activities of users have changed after the computerization of their work. First of all, the theory of activity, developed in our country in the middle of the 20th century is associated with the names of A.N. Leontiev [57] and S.L. Rubinstein [58]. Publications suggesting the use of activity theory in the design and development of human-computer interfaces began to appear in the second half of the eighties - early nineties of the last century. The prominent Russian scientist V. P. Zinchenko [59, 60] should be noted as the pioneers of this direction.

An activity-based approach to interface design implies a serious analysis of the activities of future interface users. It is necessary to understand how the users' activities have changed after the computerization of their work at the stage of "maintenance" and refinement of the interaction system.

In connection with the activity approach, instrumental interfaces are considered. Instrumental interfaces are understood as interfaces for specialists in any field, who use them as a means for carrying out their professional activities, as well as mass interfaces-general-purpose tools, for example, for booking and ticketing systems, using banking, social and government services and etc. For such interfaces, the design approaches and quality criteria used for entertainment sites or social networks are not very applicable.

An activity-based approach to the design of human-computer interaction for a specific problem involves a deep study of the future users' work in a "pre-computer" version, an analysis of all emerging issues and a description of activities to solve them. It is important to identify the main goals and motives of this activity, describe the individual stages of the activity and identify of all entities that employees deal with. It also requires an "activity-based" analysis of the new situation that occurs after the computerization of work.

The quality criterion for instrumental professional interfaces can be evaluated through the number of people satisfied with the work of the institution for a given period of time. In other words, we consider the number of clients, buyers, patients, etc., who have received a satisfactory result and have not received serious stress. Stress can be measured both by direct methods for users of professional interfaces, by users' direct and indirect methods of mass interfaces and visitors to institutions. It can also be assumed that the stress level of the professional using this interface influences the user's stress level due to possible delays, disruptions in work and general irritation. The practice of real computerization gives us examples of good and bad results [61].

It seems that the activity-based approach to interface design has very great prospects. Such interfaces can become the basis for the development of modern convenient and reliable human-computer interaction environments in the most important areas of people's life and activity.

22. Bulk interfaces

Bulk interfaces will mean mass tool interfaces such as entertainment, social media, game interfaces and mobile device interfaces.

The class of mass instrumental interfaces includes interfaces intended, in particular, for booking and ticket purchasing systems, using medical, banking, social and government services and etc.

In the case of mass instrumental interfaces, the designer, by formulating the requirements for the interface, participates in the formation of future activities. The user cannot refuse to use the corresponding system, because through it he gains access to services, resources, information and so on that are important for his life . The mass instrumental interface should focus on the "weak" link, that is it should be successfully handled by a person with minimal capabilities for inputting, perceiving and analyzing information [61].

However (as mentioned above) the design methods and evaluation criteria applied for “consumption interfaces” are not applicable in cases of “instrumental” interfaces. "Entertainment" interfaces are not usually related to purposeful activities. Their developers set themselves the task of drawing users' attention to any incentives in order to achieve the desired response to customers. However, the activity-based approach is also applicable to the design of information and advertising sites, although their visit by the user is not an activity. The user acts as an object of the owner’s activity (placeholder, customer) website’s. It is for the owner that we can build both a goal and motivation and define actions and operations. However, the latter may not be defined very clearly, since the activity aimed at a person cannot be accurately programmed.
In many cases, one way or another users of modern interfaces require programming activities. Note the fact that the designers do not clearly describe (and perhaps even they do not realize) the rules, the "programming language" and the "virtual device" itself, which must be "programmed".
The next, already modern stage in the development of interfaces is connected with the Internet. Entertainment sites, e-commerce and e-service delivery sites have become an important source of interface design ideas. Interface quality ratings have also become largely associated with e-commerce performance ratings and the effectiveness of advertisements posted on websites. The effectiveness of the latter can be assessed primarily by the number of "clicks" on an advertising banner. It seems that this is an important reason for the dominance of the stimulus-response model in assessing the quality of interfaces (behavioristic in its essence).
Bulk interfaces can be evaluated based on usability criteria. In this case, subjective methods of assessing the quality of interfaces and sites are used based on a survey of a small number of users, which are combined with an analysis of the quality of design and its ergonomics. In addition to this, instrumental techniques for tracking the movement of the eye across the screen (eye tracking) are added.
Traditionally, developers talk a lot about friendly and intuitive interfaces. The system has an intuitive interface (intuitively usable or usable at an intuitive level), if the user's unconscious application of knowledge available to him leads to effective interaction with it [62].
We should pay special attention to the interfaces of computer games. Such games appeared in the seventies and they were initially used as text-based information. Then games using computer graphics began to develop actively. However, the graphics were quite primitive and the players did not pay attention to this, for example, the three-pixel men in the background, simply because they focused on what was happening in the foreground. In the eighties, millions of people around the world already played computer games.
The game’s interfaces influenced the developers of the visualization systems who borrowed from there a number of ideas (metaphors for visualization). Modern games have started to use virtual reality. At one time, it was claimed that such games would displace all others, but this has not happened so far and, probably,it will never happen. Note that game interfaces can use input devices such as gamepads and joysticks, as well as keyboards specially designed for "gamers". Game development experts believe that a good game interface is one that the player does not notice. However, sometimes the complexity of interfaces can be part of the gameplay (the component of the game that is responsible for the interaction between the game and the player). Games on mobile devices (as well as the use of mobile devices in general) also have a large impact on users and, therefore, influence the development of mass interfaces. The interaction techniques that users have been familiar with since childhood should be taken into account when creating both mass and specialized interfaces.

23. Conclusion

The history of modern computing is almost 80 years. New computer technologies are constantly emerging, for example, cloud computing, which gives access to configurable computing resources, or 3D printing, which allows you to create ready-made parts of complex machines, as well as "print" fragments of human organs and tissues on a cellular basis. Now everyone is hearing about quantum computing, which uses quantum mechanical phenomena to perform computations. Due to the development of modern computing, the task of processing and analyzing "big data" has become in demand. Modern approaches to creating artificial intelligence based on neural networks have gained immense popularity.

The question of human-computer interaction is still relevant. This is due to the fact that if in the first two decades of its development, computers were used by thousands of programmers around the world, today they are used in one form or another by billions of people, including millions of specialists, programmers of various levels and IT specialists, scientists, doctors, engineers, teachers, bank employees, office workers etc.

One of the approaches to ensuring more efficient human-computer interaction is the use of natural languages. Moreover, in the sixties and seventies, programming tools based on natural languages were even developed. However, due to the large volume and ambiguity of natural language dictionaries and their complex syntax, it is very difficult to provide accurate recognition of natural texts.

In 1978, academician A.P. Ershov suggested using a subset of the natural Russian language - " clerical Russian", the language of reports, statements and questionnaires as the basis of the language of interaction with a computer. In his first report on this topic, A.P. Ershov used the invention of K.I. Chukovsky, the word "cancelyarit", and then used the more scientific term "language of business prose". This language is characterized by a relatively small vocabulary (about three thousand words), rather precisely with fairly well- defined syntax and semantics. In other words, the words and concepts of "clerk" have precisely defined meanings, and the phrases of reports and statements are written according to the established scheme. The language of business prose is natural for many specialists in the field of management, and in this area, a fairly effective interaction of a person with a computer on its basis is possible. In this case, the implementation of texts' recognition with a limited vocabulary and clearly defined syntax and semantics is greatly simplified [63, 64]. Work in this direction was carried out until the nineties. However, the development of modern information technologies in the field of business management has reduced the relevance of these developments.

At the same time, natural human-computer interaction remains an important element of modern computing. For example, much attention is now being paid to voice interaction with a computer using elements of natural language.

It seems that the naturalness of interaction should be linked to the main activities of future users, taking into account their experience in using information technologies. For example, it would be useful for design engineers and designers of complex equipment to combine design automation tools (CAD), mathematical modeling packages using the resources of modern supercomputers (for example, Logos [65]), as well as visualization tools based on virtual reality in a single interactive system. Prototypes of such systems are already being developed, although the creation of full-fledged design environments is still a promising goal. In this case, the development of the correct interface and methods for the visual presentation of design objects becomes an important task.

The development of "correctness" criteria requires the additional analysis. Experience shows that optimized user loading can be a negative factor, for example, for surgeon's workstation interfaces. In this case, the surgeon may become overloaded with interface management during the operation, as a result of which less attention is paid to the operation progress and the patient's condition.

And more about interfaces. Modern device control methods make it possible to control vehicles using buttons and touch screens, but will hundreds of millions of drivers abandon traditional steering wheels, pedals and gear knobs? In the case of interfaces for full-fledged computer control of complex technical objects, it is necessary to take into account the functioning of these objects, as well as the tasks of their control. Interface design requires serious knowledge from the developer in various fields.

One comment needs to be made. In terms of the main fields of activity related to computing, the main focus is on information technology. Of course, information technology is an important component of the modern world. However, it must be remembered that in industry, production technologies are secondary in relation to the design of equipment and certainly in relation to research and experimental development. In the field of computing for our country, the development of computer science should come first and technologies should be the result of our own research and experimental development in the field of software.
In this case, we will be able to achieve the technological independence and not follow the new western technologies.
The development of computer science requires the training of scientific personnel and serious efforts in maintaining existing and creating new scientific schools. Computer science education should include mathematical disciplines related to both continuous and discrete mathematics, as well as serious knowledge in various areas of software along with gaining experience in real development. The same knowledge is required for software developers for human-computer interaction and visualization. For example, mastering the elements of photorealistic graphics requires knowledge in such a mathematical discipline as equations of mathematical physics, and for serious developments in system programming, knowledge in discrete mathematics and mathematical logic is required. When developing specialized systems for scientific imaging or imaging for medical purposes, it is necessary to listen to short courses of lectures on the relevant topic. Knowledge of computer psychology or semiotics, the science of sign systems may also be necessary.
Developers must have a wide range of scientific background to be ready to grasp new ideas. Thus, not only scientists in the field of computer science, but also the developers of software systems intended for scientific computing, along with knowledge and skills in information technology, should have a very university scientific background, possibly with further specialization in specific sections.
And the last thing. At the dawn of the computer era, the father of cybernetics Norbert Wiener warned of the possibility of a new technology for those times leaving the control of its creators. Wiener spoke about William Jacobs' fantastic story "Monkey Paw", which tells about a talisman in the form of an enchanted monkey's paw, which can grant wishes, and , moreover, they are fulfilled in such a way that they bring terrible misfortune to the owner. At the same time, formally, the desires are fulfilled with accuracy, although not at all in the way the “customer” can imagine [66].
Norbert Wiener feared that computers without knowledge of human values and priorities would spiral out of control and bring misery to humanity like a monkey's paw. Then there was a lot of talk about whether computers can think and whether they will try to conquer humanity. Then there was a lot of talk about whether computers could think and whether they would try to conquer humanity. But even then, and even in the seventies, computers were under the complete control of programmers, therefore despite the development of the mathematical field of "artificial intelligence", Wiener's warnings seemed redundant. Today, there is a lot of talk again about artificial intelligence, which is implemented on the basis of neural networks. They say that computers and robots will replace humans in almost all areas of activity. It's time to remember the science fiction books of the fifties. First of all, the three laws of robotics, which were invented by the remarkable American writer Isaac Asimov [67].
These laws describe the ethics of interaction between humans and robots with artificial intelligence. The main goal of the laws, enshrined in the very design of the robots' electronic brain, is to prevent harm to humans, including those caused by the order of this or another person. In modern descriptions of artificial intelligence, little attention is paid to this issue. Using artificial intelligence without analyzing possible dangers can lead to serious problems. To overcome the effect of the monkey's paw, that is the soulless execution of algorithms and commands, it is necessary to describe ethics, as well as knowledge regarding human values. In other words, large-scale research and development is required to implement real artificial intelligence.

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. Vojskunskij 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/vxHkXaKkw8cHYj5o7, (Accessed August 20, 2020).
14. Première phase d'avancement : la fonction vectorielle, http://jeanpierre.rousset.free.fr/Informatique/Noyau_Graphique/preambule_comments.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 programmirovaniyu. / 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 graficheskoj 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 programmnogo 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/f00/s06/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/n6yrdswz.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., SHCHerbinin 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-pozvolyayushhee-imitirovat-taktilnyie-oshhushheniya-prikosnoveniya-k-myagkim-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 // MMIInteraktiv, 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 mashinoj 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.