JPRS-UCC-91-001
30 APRIL 1991

JPRS Report

Science & Technology

USSR: Computers

19980508 117

DTIC QUALITY INSPECTED 3

Science & Technology
USSR: Computers

CONTENTS

General

Software

Applications

Networks

Theory of Computation

Development Strategies of GSNTI and GASNTI in 1991-1995

[Article by S. S. Tereshchenko]

UDC 002.63

[Abstract] Like all other sectors of the USSR economy, the state system of scientific-technical information is in a state of crisis. Various teams of scholars and specialists are working to remedy the situation. Specifically, the status of both the State System for Scientific and Technical Information [GSNTI] and State Automated System for Scientific and Technical Information [GASNTI] have been assessed, and three possible strategies for their further development have been proposed. According to the first strategy, the USSR State Committee for Science and Technology would continue to develop the GSNTI and GASNTI in the manner that has been traditional for five-year programs. Without any expansion during the five-year-plan in question, the cost of all operations under this development strategy would be about 50 million rubles. If the costs of expanding existing data base capabilities to include full-text and hypertext data bases are factored in, the cost of developing the GSNTI and GASNTI in 1991-1995 would range from 150 to 200 million rubles. According to the second proposed strategy, which has been termed the "analytic" strategy, the USSR State Committee for Science and Technology would continue to develop an analytical information system to provide data base organization and support for scientific-technical progress. The difference would be that the information technology would be reoriented toward the preparation of new or "comparatively new" information products and services, including the provision of diverse analytical materials, forecasts, research on trends, and information support for the processes used to solve newly discovered problems. According to the third strategy (termed the "active interaction" strategy), the GSNTI and GASNTI would be concerned with developing all of the services and products stipulated in the "analytic" strategy but would also engage in active partnerships with developers of data bases and data banks. This strategy calls for engaging all of the country's information services and scientific-technical information systems in a cooperative effort to "informationalize" the country. Specific recommendations are presented for developing, financing, and implementing these strategies. Also presented are specific proposals for designing and developing a data base and data bank system that will function as one of a series of complex information systems constituting the country's information market.

Prospects for the Development of a Government System of Scientific and Technical Information. Abridged Account of the Meetings and Seminar of Specialists, 2-5 April 1990, Vladimir

[Article, Unattributed author]

UDC [002.63GASNTI:001.18]:061.3

[Abstract] This three-day conference was organized by the Center for Scientific and Technical Information on Electrification. The intent was to discuss the development of the State System of Scientific and Technical Information, and whether it should continue to exist. It was attended by 22 representatives of various information agencies and centers. Discussions on the functions of the system, its structure, financial support and division of appropriations, fees for services, control, degree of centralization, and direction of development revealed that the specialists held vastly different views. No consensus could be reached on any of the main topics. A group was formed to draw up a plan for the program. The article presents the texts and the question and answer periods following a number of the presentations.

Restructuring of Information Services Based on Marketing Methodology

[Article by B. M. Gerasimov and V. D. Shkardun]

UDC 002

[Abstract] Entry into the foreign market has been a perplexing situation for Soviet enterprises. Buyer's market conditions are a new phenomenon for these enterprises, who must now adopt new strategies to compete effectively. They now turn to western marketing practices. Marketing is now a necessity for firms doing business abroad, but will inevitably become necessary for firms involved in only the domestic market. It is explained that marketing involves the prediction of market trends and the active measures taken to stimulate the market to shift in the desired direction. It is indicated that marketing is not imposed by decisions from above, but may be dictated by consumer trends. The article emphasizes the importance of the exchange of information in marketing, not only the receipt of information from outside the firm (about competitors, etc.), but also the dissemination and exchange of information within the firm, and the provision of product information to the market. Information is as much a product as the product

General

itself. Existing information infrastructures in firms must be adapted to the tasks of marketing. Guidelines are provided for introducing marketing practices in enterprises. References 8: 7 Russian, 1 Western.

The Game That Created a Monster

917G0036A Moscow TEKHNIKA - MOLODEZHI
in Russian No 11, Nov 90 pp 30-31

[Article by Alexander Shedrin, student at the Moscow Automotive Roadway Institute. First paragraph is introductory paragraph in source.]

[Text] In the second issue of 1989 we published a survey article on computer viruses. A year went by and during that time the editors obtained their own computers, and on the basis of their experience they became convinced of the importance and timeliness of any information on computer viruses that the beginning user might encounter in everyday work. Even if we had not received many letters with requests to continue the discussion we would certainly have returned to this theme. However, in this case our interests and plans coincided completely with the consensus of our readers. And so we present the latest material on computer viruses.

For every action there is an equal and opposite reaction. This is Newton's law, which we remember from school. Certainly it is valid not only for mechanics but also computer science. The widespread introduction of personal computers brought with it the problem of virus programs. Since 1988 this problem has become urgent in our country.

If one replaces the organism with a computer and cells with programs, then one can draw an analogy between biology and computer science. The computer virus invents nothing new. When it enters a computer with an infected program, the virus intercepts the computer's control over it, and using the computer's resources (memory) and energy the virus begins to multiply without limit, destroying the "host" and threatening other "cells". This analogy has become so much a part of computer science that computer slang has also incorporated such purely medical terms as "infection", "incubation period", and "vaccine". This has also led to some misunderstandings, for example, when we had just begun to introduce computer equipment, there were absurd rumors about computers from abroad that spread infection (in the direct sense).

In the opinion of several specialists, the theoretical basis for the ability for computer viruses to appear can be sought in the works of John von Neumann. In 1949 he was interested in the problems of the increase in intellectual power and the development of computer equipment. This includes the possibility of self-replication of computers or individual commands.

Let us say that, due to some improbable error in which the program breaks down, the computer forms some command which is capable of destroying other programs while remaining undamaged, then due to some other system error, it acquires the ability to replicate itself. The probability of such a chain reaction occurring is nonzero, especially as computers become more complex; however, at the present stage of development of computing equipment, it is virtually impossible. For von Neumann the theoretical construct of a "virus that is the product of the operation of the computer itself" was only a mental game. Today's computer virus has completely human parents. There is but one coincidence: it actually appeared as the result of a diversion among professionals.

In the early 1960s, three young programmers working at Bell Laboratories, M. Douglas McIlroy, Victor Visotsky and Robert Morris thought up the computer game Darwin, in which, according to the law of natural selection, the strongest survived. A sequence of commands was created which used all the remaining programs in the RAM of the computer as a data base. In sum, the RAM was turned into a battlefield between programs trying to do whatever would erase or damage the code of its opponent.

In time the game became more complex and came to be called Core Wars. At that time the RAM of the computer consisted of tens of thousands of ferromagnetic cores which stored one bit of information. Indefatigable programmers even managed to get at the information carrier.

However, due to the absence of a wide market of personal computers at the time, these games did not spread. The first report about them appeared only in 1972 in a specialized journal. Their current notoriety came in 1984 from A. K. Dewdney, who writes the Computer Recreations column in the journal Scientific American. Dewdney himself thought that the idea for this game arose due to the influence of anecdotal legends about one vindictive programmer. To get revenge on his boss he wrote and started a program, Creeper, which made an exact copy of itself every time it was executed. Soon Creeper had grown so much that other programs simply had no room in the RAM of the laboratory computer. To destroy Creeper it was necessary to quickly create another program, Reaper.

After this publication two young programmers from Italy sent Dewdney a detailed description of a virus program which was capable of infecting a computer (which the readers immediately found out about). At programmer conferences the codes of new virus programs were demonstrated, and computer journals reported about them in detailed articles, not suspecting that they were helping to spread the epidemic: they did not yet think that computers had already begun to fall ill.

From the personal disks of users and programmers the viruses entered information networks, infecting hundreds of computers. Among the propagating viruses were some completely harmless ones, issuing Christmas greetings to the stunned users (internal IBM network,

December 1988) or appealing for the preservation of peace on Earth (a slogan appeared on the screens of thousands of Macintosh computers in March 1988, on the anniversary of the appearance of this personal computer model on the market). There were also extremely dangerous ones, for example, the Worm of Robert Morris Jr., son of one of the creators of Darwin. The Worm paralyzed six computer networks in the USA.

Now that the history of the generation of computer infections is more or less clear, let us try to classify some of the viruses which are encountered in the Soviet software and computer games market.

The Worm is encountered independently in different programs, and causes various side effects. It is a small program which is capable of independent existence and propagation.

The Trojan Horse is a program which has a normal exterior, but carries within it a defense unit which causes unpleasant consequences when the user takes actions which are not sanctioned in the commercial contract. As a rule, many companies use these programs to protect their copyrights, and the programs begin to work after several copies are made of the basic program.

The Bomb is a variety of Trojan Horse designed to destroy data bases. A classic example is the virus RCK-VIDEO, which creates on the monitor a portrait of the rock star Madonna and simultaneously erases the bulk of the information from the hard disk.

The Time Bomb is a variety of the Bomb. It goes into effect after a certain time interval or event. Among the time bombs are the infamous Friday the 13th virus, the Lehigh virus, and others. From the very name of the program it is clear that it is triggered by one event known only to it: for example, a Friday the 13th. The Lehigh virus is especially interesting. It begins its destructive work only after four system programs are infected.

Medicine fights bacilli using vaccines and medicines; programmers always have in store the most radical method: to completely erase the information from the hard disk and reformat it. But the most simple solution is the most expensive, especially if there are no backup copies. Thus, mathematicians had to follow the lead of doctors by creating medical programs. The majority of antiviral programs are specialized for a specific studied enemy and have no reaction to other computer parasites. The attempt to create a panacea for all diseases has been unsuccessful; the universal cure is not as effective as the specialized ones now making the main assault in the virus war. This approach does not promise a rapid victory; the computer is defenseless against any new infection. It is clear that it is necessary to bolster the computer's health.

This course, which is suggested by nature, has been taken by the American firm Computer Security Industries.

Recently, the firm presented users with the first computer which is reliably defended against viruses, and it is called the Immune System.

But one can avoid viruses in another way. Probably everyone knows this saying from childhood: "Wash your hands before you eat". The rule on hygiene plays an important role in computer science as well. Here are some practical recommendations for the beginning user:

—Before you start any program that has been sent to you, check it with an antiviral facility;

—Put a write protect tab on any diskette from which you only read information;

—Do not use programs whose behavior you do not understand;

—Give all command files with the extension .EXE and .COM the read only attribute using the standard operating system command;

—If you observe a virus in the computer, warn all your colleagues;

—You must check all new games before the first boot.

Finally, as food for thought, we present a report from the newspaper Tszintszi Zhibao (25 May 89). The translation was sent to this journal by the Chinese journalist Mo Gunmin.

13 April 1990, Friday. The computers at factories, scientific research institutes, and government agencies in China were halted by the Black Friday virus...

The first case of infection of Chinese computers by viruses was reported in April 1989, when the computers of the Sian aluminum factory were infected by the Chlorella virus. In a year these viruses had spread through the entire country with stunning speed. The situation acquired such a dramatic character that at the end of 1989 it was necessary to do a prophylactic check of 12,750 computers. It was then established that almost one fifth of them were infected with the Chlorella virus. Specialists presume that no less than 6000 computers in China are infected.

According to the calculations of the state's security agencies, no less than 18 types of computer viruses are circulating in the country, seven of which are of foreign origin. Along with Chlorella (another name is Spot), the Jute virus is spreading (the New Zealand virus), which is a damaging virus, the Pakistani, the Rain Drop (known in America as Cascade), musical viruses which play melodies at exactly 5:00 p.m., and others.

The appearance of computer viruses, which threaten information streams in China, caused a great anxiety in the government. In July 1989 the State Council of the People's Republic of China commissioned the Ministry of State Security to head the fight against computer infection.

USSR Develops 'Universal' Program Protecting Against Computer Viruses

*917G0042A Moscow IZVESTIYA in Russian No 28,
2 Feb 91 (signed to press 2 Feb 91) p 7*

[Article by S. Leskov: "Nets for the 'Black Avenger':
Domestic Program To Combat Computer Viruses
Developed"—first paragraph is boldface IZVESTIYA
introduction]

[Text] Searching for computer viruses that spread with
unprecedented speed is becoming one of the most impor-
tant problems of the modern information society.
IZVESTIYA (issue No 4 of this year, "The 'Black
Avenger' is Elusive") published a sampling of commu-
nications by our correspondents from Washington,
Bonn, and Sofia regarding attempts to "hunt" computer
viruses. Prof. Klaus Braunstein, the director of the
computer virus center at Hamburg University, reported
that according to his information, a universal antivirus
program has been created in the USSR and that there is
virtually no information about it in the West. He has
succeeded in finding the creators of the Soviet antivirus
program, which is called Strazh [guardsman].

It is above all necessary to clear up any misunder-
standing that may have arisen regarding the possibility
that a universal antivirus program, which the best minds
of the largest scientific centers throughout the world
have struggled unsuccessfully to create, could have been
developed in the USSR, which is far from the high roads
of world computerization. To this day, unified computer
networks, without which the economies of the developed
countries would be unthinkable, have remained some-
thing semimythical for us. And indeed, as is known from
world practice, it is through such unified networks that a
virus "infection" destroying computer programs
spreads. The yearly losses from computer crime in the
United States are estimated at 100 billion dollars, and
those in the countries of Western Europe are estimated
at 30 billion dollars. In 1988 alone a commercial com-
puter virus association recorded 90,000 virus attacks on
personal computers. In reality, the number of incidents
is certainly much larger inasmuch as many firms care-
fully hide such incidents out of fear for their reputation.
In our country exchange among programmers occurs by
unregulated exchange of diskettes. Can there be any
serious discussion of any appreciable harm, which is to
say, any urgency to the problem in the USSR?

In the opinion of those hunting the computer "infec-
tion", however, viruses written on floppy diskettes prop-
agate no less slowly than do those living in networks.

They are, however, much harder to detect and neu-
tralize. Another problem is that the antivirus problem
has clearly received insufficient attention in the USSR.
This is partly related to the near-beggar's wages of even
the most qualified programmers. It is simpler for a
mathematician to obtain an extra premium for a pro-
gram rewritten to replace an infected program than it is
to struggle over a complex problem. As a result, cases
such as the "logic bomb", which stopped the main
conveyer and cost at least 1 million rubles when it was
triggered in the automated process control system of the
Volga Automotive Plant, are hardly a rarity. It must be
said that the symptoms of a virus epidemic were only
detected in the USSR in 1988. By last summer, however,
32 different viruses had been identified. Only two-tenths
of them have been brought in from abroad. To all
appearances, the remainder have a domestic origin.

The antivirus program mentioned by the German
scholar was created by a collective of mathematicians
who joined together to form the small enterprise Alkor.
The inspiration for the idea came from programmer
Aleksandr Gutnikov. What is the fundamental differ-
ence between the Strazh program and other "traps" for
viruses? Antivirus programs may be divided into three
groups. The first and largest is that of programs directed
toward neutralizing already recognized viruses. The
second approach entails regular monitoring of files' sizes
so that a virus that has been built into a program can be
found in time. Finally, the third group is based on
intercepting all writing operations in files being exe-
cuted, which makes their operation tiresome and cum-
bersome. Not one of these methods can provide reliable
protection against viruses, however.

The authors of the Strazh program approached the
problem of protecting against viruses from another side.
Any virus, even the most cunning, trying to make a
change in a program must be constructed so that sooner
or later control is switched over to it. This switching may
be identified and used as the basis for finding the "body"
of the insidious virus. The situation is made easier by the
fact that the methods used by viruses to switch control
over to themselves are far fewer than are the number of
viruses themselves. During a check it was discovered
that the vigilant Strazh did not let one of the types of
computer viruses circulating in our country get through.

One can be happy for the success of the Soviet program-
mers who recently sent the Strazh program for expert
evaluation to so strict an examiner as the firm Olivetti.
But no matter how perfect the nets to catch viruses, the
fight must be waged not only against them but also
against their creators. Unfortunately, in the absence of
laws protecting intellectual property, computer hooli-
ganism poses absolutely no danger to those who are
inflicting enormous losses upon society. For comparison,
two years ago a U.S. judge sentenced the programmer

D. Burleson, who decided to use a computer virus to seek revenge on his firm, to seven years probation and to a fine of $11,800.

Finally, the beggar's wages of programmers, even given the strictest legislation, make instances of slovenliness and a lack of conscientiousness inevitable. The remuneration for the work of programmers and mathematicians who, in an informationalized society, have direct access to controlling the economic life of entire sectors, should be commensurate with the importance of their labor.

Window System for YeS Operating System and SVM YeS Interactive Processing Subsystem

[Article by A. Yu. Tandoyev and A. I. Tikhonov]

UDC 681.3.06:51

[Abstract] Multiple-window interaction is the current de facto standard for much of the interactive software currently being developed throughout the world. A series of tools that make it possible to use the multiple-window approach with YeS computers has recently appeared (including Natural 2.1 and VM/SP 5.1); however, no tools exist for developing multiple-window interaction for popular programming languages. The system developed by the authors of this article, i.e., A-Windows, is intended to fill this gap. The system operates in different environments, including the interactive processing subsystem SVM YeS, time-share systems, the interactive monitor PRIMUS, and the CICS teleprocessing system. The new system includes (1) an interactive window-creating tool with a set of window presentations in the file system and (2) an integrated set of window management subroutines for a variety of popular programming languages, including PL/1, Fortran/VS, Assembler, and Rexx. The new system is a powerful tool for creating interactive software for YeS computers, and applications programs with an A-Windows interface provide a new quality of user interaction. Such programs offer the following advantages: ease of implementing modern forms of interaction (such as multilevel menus and context-dependent help), independence of the environment in which the forms of interaction are executed, and the presence of tools that make it possible to conduct the process of designing interaction parallel with the software development process. References 2: 1 Russian, 1 Western.

Functions and Features of Executive Operating System for a Control Fault-Tolerant Multiprocessor

[Article by S. Ya. Vilenkin and I. V. Shagayev]

UDC 681.3

[Abstract] This article describes a new control fault-tolerant multiprocessor system with one stream of instructions and multiple data streams. The system is organized in a three-level structure: a system control level, an instruction level, and a decision field level. A direct memory access channel provides an information link between the levels. At any moment in time the system control operates with at least three processors, one of which is the driving processor and the others being driven processors. The system control is capable of self-diagnosis, and it functions as the operating system. Two elements of the control fault-tolerant multiprocessor system's executive operating system, i.e., reconfiguration and restoration monitors, provide software control of the hardware's fault tolerance. The control fault-tolerant multiprocessor system, together with its executive operating system, provides users with extensive capabilities from the standpoint of designing the highly productive and highly reliable hardware configurations required to perform tasks involving parallel data. Owing to the virtual nature of its hardware, the control fault-tolerant multiprocessor system and its reconfiguration tools make it possible to avoid reconfiguring the software in the event of most hardware failures and to avoid "building" mechanisms of servicing exceptional situations caused by hardware failures into the make-up of programs, which makes programs much easier to analyze in the debugging stage. The executive operating system's restoration hierarchy provides a rather wide range of tools for smooth reaction to hardware malfunctions of practically any duration and extent. The fact that the control fault-tolerant multiprocessor system contains a special hardware level supporting the executive operating system makes it possible to implement a broad set of utilities and specific functions without a loss of productivity in the main portion of the hardware (the decision field and decision field level). Performing the main functions of diagnosis, monitoring the configuration, filling and analyzing the system log, restoring the system, and repeated reloading alternative modules do not require additional equipment in the form of a special diagnostic process. One difference between other existing operating systems and the executive operating system used in the control fault-tolerant multiprocessor system described here is that the latter does not afford the capability of reacting to exceptional situations resulting from hardware failures. Since the executive operating system and control fault-tolerant multiprocessor system have been developed for use in controlling rapidly occurring production processes, in express analysis of the status of control objects, and in processing incoming data, no provisions have been made to allow service personnel to control the system's reconfiguration. They can only replace failed components. Figures 5; references 8: 6 Russian, 2 Western.

Software Emulator of Functions of Epson-Type Printer on Neyron PC

[Article by V. M. Vasilets, V. V. Gusev, candidate of physical and mathematical sciences, and I. L. Dayen, candidate of technical sciences]

[Abstract] Many graphics packages are designed to output graphics to Epson-type printers. The Neyron PC is equipped with a D-100 printer. A software emulator of the functions of Epson-type printers has been developed

to make it possible to output graphics for use with Epson-type printers on the Neyron PC. The driver simulating the graphics modes of the Epson printer has been implemented as a program interrupting the basic I/O system responsible for controlling printing (with the hexadecimal number 17). The main problem that had to be overcome in developing the driver was that of converting eight-pin graphics into seven-pin graphics. To accomplish this, a part of each graphic byte sent to be printed out on the D-100 printer will be printed out immediately, and part will be written into the buffer, which contains seven lines of sufficient width calculated from the maximum possible line length. The buffer is dumped by simulating the command for vertical release on the Epson printer. The algorithm used provides maximum preservation of the drawing's initial vertical and horizontal dimensions. The driver is resident, and protection against restarts is provided. Tables 1.

Software to Automate Production of Electronic Equipment in Instrument-Building Sector

[Article by V. V. Zhukov and M. I. Smirnov, candidate of technical sciences]

[Abstract] Production automation in the instrument-building sector is becoming increasingly important throughout the world's developed countries. This is particularly true in the case of the production of small batches of different electronic products, which entails dealing with a large volume of information and the necessity of efficiently combining the tasks of information processing and real-time control of high-speed equipment. This article proposes a number of software products designed for use in developing flexible manufacturing systems [FMS] to automate the process of producing small batches of different electronic products. All are distinguished by the fact that they make use of the characterization approach to solving the optimization problems arising during the information processing and control process. The first of these software products is the Upravleniye-GPS [FMS Control] system. It is designed to automate the creation of software for local control devices. It can be used to develop software for individual pieces of production or auxiliary equipment, robot systems, flexible production modules, and devices for group control of flexible manufacturing sections. Another new product, the Lotos system, is designed to control all aspects of a flexible manufacturing shop. It contains modules to control the flexible manufacturing process itself and the transport/distribution and storage units of FMS. A third product, the PROZA system, is an accounting system to keep track of warehousing in an instrument-building shop. It provides shop personnel with the information needed to make decisions regarding scheduling when the production of small-series products is to be started, and unlike existing programs, it

can be used with minicomputers. These software products have been introduced at many industrial enterprises. They have proved their effectiveness in automating instrument-building and have resulted in an annual savings of half a million rubles. Work is currently in progress to transfer the aforementioned software to IBM PC/AT-type computers. References 4: Russian.

Software System for Unified Hardware-Software System for Machining FCMS

[Article by I. M. Levin, A. V. Kakraynikov, and N. I. Fedchenko]

[Abstract] The introduction of machining flexible computerized manufacturing systems [FCMS] in different sectors of industry has placed increasing emphasis on the creation of FCMS ASU providing the set of control functions needed to organize production with a small number of workers. One way of accomplishing this is to create (on the basis of SM computers) a unified hardware-software system to function as the nucleus of an FCMS ASU representing a set of design decisions and software and hardware to perform a specified set of tasks entailed in controlling the sections and shops of a machining FCMS. This article describes one such unified hardware-software system that includes the following interacting subsystems: scheduling production operations, dispatching and accounting, and controlling production equipment. The FCMS unified hardware-software system is intended for use in a standard flexible machining section that includes such items as a set of machine tool modules, a rack-type automated warehousing system, a robocar-based automatic transport system, and a section (module) for preparing workpieces and tools and for washing and inspecting workpieces. The most important component of the unified hardware-software system is its software system, which includes system and developed functional software that permits real-time control; it is modular in nature and can be adapted to specific conditions given a specific class of constraints, it is mobile and functions reliably for an extended period of time, and can operate in automatic and automated control modes. Shop personnel can interact with the system for purposes of coordination, accounting, and diagnosis. It is anticipated that introducing the unified hardware-software system will reduce the labor intensity, time, and costs entailed in developing ASU for machining FCMS during the design and introduction stages. Figures 1.

Structure of a High-Performance Microprocessor With Hardware Support

[Article by V. M. Pentkovskiy, A. I. Zaytsev, S. S. Kovalenko, V. M. Feldman]

[Abstract] This microprocessor is part of the Elektronika El-90 series and is compatible with Elbrus computers. To increase productivity it combines the speed of the Elbrus 2 with the simple control of FPS computers and RISC processors. The system and application programs in the Elbrus were completely translated to a high-level language so a new command system and processor system could be developed. Two VSI integrated circuits are used: one for the central processor and one to control the buffer (cache) and virtual memories. The microprocessor supports the langauges El 76, FORTRAN, PASCAL, C, LISP and Smalltalk. The command set consists of 80 commands. There is a single 32-bit format for 3-address commands, and the format of the operands and the result is statically defined in the command. There are six address modes. Commands can be carried out in parallel due to the use of optimized compilers and automatic code compacting methods. Cache memory commands were organized to reduce interference between procedure commands, and procedure transition commands were organized so that they made only one call to memory. Built-in subroutines to organize memory can be found in a special ROM in the central processor. Figures 2; references 13: 5 Russian, 8 Western.

Compression of Tracing Information in Real Time Automated Control Systems

917G0024B Moscow PROGRAMMIROVANIYE in Russian No 3, May-Jun 90 (manuscript received 4 Aug 88) pp 44-53

[Article by R. I. Vlasenko]

UDC 681.3.06

[Abstract] An approach to the selection of a tracing information structure and a simulation model is provided which substantially reduces the volume of information. This approach is based on an analysis of the schematic of a real time automated control system. A great deal of the tracing information recorded when a program is being debugged is unnecessary, the result of processing of initial information. To find errors in the software one must choose a tracing information structure and model which reduce the volume of information while providing effective error detection and localization. Here the automated control system is considered a finite determined automaton and the trace information is the set of input signals of this automaton. This reduces the volume of information and enables one to use statistical data compression methods. It is shown that by expanding the volume of the model gradually, one can significantly reduce the volume of trace information for the automated control system. The volume reduction may be up to two orders of magnitude. More than 90 percent of the errors manifested in the automated control system can be detected and localized on the first run of the simulation. References 12

Package of Graphics Programs for Express Analysis of Information

917G0024C Moscow PROGRAMMIROVANIYE in Russian No 3, May-Jun 90 (manuscript received 7 Mar 89) pp 64-66

[Article by R. M. Vasilev, A. A. Mozelev, and A. M. Tarasov]

UDC 681.3.01

[Abstract] This package of graphics programs is intended for the color display of two- and three-dimensional graphics. It uses the RAFOS-TS operating system on an SM-1420 omputer. The color monitor driver is called from a FORTRAN program. The driver is in Assembler. Direct access to the driver is also possible for express display of graphics information on the screen. The monitor displays up to seven colors. The operating memory for one task is limited to 56 kilobytes, so the user can only choose one action from the menu at a time. The data analysis program is written in FORTRAN and can be easily modified and expanded. The user should be familiar with FORTRAN and the RAFOS operating system. Only seven commands are needed to use the package. The user then enters information on the file and the desired quantization. The package provides a sufficient level of automation, ease of use, rapid output of graphic images (two-dimensional, less than a second, three-dimensional, several seconds), which compensate for its insufficient power. It is a convenient and effective graphics package for the SM computer. Figures 1; references 2.

Current State and Direction of Development of Data Base Management Systems With an INTEREAL Relational Type Interface in Systems to Process and Use Information

917G0025A Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 9, Apr 90 (manuscript received 9 Feb 90) pp 48-54

[Article by I. M. Aberemkov, I. A. Boychenko, A. V. Peshkov, and N. I. Fedchenko]

UDC 681.3.06

[Abstract] INTEREAL is a portable relational data base management system with a relational interface which can be used in various types of computer architectures and operating system environments. The system currently supports a wide range of Soviet computers and the IBM PC, and MS DOS, VMS, and other Soviet operating systems. The system allows multiuser and multitask access to the databases. The databases can be accessed by programs in any language supported by the operating system. QUEL and QBF language constructions may be used to manipulate data. The user can generate menus and forms, perform multilevel sorting, and data compression, as well as print results. There is an expanded

help system, and data can be converted to the structure of other DBMSs. Up to 32,767 two-dimensional relation tables can be supported, each table containing up to 255 columns and 65,535 lines. Data may be indexed, and the indices stored. The system has been user tested and business applications have been developed for the system. Users can edit text, work with templates or graphs, and manipulate data. Language support enables the user to design his own applications. Data is stored in inverted lists and B-trees. The system occupies no less than 256 kilobytes on the SM-1420 and 640 kilobytes on the SM-1810. Figures 1.

Basic Principles for the Development of Expert Systems in the Area of Data Bases

[Article by O. V. Golovanov and V. N. Smirnov]

UDC 681.3.06

[Abstract] This article examines the basic principles of developing hybrid expert systems for data base design. The experience gained in designing data bases can now be used to automate various stages of the process. Expert systems are intended to bridge the gap betwen theory (of information search or systems analysis) and the end user. This system includes a data base design expert system and an information search expert system. In the data base design expert system there are three levels of expert evaluation of the design: system architecture, data organization, and programs. The expert system makes recommendations to designers on how to avoid or eliminate errors or bottlenecks, and advises users on the viability of the project, advantages and drawbacks in the project, etc. The data base design expert system has two parts. The first consists of a data structure to which the user inputs the desired characteristics of the design and associated values. The second part is knowledge on data bases. Examples are given of the algorithms used to evaluate data base structures or to formulate a data search. The information search expert system helps the user find the desired information. This is used in the case where the user knows what he wants, but does not know where or how to find the information he needs. The information search expert system gathers data in the interaction with the user on the subject field and the skills and knowledge level of the user. Various search modules are described for the initiation, execution, and report of the search. Figure 1; references 9.

Interactive System to Solve Problems in Multicriteria Optimization (Brief Communication)

[Article by V. L. Volkovich, L. F. Dargeyko, T. G. Yaremchuk, V. V. Boyko, and A. A. Volkovich]

UDC 519.853

[Abstract] The DISMOP package of applications programs is a decision-making support system specialized for the selection of alternatives according to a model developed by experts and entered into computer memory. The model consists of a set of alternatives and functions to be performed on the set to reach an optimum solution. Criteria are given in the form of linear or discrete-separable equalities. The model may use linear, integral, or boolean programming. The model uses the method of limitations to reach the optimum point. First, the user enters the data and criteria, and the model and parameters may be edited in a convenient format. Best and worst values are assigned to each criterion. Second, there is interactive determination of effective solutions. At this point the user can fine tune desired criterion values. When the user obtains a satisfactory solution, he may examine the values of all criteria in text or graphic format. The application program modules use a simplex method algorithm to solve linear models, and special algorithms to sift out undesirable outcomes in discrete problems. The package may be used on the YeS 1035 (or higher) computer working under the YeS 6.1 operating system with 5062 disks and a 7920 display. The programs are written in PL/1 and Assembler. A version is being developed for personal computers. References 7.

Image Processing System on the Delta High-Performance Computing Complex

[Article by V. I. Dianov, V. A. Smolnikov, A. V. Khatsanskaya, V. V. Shchedrin]

UDC 681.3.06

[Abstract] Many systems have been developed for the automatic digital processing of images and improvement of image quality. A list of western programs, mainly for use on the PDP-11/20, is given. The article describes an image processing system (named IPS) developed at the Glushkov Institute of Cybernetics of the Ukrainian SSR Academy of Sciences for the processing of continuous high speed data streams from remote sensing of Earth. This system is specialized to receive high speed data streams, process them, and store them so images can be processed in real time. The system has a throughput of up to 4 megabytes/second. The central computer is a supermini computer with a pipeline processor, which handles 3 MIPS. There are four selector channels, each with a throughput of 8 megabytes/second. The interactive subsystem accomodates up to 16 operator locations for interactive preliminary processing. Up to 10 gigabytes of data can be stored. The system includes a special bulk analysis processor, a high density Slumberge

magnetic tape drive, a device to digitize halftone negatives and a printer. Images are produced using recursive range algorithms which calculate local histograms and their characteristics. Examples of range variations in the image analysis algorithms are given. The user can perform arithmetic or logical operations on an element-by-element basis, as well as suppress or isolate noise, smooth the image, enhance contrast, and isolate borders. References 7.

KM1810VM83 Floating-Point Microprocessor

[Article by N. Yu. Danilov, A. D. Polukarov, and M. V. Stolyarov]

UDC 681.325.5-181.4

[Abstract] The microcontrollers in the Midikon family, which are based on an LSI K580 microprocessor complex, are intended for use in constructing text-and-graphic systems for a broad range of applications. From the standpoint of their output information, individual design and circuit engineering, and capabilities, they are classified as alphanumeric (text), graphic, or combined microcontrollers. The family consists of four basic models: the MGK-80, the MTsD-90, the Midikon-80, and the MDK-80. All are based on boards in the Elektronika 60 microcomputer standard. Modifications of the basic models exist as well. The MGK-80 is used jointly with the Elektronika 15IE-00-013 display to form a combined text-and-graphics display; its display is based on the raster principle. The ATsD-80 microcontroller is intended for joint operation with an MGK-80 microcontroller. It is compatible with three types of data representation in a KM1810VM87 (namely, integral, packed binary decimal, and real data); parameters and operating modes of the KM1810VM87 microprocessor; and connection of the KM1810VM87 microprocessor to a computer system. Figures 20; tables 12.

Test Processor for Multiprocessor Systems

[Article by B. V. Antonyuk, I. V. Omelchuk, V. N. Prisyazhnyuk, and G. G. Tereshchenko]

UDC 681.322-192

[Abstract] One of the main tasks entailed in designing high-reliability multiprocessor systems is to increase the level of the processors' monitorability. This may be done by introducing monitoring facilities into the system. Computation results can be compared by using either a diagnostic processor or by using the very processors of the structure undergoing diagnosis. In the case of multiprocessor systems with a distributed bus this can be accomplished during the process of transmission via the bus. This article describes a processor that provides selective monitoring of information sent via a distributed bus. It is based on a series K1810 multiprocessor complex LSI circuit. The test processor contains the following: microprocessor units, a unit to formulate a system bus, a unit to formulate a resident bus, a programmable interrupt controller, a programmable timer, a local memory, an address decoder unit, and a unit to monitor information sent via the system bus. The test processor described makes it possible to detect faulty processes in the multiprocessor system and fix a transmission protocol in the address space being monitored. The monitoring mode is program controlled. The presence of a reliable or unreliable transmission or the absence of any transmission can be established by analyzing the state of an exchange sector. When the test processor is used with the RMKh-86 operating system, the sector being monitored becomes an additional object of the operating system's nucleus, and the system calls become extensions of the operating system. This article includes a block diagram, time diagram, and detailed description of the processor's operation. Figures 3; references 9: 5 Russian, 4 Western.

Object-Oriented Implementation of Interaction During Access to a Relational Data Base

[Article by I. M. Korytnyy and I. O. Mukalov]

UDC 681.3.06

[Abstract] This article describes the object-oriented implementation of interaction by means of a personal relational-type data base manager using an alphanumeric display. The system described contains elements that are related to both spreadsheets and means of working with screen forms of data base managers (such as dBase). It differs from these mainly in that it presents a unified model of processing information within the framework of the interaction described. The data base manager presented here is implemented in the RT-11 (RAFOS) operating system in the Pascal programming system and may be executed on SM1300, SM4, SM1420, DVK2, DVK3, and analogous computers. The proposed means of interaction is based on direct visual editing of information. Three forms of data display are provided, i.e., access, editing, and view. These three display forms are shown to be adequate for basic data base management applications. Figures 4; references 4: Russian.

High-Resolution Display

[Article by M. V. Kollegov and V. V. Repkov]

UDC 681.326

[Abstract] This article describes a relatively inexpensive, high-resolution black-and-white display that was developed at the Nuclear Physics Institute of the Siberian Division of the USSR Academy of Sciences for use in graphics applications. Its working field measures 1024 x 768 points, it has a field change frequency of 50 Hz, one line lasts 25 μs (20 μs forward and 5 μs back), and its video amplifier has a bandwidth of 50 MHz (with 20 ns required to illuminate one point). The device is based on a 61LK8B kinescope. The display provides a resolution of no less than 200 lines throughout the screen's entire field. The display's drive is based on one binary board of the Elektronika 60 microcomputer standard. It outputs 1024 x 768 points of a scanned image with 16 brightness gradations (or it can be used with a color monitor, in which case the image displayed can contain 16 colors). The drive contains a video storage, a graphic processor, a storage for output color recording, and an interface. The graphic processor is a 12-bit microprogram automaton based on three K1804VS1 processor sections. Figures 6.

Display Microcontrollers of the Midikon Family

[Article by N. Yu. Danilov, A. D. Polukarov, and M. V. Stolyarov]

UDC 681.327.22

[Abstract] The microcontrollers in the Midikon family, which are based on an LSI K580 microprocessor complex, are intended for use in constructing text-and-graphic systems for a broad range of applications. From the standpoint of their output information, individual design and circuit engineering, and capabilities, they are classified as alphanumeric (text), graphic, or combined microcontrollers. The family consists of four basic models: the MGK-80, the MTsD-90, the Midikon-80, and the MDK-80. All are based on boards in the Elektronika 60 microcomputer standard. Modifications of the basic models exist as well. The MGK-80 is used jointly with the Elektronika 15IE-00-013 display to form a combined text-and-graphics display; its display is based on the raster principle. The ATsD-80 microcontroller is intended for joint operation with an MGK-80 microcontroller. It is compatible with the VT-52 display produced by DEC (United States). The Midikon-80 is intended for use in creating microcomputer-based text-and-graphic systems. It consists of a combination of the MGK-80 and ATsD-80 microcontrollers but has a single common microcontroller and is contained on one board with dimensions of 240 x 270 mm in the Elektronika 60 microcomputer standard. The MDK-80 microcontroller was developed for use in series-produced ATs displays to improve their technical characteristics and expand their capabilities. Like other members of the Midikon family it is based on series K580 LSI and has 32 kbytes of common memory. Real graphic displays and text-and-graphic systems built around the Midikon family of microcontrollers are currently operating as part of SM4, SM1420, and Elektronika 60 computers with P-300, PLOT-10, GRAFOR, and ATOM-84 graphics packages and PLOT and SURF interactive graphic systems. Figures 2; references 3: Russian.

Instruction Systems for Elektronika MS7401 Raster Graphic Display

[Article by A. O. Latsis and Vs. S. Shtarkman]

UDC 681.3

[Abstract] All of the functions performed by a display with respect to displaying information on screen are realized by a program in a computer that is built into the display. The video memory is directly addressed by this computer's processor. Thus, the display instruction system that is "visible" to the main computer is determined by the control program that is "sewn" into the ROM of the display's built-in computer. Several decisions must be made when designing the instruction system of a graphic display, i.e., whether to use a combined or graphic mode, which coding method to use, whether to use vector input and selective erasure, which technique to use when organizing selective erasure on a graphic display, and whether to store vector patterns. The old instruction system for graphic displays is fraught with several shortcomings: the fact that the terminal is sometimes "left hanging", the appearance of episodic "doubling" of the characters received from the computer on some displays, the fact that adjustment of the display's parameters is complex and not visible, and the fact that deleting a segment of a display file does not free the space it occupied. In the new system proposed for the MS7401 raster graphic display, each graphic command is coded by a separate ESC-sequence; thus, the graphic instruction's prefix is shortened to one ESC character. A single method of coding the parameters of graphic instructions, i.e., binary 6-bit, is stipulated. A short (1-byte) coordinate format is provided for the most frequently used instructions. The new system uses vector input and several capabilities that are specific to rasters; the drawing speed is thus increased by a factor of 2 - 2.5. There are no prespecified types of lines; rather, the user

specifies the line type by using an 8-bit code. Working with a stored vector pattern is simplified, and there is a powerful apparatus for storing and restoring the state of the display that is especially useful when working with subroutines. The idea for the raster capabilities was borrowed from the language BASIC for IBM PCs. There are only two raster capabilities, i.e., filling in a "rectangle" and working with a screen buffer. In the new system, fragments of the control program from the main computer can be loaded into the built-in computer. When desired, the user can use the program loaded from the main computer as a "secondary loader" and completely reprogram the built-in computer at his discretion. Figures 2; references 4: 3 Russian, 1 Western.

Gamma-7.1 Color Half-Tone Professional Workstation

917G0027G Moscow MIKROPROTSESSORNYYE SREDSTVA I SISTEMY in Russian No 5, Sep-Oct 90 (manuscript received 5 May 88) pp 45-46

[Article by A. A. Buchnev, V. G. Sizykh, and V. F. Minin]

UDC 681.327.23

[Abstract] The Gamma-7.1 professional display station is intended for use in automated design systems of any type in the performance of tasks related to image processing, mathematical modeling, controlling industrial processes, etc., as well as with personal computers. The screen has a resolution of 1024 x 768 in the case of progressive scanning with a frame frequency of 50 Hz and 1024 x 1024 in the case of line-jump scanning at 30 Hz. It has a total image size of 1024 x 1024 pixels, 1 Mbyte of video memory, 256 brightness or color gradations, 2^{24} possible colors and shades, a minimum pixel display time of 20 ns, a maximum video output rate of 50 Mbyte/s, a vector generation speed of 640 ns/pixel, and a character generation time of 100 µs. It is based on a modern two-processor architecture permitting parallel processing of high-level graphic data display and generation of raster data display, which makes the display system more interactive overall. This article includes a block diagram of the workstation's bit-slide graphic processor and information on the workstation's video RAM mapping, color palettes, and software. Figures 1.

High-Resolution Color Monitor for Raster Displays

917G0027H Moscow MIKROPROTSESSORNYYE SREDSTVA I SISTEMY in Russian No 5, Sep-Oct 90 (manuscript received 21 Mar 88) pp 46-48

[Article by S. T. Vaskov, O. A. Kolomeyev, O. V. Perekrest, G. F. Sitnikov, and S. Ye. Tkach]

UDC 681.327.69'22-535.6

[Abstract] The high-resolution color monitor described herein was developed to display large volumes of information at the Computer Technology Special Design Bureau of the Siberian Division of the USSR Academy of Sciences. The display's high-resolution cathode-ray tube [CRT] (diagonal measurement, 51 cm) is commercially produced in combination with a scanning yoke. The monitor has the following specifications: dimensions of the CRT screen's working field, 280 x 372 mm; line scanning frequency, 28 to 40 Hz; frame scanning frequency, 40 to 80 Hz; video path band, up to 45 MHz; number of addressable points at F = 50 Hz, 1024 x 768 elements in the case of progressive scanning and 1280 x 1024 elements in the case of line-jump scanning; and beam misconvergence, no less than 0.3 mm throughout the screen's entire field. This article includes a block diagram of the monitor, a time diagram of the signal synchronization, and a block diagram of the beam convergence unit. Special attention is paid to the scanning unit, synchronization system, digital convergence unit, and static and dynamic convergence corrector. Figures 3.

Multimicroprocessor System for Real-Time Spectral Signal Analysis

917G0027I Moscow MIKROPROTSESSORNYYE SREDSTVA I SISTEMY in Russian No 5, Sep-Oct 90 (manuscript received 18 Feb 88) pp 57-61

[Article by V. A. Zaytsev, S. N. Krotov, S. D. Proshkin, and A. V. Savchenko]

UDC 681.3

[Abstract] The proposed system for spectral signal processing is based on the use of additional processor boards in a microcomputer with a multiprocessor architecture. Parallel data processing is conducted in several processors, which permits the spectral analysis of signals on 1 to 8 channels simultaneously in accordance with a fast Fourier transform algorithm for 2^n (n = 1 to 12) points with a signal digitization frequency of up to 20 kHz, including real time operation with a digitization frequency of up to 400 Hz. As data arrive from an analog-digital converter at a frequency of more than 400 Hz, an input data queue is formed; these data are subsequently processed in the order in which they are received. The signal spectral analyzer system is based on a Labtam 3015 multimicroprocessor containing two 8086 microprocessor-based processor boards in its configuration, i.e., a central processor and high-resolution graphic display controller board. It also includes a Z80 microprocessor-based board to organize I/O to magnetic disks. The programs implementing the fast Fourier transforms and the queues of the data block and Fourier coefficient block are stored in local memory, therefore the load on the common bus is insignificant, and several fast Fourier transform boards can be connected simultaneously. The system was developed at the Experimental Cardiology Institute of the All-Union Cardiology Scientific Center

of the USSR Academy of Medical Sciences to process physiologic signals. Figures 2; tables 2; references 5: 1 Russian, 4 Western.

Real-Time Data Processing System Module

[Article by Yu. A. Padiryakov, V. T. Belinskiy, and V. A. Zhurilo]

UDC 681.327

[Abstract] Vector processors are not fast enough to permit real-time data processing, hence the need to distribute data processing tasks among several processors constituting a system. It is advisable to implement data exchange directly between vector processors by using high-speed equipment. The real-time data processing module described herein has been designed for this purpose. It is based on an Elektronika MT-70 high-speed peripheral processor and an Elektronika MS 1201.02 computer mounted inside the high-speed peripheral processor's case. Because the MT-70 processor had no I/O channels, the peripheral processor was modified to permit external data exchange. The microcomputer contains enough interfaces to connect peripherals when testing the module, debugging control programs, and loading debugged programs. I/O is implemented independently, asynchronously, in parallel, word by word, and with quantization. The module features a total I/O exchange rate in excess of 400,000 words per second; the maximum input speed is determined by microprogram length and amounts to 500,000 words per second. At such speeds, however, the peripheral processor's controller cannot be addressed by the computer. Figures 1; tables 2; references 3: Russian.

KR1801RYe2 ROM

[Article by G. Glushkova]

[Abstract] The KR1801RYe2 microcircuit is a read-only memory [ROM] with a capacity of 65,536 bits and 4096 x 16-bit organization. It is based on n-channel MOS transistor semiconductor technology and is manufactured in a type 239.24-1 24-lead case. It is a part of the series K1801 microprocessor set along with the K1801VM1, KM1801VM2, and KM1801VM3 microprocessors; K1801VP1-030, K1801VP1-033, K1801VP1-034, and K1801VP1-035 interface microcircuits; and K573RF3 reprogrammable ROM. This concise communication presents a schematic and time diagram of the KR1801RYe2 ROM and tables detailing the purpose of the KR1801RYe2 microcircuit's leads and its electrical characteristics. Figures 3; tables 2.

One Problem of Locating a Radiation Source

[Article by E. Ye. Gachinskiy, S. V. Matveyev, A. N. Trifonov, and M. Yu. Cherkashin, Moscow]

UDC 609.7

[Abstract] The problem of automatic detection of several local radiation sources with an intensity that is not known a priori is important in such applications as localizing sources of radioactive contamination of a site in the event of the accidental destruction of nuclear power facilities along with scattering of the facility's structural components throughout the surrounding territory. In such a case, each structural component may be viewed in terms of a radioactive radiation source with an intensity that is greater than that of the surrounding area. A deterministic local automatic search method is proposed to solve such a problem. The authors examine a mathematical model of such a facility and analyze the problems of the precision of determining the sources' locations, the effect of the sources' relative location on the precision of determining their coordinates, and the bounds of the theoretical distinguishability of the sources. They examine continuous and step-by-step algorithms implementing the gradient and fastest rise methods and an algorithm with synchronous detection. They discover that the algorithms implementing a gradient search method (both continuous and step-by-step versions) are faster, although step-by-step versions of the algorithms based on the fastest rise method are comparable in speed. The algorithm with synchronous detection was found to be the slowest. The authors conclude that the problem formulated may indeed be solved by local search methods. They call for further study of the problem to increase the speed of the search systems and to make an allowance for possible noise and the characteristics of actual measuring devices and actuators. Figures 9; tables 2; references 5: Russian.

Harmonic Optimal Scene-Matching Navigation

[Article by A. A. Krasovskiy, Moscow]

UDC 629.7.05

[Abstract] One of the factors impeding the wide-scale use of optimal scene-matching navigation based on geophysical fields is a difficulty associated with data base organization and support in the form of global digital models of geophysical fields. This applies particularly to the "matrix" or "grid digital" models of geophysical fields toward which existing optimal scene-matching navigation systems with point field sounding are geared. This article substantiates the use of harmonic models of navigational geophysical fields, which have long been used in describing the earth's magnetic and gravitational field in the lower-harmonic range, to provide the data base support needed for the optimal scene-matching navigation. Models in the form of a binary trigonometric arrangement with respect to latitude and longitude are recommended. These models are derived through a binary discrete Fourier transform of a matrix model of a navigational geophysical field (with a large increment). They are shown to be advantageous both for harmonic mapping of navigational geophysical fields and for no-search optimal scene-matching navigation with on-board harmonic navigational geophysical field maps. The method of local scene-matching navigation based on navigational geophysical fields that is proposed is demonstrated to have a sufficiently high precision. Figures 3; tables 3; references 11: 10 Russian, 1 Western.

Nonlinear Correlation of the Dynamic Characteristics of the Steering Control Linkage of the Energiya Booster Rocket

[Article by S. I. Bayda, D. S. Belitskiy, V. V. Kudryavtsev, and V. I. Shutenko, Moscow]

UDC 629.7

[Abstract] Due to a number of design constraints arising from the method used to control its thrust vector, the Energiya booster rocket presented those developing the steering linkage of its second stage with a number of design problems. Those designing the linkage had to meet two conflicting requirements: on the one hand, a specified phase delay had to be provided that was determined primarily by the increased value of the linkage's Q-factor; on the other hand, the resonance at the frequency of the elastic circuit (i.e., steering linkage-load), which is intensified as the Q-factor increases, had to be suppressed. Analysis of both requirements showed that it would be impossible to meet both requirements simultaneously within the framework of a conventional steering linkage structure without some sort of additional correction. The study reported herein examined the problem of correcting the dynamic characteristics of the steering linkages of the second stage of an Energiya booster rocket by using a hydromechanical correcting device containing a nonlinear component. A mathematical model of a loaded steering linkage was developed, and the effect of the correcting device's parameters on the steering path's frequency characteristics was examined on the basis of nonlinear motion equations for the system steering linkage-load. The amplitude-frequency characteristic of the circuit steering linkage-load of the

second stage of the Energiya booster rocket was determined experimentally at different stages of developing the steering linkage and booster rocket together and during separate work on the steering linkage alone. The steering linkage and correcting device developed were then subjected to design flight tests with the Buran orbital craft. The tests confirmed the correctness of the circuit engineering solutions included in the design of the second-stage steering linkage as well as the effectiveness of the procedure for selecting the nonlinear correction parameters based on continuously refining them during the process of developing the booster rocket. Figures 4; references 3: Russian.

Probability Analysis and Synthesis of Control of the Motion of an Aerospace Vehicle

[Article by V. D. Volodin, V. N. Yevdokimenkov, V. I. Karlov, and M. N. Krasilshchikov, Moscow]

UDC 629.73

[Abstract] The essence of controlling the trajectory of the prelanding maneuver of an aerospace vehicle (such as the Space Shuttle) consists of guiding it with a high degree of precision to the specified end range before the beginning of the landing path with a specified high probability from the maximum range of allowable initial conditions. Ensuring a high precision of control is especially critical for an aerospace vehicle because of its relatively low aerodynamic quality versus that of conventional aircraft and because of the various uncontrollable factors (including atmospheric disturbances, deviations of such vehicles' main aerodynamic and mass characteristics from the nominal values, and instrument errors on the part of the systems measuring the craft's state) whose effects are exacerbated by the absence of any engine thrust to control the vehicle's trajectory. This article is devoted to analyzing and synthesizing control of the motion of an aerospace vehicle during that section of its prelanding maneuver at an altitude from 20 to 4 km in a manner that provides a guaranteed probability ($\alpha \geq 0.999$) that its landing will proceed under the required conditions. The method proposed is based on a combination of deterministic synthesis methods and methods of solving probability analysis problems. The results obtained permit a quantitative estimate of the probability that an aerospace vehicle will land successfully and an estimation of the range of admissible positions for the aerospace vehicle at the beginning of its prelanding maneuvering from which its successful landing will be guaranteed. In many respects, the results obtained also supplement previously obtained results analyzing control of the Space Shuttle as well as the results of the solution of an analogous problem studied elsewhere. Figures 5; references 7: 6 Russian, 1 Western.

Constructing Algorithms to Control the Group Flight of Helicopters While They Jointly Transport a Single Cargo on an External Suspension

[Article by O. Yu. Ilin, Kiev]

UDC 629.735

[Abstract] One way of expanding helicopters' transport capabilities when transporting heavy, large, indivisible cargoes is to use several helicopters equipped with a common external suspension. This article examines the problem of constructing an algorithm to control the group flight of helicopters when transporting a single cargo on a group external suspension. The proposed solution is based on the concept of solving an inverse dynamics problem. In accordance with the proposed algorithm, the external control loop is synthesized sequentially. First, an algorithm to control the cargo's coordinates is synthesized. An algorithm to control the pendulum's angles is synthesized next. Finally, an algorithm is synthesized to control the relative motion of the helicopters. Standard helicopter autopilots are used to implement control in the internal loop. Analysis of the modeling results shows that the proposed control algorithms guarantee that the cargo will reach a point in space along the prescribed trajectory with acceptable control process characteristics without the helicopters coming dangerously close to one another or any buildup of oscillations on the part of the cargo. The method proposed, which uses two helicopters, consists solely of fundamental principles that may be extended to a larger number of helicopters, which allows development and individualization of the control procedure with other flight modes and other types of mathematical models. Figures 3; references 7: 5 Russian, 2 Western.

Developmental Support for Knowledge Acquisition Processes in the System Software Tools for the Development of Expert Systems

[Article by V. F. Khoroshevskiy and S. Yu. Shchennikov, Moscow]

UDC 519.007

[Abstract] The formulation, debugging, and maintenance of knowledge bases is a bottleneck in modern intelligent systems. Such processes have been poorly studied and formalized and have only extremely weak developmental support. Of the five levels of acquiring

knowledge (linguistic, conceptual, epistemological, logical, and realized), essentially only the last two have received attention by those creating such development tools. This work discusses developmental support of the processes of formulating knowledge, based in the environment of the system Software Tools for the Development of Expert Systems now accessible to users, and analyzes ways in which these development tools can be developed in the near future. Frame-production knowledge representation languages such as Pilot-2 are discussed. Pilot-2 is shown to be similar to the well-known production programming language OPS5, but it possesses more powerful means of describing the situations analyzed during the process of its operation. It is further demonstrated that because many of its statements are syntactically and semantically similar to those in the language C, Pilot-2 can be used to address programs and functions written in C. The main principles of designing this language are described, and the syntax and semantics of its main structures are presented along with examples illustrating their use. Concluding the article are selected examples of work on creating developmental tools related to the correlation of knowledge on the linguistic, conceptual, epistemological, and logical levels. Figures 2; references 10: 6 Russian, 4 Western.

Data Protection Key Distribution and Problem of Authentication

[Article by L. M. Ukhlinov]

UDC 621.391.7

[Abstract] The secrecy of information sent via communications channels to users interacting in an information computer network depends entirely on the secrecy of the protection key, which must be known to all users participating in the data exchange. As the sizes of networks increase, so too do the volumes of memory needed to store each user's lists of protection keys grow. At the same time, the large number of copies of the protection keys stored in the network greatly increases the probability that keys will be stolen and that the information stored in the network will be compromised. One solution to these problems is to develop special key distribution protocols that do not necessitate the storage of any secret information by users except for their key, which is known only to them. This article examines the main principles used to establish authenticity in information computer networks and demonstrates methods of using them in key distribution protocols to set up safe communications between two users and group interaction (conferencing) among network users. The key distribution methods discussed are based on the use of cryptosystems with a shared key. The need to include authentication procedures in key distribution protocols is substantiated. It is demonstrated that if the function

$R^{(j)}(I)$ (where $j = 2$) is used to formulate a data protection key to establish group interaction with $m > 2$ network users, then another individual may obtain the key by listening in on three channels of communications between any participants who are sequentially included in the set I. It is further proved that when the function $R^{(j)}(I)$ is used to formulate a shared data protection key, the safety of the key increases as the number of cofactors forming the elements of the function $R^{(j)}(I)$ increases, with maximum safety being achieved when $j = m$. References 12: 4 Russian, 8 Western.

Prospects for the Development of CAD in the Technological Preparation of Production

[Article by S. N. Grinfeld, engineer, and A. A. Nesterov, Candidate of Technical Sciences]

UDC 658.512.4.011.56"313"

[Abstract] A great deal of experience has been gained in the automation of technological processes. Many procedures have been automated, but this has been done in a piecemeal fashion, each process automated differently or separately. As a result, much redundant development work has been done. There has been no integration of automated processes, because often incompatible equipment or software has been used. Technological processes have a great deal in common, and one type of software should be developed to handle identical production processes. There should also be a standard software to produce the necessary technical documentation, both graphic and textual. In design, whether one is designing a part of or an entire machine, similar or identical steps must be taken in a certain order, and certain information is sought on various criteria (goeometry, strength, processing tools, etc.). These steps and standard information are targets for automation. An example is given of the geometric and contour analysis of a part, and how this analysis could be automated using a computer, database, and expert system. It was found that it was expedient to design software for personal computers compatible with the IBM PC/AT with subsequent development of software for 32-bit machines. Soviet CAD programs which apply these principles are listed. References 3.

CAD for Technological Processes for Dynamic Use

[Article by A. E. Polonskiy, Candidate of Technical Sciences]

UDC 658.512.4.011.56.004:621.9

[Abstract] Changes in plans, materials, and design often decrease the efficiency of CAD programs. CAD

programs for technological processes need to be supplemented with software to increase their flexibility: a hierarchical database with the possibility of recursive composition of information, algorithms to automate the design of certain parts or types of parts, and libraries of procedures. Also important is the capability of creating and editing technical documents in interactive mode and the ability to compress information. The Dipol system makes it possible to create a variety of technical documents for machine and instrument building and meets the production and technological needs of the industry. It was designed for the SM computer. Dipol 2 may be used on the IBM PC. The article describes the system used to store information and how commands are used to access the information. The Dipol system reduces the design process to the accurate selection of format rules which form a technological algorithm. If there is insufficient input information, technical documents can be partially completed. Fifteen standard forms are supported. References 4.

CAD Control Programs for Metal-Working Equipment With Numerical Control

[Article by M. V. Zhuromskiy and V. S. Suvorov, Engineers]

UDC 658.512.4.011.56:621.952-529

[Abstract] A CAD system has been developed to design the control programs for equipment with numerical control. It was developed by the Minsk branch of the All-Union Scientific Research Institute for the Organization of the Machine Tool and Instrument Industry. The system can be used for the following machinery: lathes, drilling and milling machines, electro-erosion machines, polishers, multi-function machines, plasma metal cutters, punch presses, and other numerically controlled machines which process contours or positions in a plane. The system requires 512 kilobytes of memory, 29 kilobytes on magnetic disks, a paper tape output device (for machines which require paper tape input), and operating system YeS 6.1 or higher. It was designed for SM computers and SM-based workstations with the RB operating system. Work is done in batch mode; an upgraded version permits work in interactive mode. Once the user imputs information on the part's geometry, the processing to be done, and the machines to be used, a pre-processor handles special problems and passes the results to the processor. The processor outputs the result in the CLDATA language. The post-processor forms machine-specific commands. In addition, it provides all necessary production forms. The article lists the types of functions implemented by the control programs, for example, varieties of milling geometry, hole drilling

specifications, etc. Work is continuing to expand the capabilities of the system (more cutting and drilling functions, more types of machines). The program is being reworked to support the YeS 1841 and 1842 and the IBM PC, as well as other computers.

Model of Protocols in the Local Computer Control Network STANSET for Flexible Manufacturing Systems

[Article by V. L. Sosonkin, Doctor of Technical Sciences, Prof. S. G. Chekin, Candidate of Technical Sciences, and A. V. Kovtunov, Engineer]

UDC [681.3.06 + 681.3].004:658.52.011.56.012.3

[Abstract] While there are standard models for the support of communications in a local computer control network (ISO-OSI and MAP), there is no full information schematic for the support of network services. When STANSET was being developed, an information model of the communications environment was constructed using the formalism of hierarchical Petri networks. This article offers a brief description of STANSET and the principles used in constructing a formal model of this communications environment. The STANSET protocols are a simplified version of MAP. The STANSET network is a fiber-optic ring LAN. The network may be expanded to a multi-ring system with a separate ring for engineering workstations. The communications environment is described, and an example is given of how a request for service is processed. Network service is formed by a united group of functional calls and responses. A three-phase life cycle of communications is constructed during these calls and responses. The events of individual phases are supported by a multilevel system of protocols which can be visualized using the formal vehicle of hierarchical Petri networks. This makes it possible to define a single information model of the communications environment protocols that encompasses the internal and interface functions of the protocol levels associated with the work of communications environment primitives. Figures 8; references 4: 1 Russian, 3 Western.

Development of the Graphics Capabilities of Printers

[Article by L. I. Manpil, Candidate of Technical Sciences]

UDC 681.327.54.071.1 (048.8)

[Abstract] The trend in the development of printers is toward better graphics capabilities, that is, higher resolution. This article examines dot-matrix printers. The

principles of dot-matrix printing, for example, the configuration of pins in the print head, directionality of printing, and the number of pins, are discussed and it is explained how these factors affect resolution. The advantages, disadvantages, and common features of dot-matrix printers are discussed. A table of the specifications of 26 common dot-matrix printers from the United States, Europe, and Japan is given, and some models are discussed in more detail. It is indicated that Epson's 48-pin model has a better resolution (360 dpi) than table-top laser printers (300 dpi). The article indicates how dot-matrix printers are being further improved by reducing noise levels, increasing quality and durability, expanding functions, and decreasing size, power consumption, and cost. References 4: 2 Russian, 2 Western.

Implementation of a Noise-Suppressing Serial Interface for a Distributed System of Telemetry and Remote Control

917G0022F Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 8, Aug 90 pp 18-19

[Article by A. G. Tsokur, engineer]

UDC 681.324:621.398

[Abstract] A noise-suppressing interface was developed for high-speed distributed telemetry and remote control systems. The system used two-wire coaxial cable and was intended for use in automated systems for experiments and machine testing, including high-current AC and DC equipment. This interface has an improved noise-suppressing serial information channel. The sequence of square pulses transmitting the message enter the main radio frequency transmitter from the coder. The pulses are transformed into radio frequency pulses which enter the trunk through a high frequency transformer and a resistor. The pulses leave the trunk, pass through a resistor and transformer to enter the main radio frequency receiver. The pulses are selected by frequency and amplitude and are sent to the decoder. The trunk is a coaxial cable. To reduce the number of spectral components in the signal, the radio frequency pulses entering the transmitter from the power amplifier have a rounding bell-shaped form. The radio frequency pulse is about 700 ns long, with a modulating pulse 500 ns long. The filling frequency is 20 MHz, and the transmission speed is 1 megabit/second. The power supply is +12 V, and the output power of the transmitter is 2 VA. Figures 4; references 4.

Hardware and Software Complex for an Automatic Control System With Remote and Local Communication

917G0023A Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, Sep 90 pp 1-4

[Article by A. M. Pshenichnikov, Doctor of Technical Sciences]

UDC 621.398.004:658.51.011.56

[Abstract] The system consists of a set of controller and programmable stations, devices for the collection and output of data, color video terminals, external switching devices, local network stations, personal computers (IBM PC, SM 180-13, and SM 1810-63), black and white video terminals, printers, and floppy disks. The programmable devices are intended for the formation and storage of databases, sending and receiving of messages, organization of paths for local and remote networks, data compression, and peripheral control. Values from remote sensors can be received, compared with other values, scaled, and displayed. External KMOS RAM memory is used for data base and program storage. The external memory has an emergency power supply. A very detailed description is provided of how the units may be combined and connected to suit the varied needs of the purchaser. Schematics are offered, and the specifications of individual converters and memories are described. The software includes a CAD system, a distributed data base, a fast real time operating system, a program to collect and transmit information, programs to control transmissions, a data analysis program, diagnostic programs, and other programs. The system is intended for use in electrical power systems, transport, metallurgy, the chemical and petroleum industries, and communications. Figures 6; references 7.

PS 2100 Multiprocessor

917G0023B Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, Sep 90 pp 5-6

[Article by I. I.Itenberg, A. S. Nabator, Candidates of Technical Sciences, S. I. Danilchenko, V. I. Kheyfets, and V. I. Yashchenko, engineers]

UDC 681.322:681.3.012

[Abstract] The PS 2100 multiprocessor was designed for the processing of large data files using regular algorithms. It is compatible with the PS 2000. It can be used (with an interface card) as a procedural arithmetic expander for any 16- or 32-bit computer. The multiprocessor uses a SIMD architecture. Microcommands are carried out in 0.14μs. There are four input and output subchannels. Four (input or output) may operate in synchronous or multiplex mode. The multiprocessor is mounted on an 800 x 440 x 1600 mm base. The stand is supplied by a three-phase 380/220 V power supply network. The multiprocessor can handle fixed and floating point 16- and 32-bit data (also 64-bit for floating point data). The MIPS rates are as follows: 16-bit fixed point addition, 150; multiplication, 90; 32-bit floating point addition, 64; multiplication, 30. The RAM volume is 8 - 32 megabytes. The throughput of the data channel is 40 megabytes/second in synchronous mode and about 7 megabytes/second in multiplex mode. Figures 1; references 3.

Characteristics of the Introduction of Typical Software in Automated Data Processing Systems in Agroindustrial Complexes

[Article by S. K. Simkin, Cadidate of Technical Sciences]

UDC 681.3.06.004:631.145

[Abstract] A network of specialized collections of algorithms and programs has been developed. Each collection is oriented toward problems in a specific field. The current problem is determining how they can interact with interbranch collections so that general purpose software can be introduced. The role of these central collections is to disseminate typical software for the organization and management of the information bases of automated data processing systems, organization of the computing process, and the implementation of current economic and mathematical methods. Of special importance is the introduction of personal computers. This requires a reexamination of the use of software and automated control systems on larger machines and a reorientation toward small machines. Outlines are developed for the efficient cooperation of specialized and centralized algorithm and program collection points. It is recognized that standardization is also an important issue in terms of software and hardware. User services must also be developed.

Computer Complex to Automate the Design of the Technical Structure of Flexible Production System Control Systems

[Article by E. G. Godynskiy, A. N. Kozlov, N. A. Popkov, M. B. Tsydikov, Candidates of Technical Sciences, and I. Yu. Shabalin, Engineer]

UDC 658.512.2.011.56

[Abstract] A systems approach is taken in the design of flexible production systems. The first design stage involves the synthesis of the technical structure, which is the foundation for the selection of equipment. The second stage is the development of an automated control system. The article examines a computer complex to automate all stages of the design of the technical structure. It prepares the technical assignment, automates the synthesis of the technical structure, creates and manages databases, and prepares textual and graphic documents in accordance with standards. The system contains an SM-4 minicomputer and a DVK microcomputer. The user inputs specifications, and the development of the technical structure is interactive. The structure is optimized according to user criteria. The user can store various structures in a data base for modification. The

complex software includes material on standards, document forms, and data bases. Programs are written in PASCAL. They have an overlay structure and occupy 230 kilobytes of external memory. Figures 1.

Kronos: Family of Processors for High-Level Languages

[Article by D. N. Kuznetsov, A. Ye. Nedorya, Ye. V. Tarasov, and V. E. Filippov]

UDC 621.3.049

[Abstract] Kronos is a general name for a family of 32-bit processors intended for creating micro- and minicomputers. Their architecture is geared toward supporting such high-level programming languages as C, Modula 2, and Pascal for open-architecture computers from built-in microcontrollers and single-processor workstations to multiprocessor configurations. Modula 2 has been selected as the base language for the family's development. The architecture of the Kronos processors differs from the conventional processor architecture in several important respects. Expressions are computed on a fast hardware stack (with a low fixed depth) that preserves its contents when functions-procedures are called or when processors are switched; the code and data regions of any process are separated; even in a code segment there is no absolute addressing; the developments of types of address reflect the concepts of modern programming languages (addressing of local, global, and peripheral objects and objects of statically embedded procedures); and it uses a patented method of intermodule addressing that permits the organization of dynamic loading, linking, and running programs. The Kronos 2.X family includes three processors, i.e., the Kronos 2.2, 2.5, and 2.6. All use one instruction system but differ from the standpoint of their internal configuration, speed, and design features. This article details the architecture of all three processors and their operating system. Special attention is paid to the convenience of their interfaces. References 7: 5 Russian, 2 Western.

Single-Board RAM With Data Storage Checking

[Article by D. A. Bruyevich, R. M. Vorobyev, and A. G. Kulikov]

UDC 681.327.2-185.4

[Abstract] The RAM with self-checking and error correction [OZU-SKM] is based on an Elektronika 60 half-board. It has a capacity of 128 kbyte, a word length of 8 or 16 bits, a cycle length of 1.2 µs, a power source voltage

of 5 +/- 0.25 V, a power requirement of 5 W, a working temperature range of 5 - 50°C, an MPI interface, overall dimensions of 135 x 240 mm, and a mass of 200 g. Its storage consists of over 26 dynamic KR565RU5 LSI circuits. The new RAM with data storage checking has a number of capabilities not possessed by analogous domestically produced devices. These include hardware writing of code words into each cell of the storage after the power has been connected, sequential reading of all cells during the operating process for regeneration and error detection, issuance of an interrupt vector corresponding to the type of error detected, storage of the address and syndrome of an error in accessible registers, and creation of a redundant system consisting of two OZU-SKM boards. The new RAM is recommended for use in high-reliability computer systems for which rigid speed requirements have not been stipulated. This article details the operation and capabilities of the new RAM and includes a block diagram of it. Figure 1; references 2: Russian.

Automated Cash Accounting System

[Article by V. G. Kolomyts, I. V. Korolev, and A. D. Chernyavskiy]

UDC 681.3

[Abstract] The automated cash accounting system described in this article is based on DVK3M2 series-produced interactive computer systems with additional peripheral controllers for the Elektronika family of microcomputers and specially developed equipment. It consists of central and peripheral systems. The central system is based on a DVK3M2 that includes an Elektronika MS1201.02 single-board microcomputer and a KTLK board to exchange information with peripheral systems. The central system is responsible for the following: loading all necessary information (menu and permitted credit numbers) into the peripheral systems; permitting the peripheral system to work in one mode or another; receiving, accumulating, and storing information regarding paid meals; preliminary formulation of a carrier to transfer information regarding paid meals to a YeS computer; formulating documents and reports regarding a cafeteria's operation; and performing auxiliary tasks. The system also includes a reader to handle credit cards and a user panel to simplify interaction between cafeteria patrons and the computer. This article includes a schematic of the new system as well as detailed discussions of the software of the central and peripheral portions of the system and the system's functioning. Figures 1; references 3: Russian.

Elektronika MS8301 Local Area Network Station

[Article by A. M. Dyachenko, V. M. Korolev, S. I. Simanenkov, and B. V. Tseytlin]

UDC 681.324.01

[Abstract] One MS8301 local area network (LAN) station may be connected into a ring network of up to eight user terminals. The MS8301 station can transmit data at a rate of 1.6 Mbit/s, has a unidirected ring network topology, can operate with a distance of up to 300 m between stations, can operate in an LAN with up to 512 stations, can exchange data with a user terminal via a communications channel at a rate of up to 19,200 bit/s, and has overall dimensions of 350 x 448 x 233 mm. The station's data transmission environment is a series-produced Elektronika MS4101 digital data transmission device consisting of a receiver-converter, transmitter-converter, and fiber-optic communications lines. Data are transmitted along the communications live in accordance with the time-share principle. At the logical channel level, exchange is implemented by means of two types of frames. The network controller is based on two boards with series K555, K531, K556, K559, K1802, and K1002 microcircuits; its main functions are to receive, send, and process frames. The processor module of the MS8301 LAN station is based on a single-board microcomputer with 256 kbyte RAM, 128 kbyte ROM, and a speed of 600,000 operations per second. The station also includes an eight-channel asynchronous multiplexer permitting exchange in 5-, 6-, 7-, and 8-bit start-stop codes. The station features two types of monitoring and diagnostic tools, i.e., for direct monitoring of the station's operatin as it is in use and express-diagnosis programs to determine faults during repairs and preventive maintenance. A version of the station with an interface to link with user terminals having a Unibus bus is currently being readied for series production. Figures 2; references 2: Russian.

General Purpose Parallel Interface for Modular Microprocessor Measurement, Testing, and Control Systems

[Article by Yu. V. Novikov]

UDC 681.327

[Abstract] The low bus width of the external buses (16 incoming and 16 outgoing lines) and rigid assignment of the direction of transmission along the buses in the widely used parallel exchange devices, particularly in the I2, reduce the efficiency of these exchange devices.

Increasing system flexibility while reducing hardware costs requires the use of an interface with a large number of external lines and simplicity of changing the direction of transmission along these lines (i.e., the organization of two-way data exchange). The general purpose parallel interface described in this article permits information exchange in a program mode along 64 two-way lines forming a 16-bit port. Each port has its own address in the microcomputer's address space. The interface makes it possible to set up information input and output modes and output that includes verification of the correctness of information transmission along each of the four ports. Program switching of the direction of the external lines' transmission with a precision to one byte is provided. The interface's external lines are also two-way lines. To keep it from failing, the interface is provided with protection of its external lines. The input buffers have outputs with three states so as to permit multiplexing of data received from all four ports. Sixty-four information line and feed buses as well as a series of control signals are connected to the interface's 90-contact connector. The general purpose parallel interface described here contains 30 common low- and medium-integration integrated microcircuits. Figures 1.

Electronic Disk With Direct Memory Access for DVK2M

[Article by A. I. Knyazev and S. M. Shestimerov]

UDC 681.3

[Abstract] The SM5902 semiconductor external memory device is used as an electronic disk for the DVK2M. It is intended to store large data files and organize exchange during operation as part of an SM computer system. It may be considered the electronic analogue of a disk system with fixed heads and has been series-produced since 1986. The SM5902 provides self-diagnosis with detection of binary errors and correction of single and binary soft failure errors against the background of one or two permanent faults. The SM5902 uses an OST 25 795-78 common bus, operates in a direct memory access (DMA) mode, has a capacity of 4 Mbytes, and has a word length of 16 bits plus six check bits. It is capable of reading, writing, and checking writing and has an average transmission rate of at least 13.3 Mbit/s. Its access time averages not more than 1 µs/word, its cycle of addressing registers takes less than 0.75 µs, and it has a regeneration mode periodicity of no more than 7.5 µs. A system consisting of a DVK2M and SM5902 is far superior to a system consisting of a DVK2M and NGMD6022 floppy-disk drive; the former takes 12.5 minutes to translate a software file of 12 blocks written in C (i.e., for compilation, assembly, and linkage) and requires 102 seconds to copy a file of 432 blocks, whereas the system with the SM5902 only requires 1.75 minutes to complete the first task and one second to complete the second. Figures 3.

Redundant Interface for a Process Interface

[Article by N. I. Gaydashenko]

UDC 681.327

[Abstract] The PS1001 control computer system is an original system of hardware and software without any direct foreign prototypes. Its main distinctions are its increased flexibility and extensive expansion capabilities and its improved quality characteristics (i.e., its improved reliability, failure resistance, and survivability thanks to the introduction of hardware and software redundancy and in-depth diagnosis and monitoring). The redundant process interface described in this article was designed for use with PS1001 systems. The redundant process interface is connected to the system interface by a BUSO process interface and several redundant interface controllers that are mounted in the same case as the process interface modules. The redundant interface's controller may be passive, in which case it performs the functions of linking and branching signals, or it may be active, in which case it has intelligence capabilities. The interface is based on a radial bus principle and is capable of information exchange in three directions in accordance with an asynchronous query-response principle. This article presents a block diagram illustrating the place of the new redundant interface in a PS1001 system and details the new interface's features and operation, with special attention given to one- and two-way information exchange between the redundant and system interfaces. Figures 2.

Hardware-Software Interface for Exchange Between YeS1840 PC and DVK3

[Article by I. P. Statsuk, A. P. Aleksandrovich, and A. V. Zakharevich]

UDC 681.069

[Abstract] This brief article explains a method of establishing a hardware-software link between a YeS1840 PC and a DVK3. The link is based on the RS232 serial interface of the YeS1840 PC and radical serial interface of the DVK3. Only five circuits of the RS232 are used. This article charts the method of connecting the radical serial interface and the RS232 and the circuitry of the output amplifiers of the RS232 interface circuits used. It also details the role played by each of the five RS232

circuits used to effect the link. The link in question makes it possible to send and receive messages between the two computers at baud rates of 50, 75, 100, 150, 200, 300, 500, 1,200, 2,400, 4,800, and 9,600 in formats of seven or eight information bits as well as to formulate and check a parity bit or operate without one. Information blocks sent along the interface each contain both binary and standard ASCII characters. Figures 5; references 3.

A Class of Instructional Computers for Laboratory Practicum Entitled "Operating Systems"

[Article by R. I. Grushvitskiy, A. V. Preobrazhenskiy, A. V. Simonov, and V. M. Yakker]

UDC 681.322.042

[Abstract] This article describes a class of instructional computers for use in a laboratory practicum dealing with the use of operating systems of microcomputers and personal computers. The system described has been designed to instruct students in the basic principles of working with the operating system CP/M80. A local area network (LAN) architecture with a hierarchical star topology, central control, and a coiled pair of conductors as a data transmission environment was selected as the training program architecture because it combines a high degree of equipment utilization with simplicity of access to the system's resources. An additional level of file memory, i.e., an electronic disk simulator, has been added to increase the capacity, reliability, and economy of the LAN. The system is designed to address four training topics. The first, introduction to computer hardware and software, introduces the concepts of the current disk, directories, file systems, program debuggers, and resident and transit operating system instructions. The second topic, i.e., using system devices, covers the concepts of system function, character input/output (I/O), buffered I/O, console status, and program routing. Topics addressed in the third topic area (i.e., working with an operating system's file system in an applications program) include sectors, groups, extents, operation completion codes, and direct memory access addresses. The fourth topic area is development of a simple file control system. This article details the hardware used to implement the proposed laboratory practicum and includes a block diagram of this hardware as well as a schematic detailing the distribution of PC workstation memory required to use the software developed for the practicum. Figures 2; references 3: 2 Russian, 1 Western.

K564RU2 Static RAM

[Article by G. Glushkova]

UDC 681.325-181.48

[Abstract] This concise reference article details the function and operation of the K564RU2 integrated microcircuit. The K564RU2 is a complementary metal-oxide-semiconductor (CMOS) static RAM with random sampling based on 256 single-bit words. It has direct and inverted outputs and a control circuit. Eight input address buses are required to select one word. Data are read without authorization. The K564RU2 microcircuit includes outputs to execute instructions to read and write digital information and to authorize microcircuit selection. This article contains block diagrams of the circuit and tables detailing the purpose of the circuits leads, electrical characteristics, and admissible operating modes. A truth table for the K564RU2 static RAM and a table listing the time relationships of its parameters and the main characteristics of its electrical modes are also provided. Figures 2; tables 5.

Syntactic Pattern Recognition in a Class of Finite Quantum Automata

[Article by V. A. Yatsenko, Institute of Computer Science imeni V. M. Glushkov Scientific Engineering Complex, UkSSR Academy of Sciences]

UDC 681.51

[Abstract] In the past few years there has been active research in the area of creating fundamentally new approaches to information processing based on microscopic processes. The case of syntactic quantum recognition has yet to be considered, however. The authors of the study reported herein examined the task of synthesizing a quantum automaton implementing syntactic recognition of macroscopic patterns. The proposed approach is based on the idea of describing a model of the pattern in question by using a chain of very simple elements in accordance with some grammar and subsequently implementing it on a quantum level. The authors illustrate this approach by way of the example of two patterns, R_1 and R_2, represented by chains of admissible symbols from some grammar. They proceed to prove a theorem stating that syntactic pattern recognition based on a grammar with finite chains is possible by using a spin lattice system that may be described by a model that they present. The proof of the theorem presented follows from the mappability of an algorithm for recognition by a Turing machine and the subsequent feasibility of the base model presented. They then propose a soliton syntactic recognizer based on the states of a finite automaton that is formed from a finite set of solitons with different parameters. The transitions between the automaton's states are effected by mappings transforming groups of symmetries associated with the nodes of a graph. Figure 1; references 4: 2 Russian, 2 Western.

Nonparametric Combinatorial GMDH Algorithm Based on Analogue Retrieval Statements

[Article by A. G. Ivakhnenko, V. A. Chainskaya, and N. A. Ivakhnenko, Institute of Computer Science imeni V. M. Glushkov Scientific Technical Complex, UkSSR Academy of Sciences]

UDC 681.513

[Abstract] The authors of this article discuss a nonparametric combinatorial group-method-of-data-handling (GMDH) algorithm based on analogue retrieval statements. The width of the data-sampling intervals or patterns, the number and make-up of variable sets, and the number of analogues constituting a set are optimized by a rationally organized, complete successive trial of possible versions from the standpoint of a precision redefinition criterion. They assert that the best version is the one leading to the deepest minimum of the criterion. They present a method of complexing analogues (also referred to as hybridizing predictions) that entails the weighted summation of their predictions by using special splines. Figures 5; tables 4; references 17: Russian.

The Weiner-Kolmogorov Method in the Synthesis of Multidimensional Control Systems

[Article by V. B. Larin, Mechanics Institute, USSR Academy of Sciences]

UDC 519.71

[Abstract] One direction in the frequency method of synthesizing optimal linear stationary systems entails the examination of systems with several degrees of freedom. Accordingly, the problem of synthesizing a servo system has two degrees of freedom; in the case of an external perturbation, however, this same problem has three degrees of freedom. It is essential that each of these problems have its own corresponding parametrization of the set of feedback regulators. This article demonstrates that algorithms for parametrization of a set of stabilizing controllers in problems of synthesizing servo systems and systems containing external perturbation meters may be derived from the parametrization of the set of stabilizing controllers of a stabilization system synthesis problem. Expressions are presented that make it possible to formulate the problem in terms of a compensation problem and to find that solution that minimizes the H^2-norm (in other words, by using the Weiner-Kolmogorov method). References 10: 6 Russian, 4 Western.

Brightness Segmentation of Images by Neuron-Like Net

[Article by A. D. Goltsev, Institute of Computer Science imeni V. M. Glushkov Scientific Technical Complex, UkSSr Academy of Sciences]

UDC 612.821

[Abstract] The problem of image segmentation is acknowledged as the main problem in the task of creating robovision. Different image segmentation techniques, including segmentation by brightness, have been proposed. Methods of image segmentation based on brightness are divided into two main classes: discovering internal points of regions of uniformity (this is generally done by some version of the threshold section of the brightness matrix of the initial image) and identifying the boundaries of images and subsequently tracking them. This article presents an algorithm for use in identifying the brightest, longest, and most evenly illuminated regions. The algorithm's distinction lies in the fact that it organically combines operations belonging to both of the aforementioned classes of brightness-based image segmentation methods. The algorithm represents a structured neuron-like net with an ensemble organization that is intended to make a precision determination of brightness spots on an image. The main component of this net is a layer of neuron-like elements containing a regular structure that is close to the mutually exciting links between the elements constituting the layer. The net implements preliminary identification of bright, contourless, evenly illuminated regions in an image. The model's performance is verified by using it in the brightness segmentation of images of natural objects. Figures 4; references 12: Russian.

Development of Architecture of Neuron-Like Network to Recognize Shape of Objects in Image

[Article by E. M. Kussul and T. N. Baydyk, Institute of Computer Science imeni V. M. Glushkov Scientific Engineering Complex, UkSSR Academy of Sciences]

UDC 007.001.362

[Abstract] In the past few years there has been a significant increase of interest in creating, researching, and using neuron-like networks. Such networks have proved to exhibit tolerance of errors, self-organization, the ability to adapt their behavior, and the capability of making associations when retrieving information. Owing to their parallel structure they are also quite fast. One way of increasing their efficiency is to develop large, standard, simple structural units that can be used to design and construct different neuron-like network

architectures for applications. The principle structural element of such blocks is the neuron field, i.e., a subset of neurons performing one and the same function. Three types of neuron fields are used: associative, buffer, and difference neuron fields. Those subsets of neurons that are connected by mutually exciting links are termed neuron ensembles. All of the neurons in an associative field are interconnected by associative links. Two other types of links between neurons, i.e., projective and receptive, are also possible. This article describes the functions of these links and, by way of the example of a neuron-like network used to perform the task of recognizing the shape of objects in an image, illustrates how a network architecture can be constructed from these basic building blocks (units). The recognition of complex shapes by means of a neuron-like network is shown to be analogous in many respects to recognition by using the potential functions method; however, recognition based on a neuron-like network is virtually independent of the number of flags and number of "reference" images introduced during the teaching process and the formulation of the relief of potential functions on which recognition is to be based. When operation of the proposed neuron-like network algorithm was simulated on an IBM PC AT, the task of recognition took about one minute to complete. It is thought that if the same network were to be implemented on the neurocomputer developed at the Institute of Computer Science of the UkSSR Academy of Sciences, the time required for recognition could be reduced to fractions of a second. Figures 6; references 5: 3 Russian, 2 Western.

Controlling Process of Transfer in Periodically Inhomogeneous Media

[Article by S. A. Belikov, Kiev]

UDC 517.944

[Abstract] This article examines the problems of averaging a quasilinear first-order partial differential equation with coefficients that are strongly oscillating functions of phase variables. Problems of controlling an averaged system as they apply to the phenomena of transfer in periodically inhomogeneous media are examined as well. Also considered are unusual phenomena occurring in the solutions of this type of parameter-dependent system that are of a general nature with regard to nonstandard analysis. A sample problem is presented to illustrate the application of the simplest concepts of nonstandard analysis in a specific applications problem and to demonstrate nonstandard bifurcation. Methods of solving control problems are presented by way of the example of the solution of an equation describing the task of controlling a biological substance moving along a closed loop during a life support process. Figures 4; references 8: Russian.

Solving One Problem of the Minimax Estimation of the Parameters of a Linear Regression in a Nonstochastic Formulation

[Article by Yu. L. Kocherga and A. A. Kavalets, Institute of Computer Science imeni V. M. Glushkov Scientific Technical Complex, UkSSR Academy of Sciences, Kiev]

UDC 519.6

[Abstract] The problem of minimax estimation of the parameters of a linear regression has recently enjoyed a great deal of attention in the literature. Most publications on the problem hypothesize the random nature of the measurement errors, in which case the quality of the estimates of the parameters is generally guaranteed on the average on the basis of all possible realizations of the random measurement errors. In a number of cases, however, interest has been expressed in a guaranteed result obtained in a nonstochastic formulation. This article presents the solution of a problem of the minimax estimation of the parameters in a linear regression for one class of linear estimates. The problem is solved under nonstochastic assumptions relative to the measurement precision and admissible set of desired solutions. The proposed nonstochastic minimax approach differs somewhat from analogous existing approaches with respect to formulation of the problem and the results it yields. The proposed method uses a narrower class of estimates and a different type of constraints on the set of admissible solutions. Furthermore, the quality criterion is maximized not only with respect to realization of the measurement noise but with respect to the regression parameters. From the standpoint of the resultant minimax estimation, the main difference between the proposed approach and analogous existing approaches is that the proposed approach produces a guaranteed result by contracting the spectrum of the information matrix whereas in other existing approaches the result is obtained by shifting the information spectrum to the side of reducing the amount of its conditionality. It is thus shown that in the case of poor conditionality of the information matrix and a high level of measurement noise, the proposed minimax approach makes it possible to achieve a result with a guaranteed precision. References 8: Russian.

Designing and Implementing Distributed Enterprise Management Network

917G0016A Riga AVTOMATIKA I VYCHISLITELNAYA TEKHNIKA in Russian No 4, Jul-Aug 90 (manuscript received 28 Nov 89) pp 54-57

[Article by Syao Debao]

UDC 681.324

[Abstract] This article examines the problems of creating an integrated enterprise management system combining a series of subsystems (subdivisions) into a unified whole. The system in question provides the following: management in each department and shop and formation of an information structure to manage the entire enterprise; monitoring of all production processes (including automatic data collection, analysis, and processing; issuing of real-time information about the status of processes and warnings of accidents); and unification of all of the shop-monitoring subsystems and the overall enterprise management system. Two methods of implementing such a system are examined. The first is based on microcomputers with a high degree of specialization and forming a multilevel hierarchical distributed monitoring and management system. In this version of the system, IBM PC and APPLE II personal computers are connected to an Intel 86/300 in a starlike configuration. Tests of the system at a textile factory revealed the drawbacks of a starlike configuration: too great a load on the Intel 86/300 microcomputer and telecommunications controller, further complications if the number of stations is increased, low system reliability, and nonoptimal distribution of functions. The second (successful) version of the enterprise management system discussed is a two-level system. Its upper level, i.e., its information level, manages the entire plant, while the lower level is the level at which production processes are monitored. The two levels are connected by an Ethernet 3+ network, and the system's software operates in an environment with the MS-DOS 3.1 operating system. The server used in this network makes it possible to provide diverse service to more than 50 users. The control microcomputer (STD) used has a modular architecture conforming to distributed processing. The use of a modular design makes the system well suited to the creation of multimachine and distributed systems. Detailed information on the system's architecture, hardware, and software is provided. Figures 6.

Synthesis of Test Experiments for Some Classes of Automata

917G0016B Riga AVTOMATIKA I VYCHISLITELNAYA TEKHNIKA in Russian No 4, Jul-Aug 90 (manuscript received 15 Feb 90) pp 59-65

[Article by N. V. Yevtushenko and A. F. Petrenko]

UDC 519.713:681.324

[Abstract] Test experiments with automata serve as the basis for designing tests detecting errors (faults) in different types of engineering systems. The classic formulation of the problem of synthesizing such tests has been limited to complete (completely determined) automata. Other classes of automata have only recently begun to be examined. There is still, however, no method for organizing a minimum-length test experiment for such automata. This article proposes methods of designing a test experiment for completely and partially determined and "partially indeterminate" automata and sets of them. The problem of interpretations of the expansion of a partial automaton is also examined. The proposed methods are based on the assumption that faults do not increase the number of an automaton's states. The authors demonstrate that in the classic case, the proposed method may lead to a more economical experiment than one designed by using the existing method. They further demonstrate that when their approach to expanding a partial automaton is used, the class of guaranteed detectable errors is limited solely by the systematic errors (those that do not depend on time or other factors), which in most cases satisfies practical requirements. References 5: Russian.

Architecture of a Uniform Distributed Computer System With Shared Resources and a Network Operating System

917G0030A Riga AVTOMATIKA I VYCHISLITELNAYA TEKHNIKA in Russian No 6, Nov-Dec 90 (manuscript received 19 Feb 90; after revision 7 Jun 90) pp 3-9

[Article by A. A. Malyavko and L. P. Satyshev]

UDC 681.324

[Abstract] The principles of enabling the collective use of computer resources may be divided into two general classes: networks of personal computers and network computer systems. The first class of principles entails multiuser access to shared disk storage devices, high-speed printers, and other peripherals. Network computer systems, on the other hand, are based on the principle of the collective use of computers, peripherals, and data so as to make the given distributed systems more efficient to operate and to enable distributed processing. This article describes work that has been done at the Novosibirsk Electrical Engineering Institute in order to create network computer systems with shared resources. This work was based on the following principles: 1) all resources of the computers in the given computer system (including peripherals, processors, external memory, and data files) represent a single pool of resources intended for collective use; 2) a specified amount of resources constituting a user computer environment is allocated for each user to perform his computing tasks; 3) these computer resources are used by means of access methods for the different types of resources that, while specific to

the various types of resources, are uniform within the framework of the computer network; and 4) the user interface in the system created should be simple, understandable, and familiar to and convenient for network users. These principles have been used as a basis for the development of virtual computer systems controlled by a single network operating system that are developed by using the methodology of an abstract computer. The proposed approach to designing uniform distributed computer systems has been tested in actual practice and may be used when creating diverse universal and special network computer systems and distributed data processing systems. Figures 2; references 12: 11 Russian, 1 Western.

Method of Interval Accounting for Nonequivalence of Heterogeneous Parameters of Alternative Versions of Complex Computer Systems

[Article by A. Yu. Shcheglov]

UDC 621.37/39

[Abstract] The existing approaches to accounting for the nonequivalence of the heterogeneous parameters of alternative versions of complex computer systems are based exclusively on the physical sense of the criteria. No allowance is made for the change in quality indicators in the set of alternatives being compared; this affects the reliability of the results of multiparametric optimization. This article proposes a method of increasing the reliability of multicriterial optimization of alternatives by means of interval-by-interval accounting of the heterogeneity of the parameters of different versions. This approach makes it possible to rank (i.e., weight) indicators by proceeding from two criteria. These criteria are the physical meaning of the parameter and the value of the alternative's quality indicator. The use of the proposed method is illustrated by way of the example of selecting the best version of a real-time computer system. The proposed method of interval accounting for homogeneous parameters makes it possible to increase the reliability of the results of multicriterial optimization of alternative versions of computer systems. The method may be used in an adaptive structured decision-making procedure (in which the "weakest" expert appraisal is used), where, in each successive stage of the comparison, the decision maker divides the intervals into subintervals and ranks the subintervals of the change in the values of the parameters of the alternatives. Figure 1; tables 2; references 9: Russian.

Universal Network With Hybrid Access

[Article by V. F. Baumgart and D. K. Zibin]

UDC 681.324

[Abstract] This article proposes several revisions of and additions to second-level local area network (LAN) protocols that have been developed in accordance with International Standards Organization (ISO) recommendations ISO 8802/3. The proposed changes are intended to maintain all of the existing positive features of LANs while eliminating many of their shortcomings. The following are among the proposed changes discussed: develop guaranteed time periods within which frames will be delivered, introduce priority servicing under conditions of increased loads, introduce a system of priorities to control data streams, and develop centralized monitoring systems that can be used with LANs. These changes are said to be easily achievable without greatly complicating existing LAN hardware. They are further praised as ways of significantly expanding the range of potential applications of LANs. Tables 1; references 7: 5 Russian, 2 Western.

VLSI-Based Bit Conveyers for Digital Signal Processing

[Article by V. D. Baykov and S. D. Bulgakova]

UDC 681.325.5:518.5

[Abstract] In digital signal processing systems, continuously arriving input information first passes through analog-digital converters. Most often, these analog-digital converters output digital information in a sequential code beginning with the higher-order bits. Because parallel digital processors generally operate faster than analog-digital converters do, the parallel processing of information in a digital processor may be combined with sequential bit-by-bit input of the information. Such devices are termed bit conveyers. Most works devoted to creating bit conveyers use a redundant counting system (for example, with the digits 1, 0, 1) for their computations. This article proposes using the traditional binary counting system with the digits 0 and 1 to create bit conveyers. This proposal is made so as to permit extensive use of standard modules. The authors propose algorithms for making functional transforms that make it possible to combine the bit-by-bit input of information with the parallel processing of that same information. These transforms include those elementary functions, coordinate transforms, computations of statistical characteristics, and solutions to linear equation systems that

are characteristic for digital signal processing systems. The article includes time diagrams and gives relative estimates of the proposed bit conveyer's precision and speed. Figures 2; tables 5; references 5: 2 Russian, 3 Western.

Nonexpanding Blocking Protocol for Parallel Access to Paged Data Base

[Article by S. V. Martyanov and S. B. Meleshchuk]

UDC 681.03

[Abstract] Blocking/deblocking protocols are often used in synchronizing parallel tasks and in updating a common data base. A blocking protocol is termed expanding if the size of the blocking unit (for example, a page of the data base) exceeds that of the logical blocking unit (for example, a record). Increasing the size of the blocking unit worsens the parallelism indicators, whereas reducing it improves the level of parallelism and simultaneously increases the amount of message-synchronizing processes circulating in the system. At the same time, however, implementing a nonexpanding protocol constitutes a complex problem because of the need to monopolize the page being updated by the data base manager's modules. The nonexpanding protocol described in this article is a compromise protocol in that it provides blocking on the level of individual records and simultaneously monopolizes the page being updated so as to ensure the correct functioning of the data base manager's modules. Another advantage of the proposed protocol is that it affords the capability of using most programs with expanding blocking in the old algorithm. Operation of the proposed compromise protocol together with route changes revealed a radical reduction in the frequency of mutual delays of user processes related to the freeing of pages. The frequency of the occurrence of dead-end situations was shown to remain rather low, although there was a slight increase because of the addition of a level of page blocking. In addition, execution of the compromise protocol resulted in an additional load on the central processor compared to what occurs when a page protocol is used. Tables 2; references 5: Russian.

Graphic Representation of Combinational Circuits

[Article by A. D. Zakrevskiy]

UDC 519.95

[Abstract] When automating the stage of logic design in an automated system to design discrete systems it is useful to have a graphic representation of the results obtained. Furthermore, it is desirable that this representation be in the form that is most convenient for visual perception. One problem that arises during the logic design stage is an optimization combinatorial problem of the rational arrangement of the elements of the circuit being synthesized in the plane of the image and the routing of connections between them. The optimization criterion in this process is very different from those that are used during engineering design. This article proposes a method of solving such a problem that is suitable for combinational circuits. The problem examined is based on the assumption that there are no loops in the circuits and that their elements realize symmetrical boolean functions (i.e., they remain unchanged when the arguments are reformulated). The proposed solution method includes the methods of staging the circuit, ordering and arranging its elements in stages, mapping the circuit's connections (including vertically and horizontally), and plotting a graphic image of the combinational circuit overall that is optimized with respect to visual perception criteria. The proposed method makes it possible to produce a graphic representation of combinational circuits of rather large size by formulating a description of the combinational circuit in a computer's memory in the form of a discrete two-dimensional coordinate system. Subsequent display of the circuit on screen or on paper does not entail any combinatorial problems; rather, it may be based on a standard graphic utility method (for example, a method regulating the scale of the image or outputting the image in parts in the case of a very large circuit). Figures 3; references 1: Russian.

Multichannel Signature Analyzer for Built-In Testing of Digital Circuits

[Article by V. N. Yarmolik and I. V. Kachan]

UDC 681.326.7

[Abstract] Single-channel signature analyzers described by a primitive characteristic polynomial are generally used for self-testing of digital circuits. When the circuit that must be tested contains multiple outlets, the single-channel signature analyzer must be combined with a multiplexer successfully connecting the check points to the signature analyzer's input. Because of the number of check points that exist in the actual digital circuits that must undergo such testing, using a single-channel signature analyzer to test digital circuits requires the expenditure of large amounts of time. Multichannel signature analyzers have been developed to overcome this problem. Multichannel signature analyzers with internal and external modulo 2 adders exist; they, too, are fraught

with shortcomings. Among their drawbacks is the fact that multichannel signature analyzers with a large number of inputs require significant hardware expenditures and an unjustifiably long signature; for example, a multichannel signature analyzer with 300 inputs must contain 300 D-triggers and a signature 300 bits long. For this reason, the authors of this article propose a new method of synthesizing a simple two-stage multichannel signature analyzer that is shown to be as reliable as a classic single-channel analyzer described by a primitive polynomial. Furthermore, the number of inputs that the proposed analyzer can have is not restricted to the number of bits that it possesses. Figure 1; references 9: 4 Russian, 5 Western.

One Method of Parallel Modeling of Logic Circuits

[Article by A. I. Andryukhin]

[Abstract] The modeling of logic circuits on currently existing computers having a von Neumann architecture is a slow process because simultaneous and parallel changes in a circuit are simulated sequentially, hence the importance of increasing the speed of logic modeling on different levels (valve, functional, etc.). Parallel modeling is one well-known method of accelerating the process of logic modeling. Computers can simultaneously and independently execute logic operations on a whole word (generally 2 or 4 bytes or, in the case of YeS computers, on a field of 256 bytes). This concise communication describes an algorithm and data structures for parallel modeling of pseuodocombinational logic circuits. Also presented are data from experiments on the effectiveness of program implementation (in the language Assembler) of the proposed algorithm on a YeS-1045 computer. In view of the dimensions of the circuit modeled, the proposed algorithm requires only a small amount of computer time to execute. Tables 1; references 4: 3 Russian, 1 Western.

Increasing Reliability of Storages by Giving Them Fault-Tolerance Properties

[Article by I. A. Baranov and A. A. Shikin]

[Abstract] This abstract of an article deposited in the All-Union Institute of Scientific and Technical Information (VINITI) details the solution of a problem of increasing the reliability of a storage by giving it fault-tolerance properties. For the errors $0 \rightarrow 1$ (the appearance of false units) and $1 \rightarrow 0$ (the loss of units) caused by memory cell faults occurring before some information is written into the cell (case a) and while that information is stored in the cell (case b), the authors formulate and solve the problem of finding that correcting code and constructing that multiple error correction algorithm that will make it possible to render the memory device fault-tolerant. Since the correcting capabilities of the codes found are inherent to every q-th bit, the stability indicators achieved will be extended to each of n bits for a memory device consisting of q n-bit words.

Approximate Synthesis of Optimum Control of Systems With Distributed Parameters by Using Incomplete and Imprecise Measurements

[Article by Z. M. Gasanov, Dnepropetrovsk Institute of Railway Transport Engineering]

UDC 681.513

[Abstract] The problem of optimum control of systems with distributed parameters given incomplete measurement of their state has attracted many authors' attention owing to the fact that it is virtually impossible to completely measure a distributed system. In the case where a state with random noise is being measured, it becomes necessary to address the problem of designing an optimal regulator that takes the noise into consideration when using the measurement results. This problem is generally solved by using what is called the subdivision principle. According to that principle, the state of the object being measured is initially measured by using a Kalman filter, after which a regulator using the filter's output to control the object is constructed. The present article presents another approach to the problem. This approach calls for designing and constructing an approximately optimal regulator that uses the readings from measurement sensors at individual isolated points of a heat conductivity process. The measurement noise is filtered during the process of shaping the control signals. References 8: Russian.

Sign Interpretation of Similarity for the Compatibility of a Set of Objects

[Article by S. K. Dulin, Moscow]

UDC 519.113.115:681.3

[Abstract] Heider and others after him have developed the idea of ascribing signs to the links between like objects of some set on the basis of an analysis of their similarity or some expression of an attitude of sympathy, respect, etc. The study reported herein examines the process of deriving the sign of the link between a pair of objects possessing a number of selected attributes. The author begins by formulating the problem of distributing n objects possessing m flags (attributes) among N classes so that each class ends up containing those objects that are most similar to one another from the standpoint of a set of k selected attributes ($k \leq m$). He introduces the concept of the similarity function of the objects in question (i.e., their similarity from the standpoint of the set of selected attributes k). Next, he examines the process of selecting the similarity threshold of the function f (i.e., that value α of the function f before which objects are considered dissimilar, i.e., when $0 \leq \xi_{ij} \leq \alpha$ with respect to the k attributes and after which they are considered similar, i.e., $\alpha < \xi_{ij} \leq 1$). He then goes on to examine a more general case in which two threshold values (α and β) are specified (the sign "-" is assigned to all links where $0 \leq \xi_{ij} \leq \alpha$, a "0" is assigned if $\alpha > \xi_{ij} \leq \beta$, and the sign "+" is assigned if $\beta < \xi_{ij} \leq 1$. The indifference range (α, β), within which the value of the link is discussed in terms of similarity with respect to k selected attributes, is thus introduced. Unlike previous discussions of the structures with sign links, which do not examine the formulation of the structures themselves, this study proposes a method of calculating the strength of the link and establishing the sign of the link by specifying threshold values. While the proposed similarity function is not the only method of determining the proximity of objects with respect to a series of selected properties, it is important in that it illustrates the possibility of making such a determination. The author concludes by presenting a problem that is the inverse of the initial formulation and that may be useful in such applications as subdividing n objects into N classes based on similarity with respect to the attribute in question given the requirement that the least expenditures (in the sense of total change in the values of the similarity function f) be made. References 8: 5 Russian, 3 Western.

Ordering Generalizing Sets in Structured Image Recognition

[Article by S. I. Akimov, Moscow]

UDC 519.6

[Abstract] In structural image recognition, objects are characterized above all by the system of relationships in the set of their primitive elements. Different data structures are used to represent this system of relationships. Some of these structures lend themselves to representation in terms of a hierarchical (treelike) structure of the links between elements in the object, with the data represented in the form of a chain of symbols. While these methods have been rather well studied and examined within the framework of the linguistic approach, they are limited from the standpoint of expressiveness when it comes to representing relationships. Another more general form of describing the structure of an image is to specify it in terms of a graph (matrix) of relationships; however, the complexity of this data structure and the fact that it has not been fully developed to the user level limit its usefulness as well. Trees, matrices, webs, and similar structures occupying an intermediate position between a chain and graph of relationships have

Theory of Computation

been developed in an attempt to better describe the relationships among the set of an object's primitive elements. The approach to structured image recognition that was proposed by R. Kharalik has succeeded as a general theory of the subject. According to that approach, the recognition process is reduced to covering the objects under observation with generalizing sets and plotting a relationship system formulating a decision rule. The study reported herein demonstrates that the set of homomorphisms (combining functions) involved in formulating the covering of recognition objects is an algebraic structure, i.e., a lattice. The study further shows that the set of elements of a covering (and consequently, the generalizing sets) may be partially ordered and that binary operations may be performed on the set, thus transforming it into an array. Finally, it is demonstrated that ordering the generalizing sets and discerning their algebraic structure makes it possible to construct effective recognition algorithms. The author illustrates the use of the proposed recognition algorithm throughout all of its steps (instruction, recognition, refinement, and further instruction) by way of the example of the problem of recognizing plane images consisting of continuous curves and lines and by using the ends and intersection points of the lines and selected characteristics of the position of these elements in a plane as the images' primitives. Figures 3; tables 3; references 9: Russian.

Recovery of Moving Three-Dimensional Objects

[Article by Mechislav A. Klopotek, Warsaw]

[Abstract] Research on the recovery of moving objects has not been developed extensively; instead, it has concentrated primarily on easily trackable points (such as boundary and other characteristic points). The recovery of smooth curves, i.e., recovery of the position of the untrackable elements of a scene, has received relatively little attention. One previously published study examined smooth curves rotating continuously around a fixed direction with different time intervals between the "photographs" in the orthogonal projection. This work demonstrates the possibility of recovering smooth three-dimensional curves in a broader range of applications than had been predicted previously. The goal of the study reported herein was to confirm this possibility by examining cases that are closer to reality, i.e., the study examined not only a constant rotation but free motion as well and not only for an orthogonal projection but for a central (and stereoscopic) projection also. The significance of stereoscopic, central, and orthogonal projections and the two stages in recovering three-dimensional curves are examined. The remainder of the study is devoted to a detailed analysis of the problem of recovering various types of orthogonal and central (perspective) projections and the problem of recovering objects in the case of stereoscopic vision. On the basis of his analysis, the author determines that the following numbers of trackable points and frames are required to recover three-dimensional curves given the various types of projections and vision: an orthogonal projection with free rotation (two trackable points and four frames); an orthogonal projection in the case of free rotation around a fixed direction (two trackable points, four frames); an orthogonal projection in the case of free motion (three trackable points, three frames); a central (perspective) projection in the case of a motion that is free of rotation (two trackable points, two frames); a central projection with even rotation (two trackable points, five frames); a central projection with free rotation around a fixed direction (three trackable points, three frames); a central projection with free motion (four trackable points, three frames); and a free motion in the case of stereoscopic vision (two trackable points and one frame). Table 1; references 4: Western.

Algorithms for Recognizing Partially Ordered Objects and Their Application

[Article by A. B. Frolov and E. Yako, Moscow and Budapest]

[Abstract] This article examines algorithms for compressing the description of a function with a finite partially ordered domain and linearly ordered range of values and for interpreting its meaning on the basis of a specified element from the domain in accordance with the compressed description. The complexity of the algorithms derived for several classes of tasks is estimated, and analytical and circuit design methods of implementing the proposed recognition algorithms are examined. The application of the recognition algorithm in microbiology is illustrated by way of the example of compression of a genetic code, and the algorithm is also recommended for use in performing such tasks as classifying and recognizing more complex polyamide chains and other microbiological objects that lend themselves to partial ordering and that are studied as elements or subsets of some finite-dimensional space. In short, the algorithms are deemed suitable for use in recognition tasks examined in finite partially ordered sets in which a number of nonintersecting classes have been identified (their designations may be viewed in terms of functions defined on the basis of the union of these classes and undefined with respect to the remaining elements of the partially ordered set). They thus represent an approach to dividing the subsets of a k-unit n-dimensional space or

any other partially ordered sets. As such, they may be used for purposes of decision making in expert systems, logical design, and technical diagnosis. Figures 2; references 12: 7 Russian, 5 Western.

KOGR—Cognitive Graphics System: Development, Implementation, Use

[Article by V. A. Albu, V. F. Khoroshevskiy, Moscow]

UDC 519.007

[Abstract] The KOGR system is representative of a new direction in the field of knowledge representation, namely, cognitive graphics. The concept of cognitive graphics may be discussed in terms of a set of methods and equipment for representing knowledge and working with it at the level of the graphic image (static and/or dynamic). Overall, the architecture of cognitive graphics systems correlates well with that of classical knowledge-based artificial intelligence systems; however, they represent a new step in the theory and practice of representing and manipulating knowledge. The representation of knowledge in cognitive graphics is itself dual in nature: on the one hand, such systems explicitly and singularly describe both graphic primitives and complex graphic gestalts; on the other hand, the descriptions themselves may be considered declarative structures processed by using special decision procedures and in turn based on knowledge about the laws governing transformation of the resultant "pictures". The main objective of the KOGR system is to provide tools to support the development of cognitive graphics systems intended for different purposes. As such, it accomplishes the following:

—formulation and on-line visualization of complex images based on a set of graphic primitives accessible to the system;

—maintenance and correction of the image base;

—generation of new graphic objects and scenes based on the current status of the image base by means of "traditional" computer graphics and/or text description of the required result;

—acquisition of knowledge regarding new objects and the links between them both by default and in an "interpretation" mode;

—formulation of scenarios of animation of the image base based on knowledge about some subject area;

—dissociation of cognitive graphics systems developed within the framework of the KOGR to permit their independent use; and

—support of the functions of providing help, explanations, and instruction needed to make them convenient for users. The KOGR system may be used in a variety of applications, including the following: geometric design of production programs as in software tool systems for designing expert systems, visualization of the results of the operation of various expert systems in various applications areas (such as in psychological testing) currently under development at the Institute of Artificial Intelligence Problems at the Computer Center of the USSR Academy of Sciences, and to support cognitive graphics in the TEKRIS (text-drawing) system. Table 1; references 21: 9 Russian, 12 Western.

Programmable Shell of Expert System ZNATOK and Problems of Its Theoretical-Categorical Description

[Article by A. V. Zhozhikashvili, V. L. Stefanyuk, Moscow]

UDC 517.11

[Abstract] Previously, the authors of this article developed a theoretical-categorical language, the base element of which is the pattern. The language is based on comparing patterns with one another and generalization. In the present article they use this language to present a theoretical-categorical description of a new development, i.e., the programmable shell of the ZNATOK expert system. Their primary objective in so doing is to demonstrate the use of their language in describing several aspects of the ZNATOK programmable shell and the applications expert system SVAYa 1.0 that has been based on it. The authors developed the SVAYa 1.0 system to design bearing pile structures when constructing bridges and other engineering structures. The authors compare the pattern and products in terms of the theory of categories, discuss the main operating features of the ZNATOK programmable shell, present a theoretical-categorical description of the knowledge included in the SVAYa system and its output equipment, and discuss the use of attached procedures. They conclude that discussing a system that has already been created and is now in operation in terms of a theoretical-categorical description is useful for several reasons. First, because the language in question uses concepts from the field of artificial intelligence, it can describe structures that appear to be very different externally from the standpoint of a set of unified principles. Second, using a formal description of a system that is independent of the data representation method used makes it possible (during programming) to separate procedures for working with data from those higher-level procedures that are not connected to the form in which the data are

represented (which makes it possible to move to a more complex form of rule without disturbing the remaining portion of the system, but instead only substituting the comparison procedure and some few others). Third, translating concepts and algorithms related to expert systems into a theoretical-categorical language makes it possible to use existing automatic knowledge storage and instruction algorithms. References 8: Russian.

Fuzzy Inference Parallel Processor for Situation-Based Expert Systems

[Article by L. S. Bershteyn, V. M. Kazupeyev, S. Ya. Korovin, and A. N. Melikhov, Taganrog]

UDC 681.3

[Abstract] Most modern expert systems permit the processing of insufficiently definite information to one degree or another. In the simplest case, the indeterminacy is allowed during the decision-making process by recalculating the truth (certainty) coefficients of the knowledge base's rules. This inference scheme is effective when absolute certainty of the rules used is lacking but the rules themselves are described exactly. A number of developments related to fuzzy inference processors for decision making in expert systems exist. All of them are intended to make some modification or other in a composite fuzzy inference based on the minimax composite of some input fuzzy set and some fuzzy "input-output" relationship describing the decision-making rule. The present article describes the architecture of a fuzzy processor to support decision making in a special class of expert system, i.e., the situational expert system. In such systems decisions are based on fuzzy recognition of some input situation by comparing its description with that of some classes of fuzzy situations characterizing the status of the decision making object. This new inference method is rather simple to implement and possesses a number of advantages over the composite inference: there is no need to make a linguistic interpretation of the fuzzy sets derived from the composite, and the amount of information that must be stored is reduced since making a fuzzy composite inference requires storing at least one matrix of the "input-output" relationship for each rule or else deriving it during the decision making process. When compared with the best-known composite fuzzy inference processor, the situational fuzzy inference parallel processor described in this article operates about three times as fast when a two-level inference scheme is used and about an order of magnitude faster when a one-level inference scheme is used. In addition, since the standard classes of comparison situations used with the new processor are stored in peripheral memory, the number of situations used has no bearing on the processor's time characteristics and is
limited solely by the production capabilities of the processor's manufacturer. Figures 5; references 9: 3 Russian, 6 Western.

Completeness and Effectiveness of Navigation Operations in Decentralized Intelligent Systems

[Article by Ye. K. Gordiyenko, V. N. Zakharov, V. Yu. Kirillov, and A. Yu. Mironov, Moscow]

UDC 681.142.2

[Abstract] The problems of the completeness and effectiveness of the navigation operations entailed in sending messages in a decentralized computer environment are analyzed for three standard tasks falling into the category of processing knowledge in intelligent systems. These tasks are adding and revising information in an information environment, performing a pattern retrieval, and making a logical (deductive) inference. The authors also examine two possible methods of storing information in modules of the computer environment when special wave-type languages are used. It is concluded that since the structure of data exchange between the different processors in a network is unpredictable, each processor must have the capability of linking with any other processor or group of processors in the maximally efficient manner. Meeting this requirement when a system contains a large number of processors results in significant complication of the instruction set for each processor and reduces the operating efficiency of the system overall. It is therefore deemed advisable to furnish each working (master) processor with its own "secretary" or communicative processor and with a switch structure. A navigation instruction set would then be provided to enable the aforementioned processors to perform artificial intelligence-type tasks. Figures 3; references 7: 5 Russian, 2 Western.

Parallel Microprograms as Formal Systems

[Article by S. M. Achasova]

UDC 681.322

[Abstract] A parallel-type algorithm system termed generalized permutation algorithms has been proposed elsewhere. This proposal gave rise to the precise term "parallel permutation algorithms". Such algorithms are intended for efficient processing of large data files. One class of parallel permutation algorithms, namely, stationary permutation algorithms (they do not change the size of the file being processed), have been termed

parallel microprograms. The parallel microprogram may be interpreted by a network of automatons in two possible network operating modes, synchronous and asynchronous. Consequently, two types of interpretations of parallel microprograms, namely, synchronous and asynchronous, are possible. In a synchronous parallel microprogram, all microinstructions (permutations) are executed in each step simultaneously. In an asynchronous interpretation, each microinstruction is executed as it becomes ready for execution. In an asynchronous interpretation, parallel microprograms are classified as formal systems or substitution systems. This article determines the place of asynchronous interpretations of parallel microprograms among formal systems and shows how the results obtained in the theory of formal systems may be used for parallel microprograms. The author formulates an additional requirement for a parallel microprogram as a formal system proceeding from the fact that a microprogram must yield the same result in both asynchronous and synchronous interpretations for any set of source data. A semantic line between synchronous and asynchronous interpretations of a parallel microprogram is demonstrated that makes it possible to establish the properties of one interpretation based on the properties of the other. Figures 3; references 11: 7 Russian, 4 Western.

A-Covers for Context-Free Grammars

[Article by V. N. Glushkova]

[Abstract] In all formal descriptions of programming languages, the result of syntax analysis is considered the basis for specifying the language's semantics. It is more convenient to link semantics with the context-free grammar used to describe the language rather than with a well-analyzed grammar. This is because the initial context-free grammar identifies the syntactic structures and reflects the link between them in a manner that is adequate to the sense of the linguistic concepts, whereas the transformed grammar is semantically overloaded with senseless rules and symbols that have been introduced exclusively for speed of analysis. Using one grammar to perform a syntactic analysis and then using another, "semantic", grammar in the later stages of compilation requires that the grammars used be similar to one another in some sense. The task of formally determining the identity of two grammars has long been a topic studied in research on the theory of grammars. The most productive concept for use in such research is that of the cover of grammars, which was introduced by J. Reynolds in 1968 and later developed by others. This article expands the relationship of the cover for context-free grammars and introduces the concept of an automaton or A-cover in which finite transducers with bilateral access to the output tape are used to determine the correspondence between the two grammars and update the outputs. The author demonstrates the existence of A-covers for grammars derived after two transformations, namely, left-factoring and elimination of left recursion. References 6: 2 Russian, 4 Western.

Statistical Assessments of the Possibility of Discovering Parallel Structures in Sequential Programs

[Article by VI. V. Voyevodin]

[Abstract] The increasing use of high-productivity parallel computers has made it necessary to address a whole series of problems related to optimizing the form and efficiency of source programs. A number of problems have also appeared that are connected with the efficient transfer of existing programs to parallel computer systems and with developing new programs for parallel computers. This article examines the actual capabilities of statistical analysis in finding the parallel characteristics of sequential programs. The author presents extensive statistical materials that he obtained by analyzing a set of programs (constituting more than 12,000 instructions) written in FORTRAN IV to solve minimization, linear algebra, and mathematical physics problems. The programs were subjected to a statistical analysis based on a system that was developed in the Computer Systems Laboratory of the Computer and Computer Systems Department of Moscow State University in cooperation with the Computer Mathematics Department of the USSR Academy of Sciences. Selected bottlenecks and many typical fragments of programs are presented. The following topics are covered: the structure of do-cycle entries, the capacity and external nesting of do-cycles, the maximum number of embedded cycles in a program, the structure of bodies of internal cycles, the structure of the entry of alternative statements into cycles, the form of assignment statements in cycles, the number of operations (statements) in cycles, the structure of do-cycles' characteristics, conditions in alternative statements, and the structure of index expressions. Figures 2; tables 12; references 7: 4 Russian, 3 Western.

Section Model of Parallel Computations

[Article by M. K. Buza and L. F. Zimyanin]

[Abstract] The development of parallel computer systems necessitated the development of a number of

models of parallel computations and the creation of practical parallel programming systems based on them. This article proposes and examines a method of organizing parallel computations that is specified in the form of a coordination system (synchronization scheme). The model is based on an asynchronous principle of controlling parallel processes over a common memory. The section model outlined is presented as a base model for designing parallel programs for YeS microprocessor configurations controlled by a system of virtual machines. The model makes it possible to describe parallel computations by using a method that directly models the requirements of the problem at hand and enables the description of a set of processes with a complex interaction. It is also effective for use in designing algorithms for a wide class of problems. References 8: Russian.

Visual Style of Programming: Concepts and Capabilities

[Article by G. Ye. Khlebtsevich and S. V. Tsygankova]

UDC 681.03

[Abstract] This article reviews the development and current status of visual programming. Visual methods of programming are distinguished from conventional methods, which require one-dimensional textual or static representations, in that they propose a more natural route of expressing algorithms and data structures than is possible by linear languages. Visual programming began only recently but has already undergone significant development. Elsewhere, visual languages have been subdivided into languages for processing visual (graphic) information and languages intended for processing objects that, while not visual initially, are presented in visual form to make working with them more convenient. Such languages are termed languages that support visual interaction. Visual programming languages are two-dimensional iconic languages that permit the development of programs that accomplish the following: 1) select images that visually represent data structures and necessary variables; 2) select a desired algorithm in a logically structured multi-dimensional form; 3) observe the execution program, and see the results; and 4) actually see where and when incorrect actions are occurring in the event that a program does not produce the expected results. The author proposes classifying visual languages in four categories: 1) languages supporting visual interaction, 2) visual programming languages, 3) languages for processing visual information, and 4) iconic languages for processing visual information. The following systems supporting visual interaction are described: Iconlisp, Tinkertoy, MOPS (Modular Programming System for System Software), PegaSys, Pigsty, AdaGRAPH, and VICON. The following visual programming systems are discussed: FPL (First Programming Language), Vips (Visual Programming Synthesizer), STILE (Structure Interconnection Language and Environment), HI-VISUAL, STL (Show and Tell Language), CATY, BLOX, and Pict. References 20: Western.

Unexpected Properties of Competition Among Biological Species in a Variable Environment

[Article by V. G. Ilichev, candidate of physical and mathematical sciences, Rostov State University]

UDC 62-501.72:57

[Abstract] A periodic change in factors is the simplest version of a variable environment. Even in such a case, however, nontrivial effects in population dynamics may occur. The field of theoretical ecology is currently in the midst of an active search for unexpected dynamics of modes, i.e., "paradoxes" explaining the mechanism of the origins of selected biological phenomena and catastrophes. The study reported herein proposes a special class of models, termed D-systems, in which periodic delta-functions have been selected as coefficients. This class of models makes it possible to find, in virtually explicit form, asymptotic modes and thus discover a rich collection of paradoxical phenomena in the dynamics of competition among species. The dynamics of an isolated population is examined in terms of the model developed by Contois. Next, using his own concept of D-systems, the author examines the growth rates of populations in terms of T-period delta-functions and proceeds to describe the dynamics of competing species. His analysis results in three paradoxes related to the dynamics of competition among species, i.e., the displacement of higher species by lower species, the intransitivity of displacement, and the intransitivity of coexistence. Figures 4; references 8: 6 Russian, 2 Western.

Algorithms for Performing Selected Quadratic Programming and Minimax Estimation Tasks

[Article by V. N. Solovyev, candidate of physical and mathematical sciences, Moscow Aviation Institute imeni S. Ordzhonikidze]

UDC 519.853.32

[Abstract] This article presents an algorithm for performing the task of quadratic programming with a Stieltjes matrix, which is frequently encountered after

Theory of Computation

inverting a matrix with non-negative elements. After formulating the problem, stating the necessary and sufficient conditions of optimality, and providing analytical examples, the author turns his attention to the problem of minimax estimation. A three-step quadratic programming algorithm is presented. References 9: 8 Russian, 1 Western.

One Group Choice Problem

[Article by Yu. A. Bobrovskiy and O. K. Ilyunin, candidates of technical sciences, Kharkov Electronics Institute; B. V. Novikov, candidate of physical and mathematical sciences, and A. Ya. Rapaport, Kharkov State University; and M. L. Sodin, Institute of Low-Temperature Physics and Technology, UkSSR Academy of Sciences, Kharkov]

UDC 65.01

[Abstract] One important problem within the framework of the problem of group choice is that of constructing algorithms to find the median in certain classes of binary preference relationships in a specified set of alternatives. When solving the most useful and interesting of these problems, practitioners must be satisfied with either algorithms of the type based on the method of branches and bounds or heuristic algorithms, which are generally not optimal. In the present article the metric approach is applied to a class of mixed models in which individual preferences are represented in terms of linear sequences and the group preference is represented in terms of dichotomous ranking-type relationships corresponding to the isolation of the subset of preferable alternatives from the set of alternatives. Optimal algorithms for determining the set of preferable alternatives that are not reduced to successive trials are derived. References 10: 7 Russian, 3 Western.

Method of Optimizing the Physical Structure of a Numerical Data Base in an ASU

[Article by A. A. Asratyan, candidate of technical sciences, Yerevan Polytechnic Institute]

UDC 658.52.011.56:517.977.5

[Abstract] Optimization of data structures in automated management systems (ASU) is critical to increasing the efficiency of modern and future management systems. The problem of optimizing numerical data structures is especially critical since the volumes of such data and the areas in which they are used are constantly increasing.

This article proposes a method of optimizing the physical structure of a numerical data base. The method is based on compressing the numerical data by transforming a binary decimal Aiken code into a compressed binary decimal code. The processes of compressing the initial forms and reconverting the compressed forms of representing the numerical data are examined. The Aiken code transform-based method is compared with the existing method based on Huffman codes. It is concluded that the Aiken method offers a number of advantages; for example, using Aiken codes makes it possible to synthesize simple rules for forward (compression) and backward (recovery of the initial forms) code transforms. It is further concluded that the proposed method provides a positive technical and economic savings when used in storing large volumes of numerical data and when sending such data along communications channels. Figures 1; tables 5; references 6: 5 Russian, 1 Western.

Local Equilibrium Stability of Interaction Among Entropic Subsystems

[Article by B. L. Shmulyan, doctor of technical sciences, All-Union Scientific Research Institute of Systems Research, USSR Academy of Sciences, Moscow]

UDC 517.977.58

[Abstract] Other works have examined the interaction of subsystems that each maximize the entropy of the system's state, while the allowable state set depends on the states of the other subsystems. This scheme is a generalization of the entropy approach to modeling the interaction of communications systems. This approach has been used most extensively in modeling municipal systems, for which the equilibrium of the interaction in question is interpreted as the "rational" arrangement of facilities with different purposes throughout a city's territory. In such a situation, the conditions of the stability of such equilibrium (even if they are only sufficient) become exceedingly important from a practical standpoint. The present article attempts to clarify the local stability conditions of such equilibria for models of a rather general type. A general method is given for calculating variations in states given a variation in the constraints on the admissible state set. The method entails assuming that the system under examination is in a state of equilibrium, formulating the problem of determining the conditions of its local stability (which is understood to mean the stability of the "game" iteration algorithm used, given only a slight deviation of the initial conditions from equilibrium), linearizing the nonlinear operator used in the problem, and considering the conditions of local equilibrium stability equivalent to the conditions of the convergence of the process defined by that linear operator. Several

examples of the sensitivity of stochastic communications systems are examined; systems with bilateral constraints, unilateral constraints, and constraints on their average values are examined. It is shown that the results obtained during the study reported herein are a generalization of analogous results obtained when using the "normal behavior" approach. References 15: Russian.

YeS Operating System Multiprocessing Supervisor and Its Firmware Support

[Article by R. E. Asratyan, candidate of technical sciences, Control Problems Institute, Moscow; V. Yu. Bayevskiy, Yakor Architectural Design Office, Moscow; V. G. Vasendo, candidate of technical sciences, Computer Scientific Research Institute, Minsk; A. F. Volkov, doctor of technical sciences, and O. V. Gronda, Control Problems Institute, Moscow; and A. V. Zhevnyak, Computer Scientific Research Institute, Minsk]

UDC 681.3.012:681.3.066

[Abstract] It is now generally accepted that firmware implementation of software functions and, above all, operating systems is an important reserve for increasing computer productivity. In the literature, the transfer of software functions to the firmware level has been termed "vertical migration". It has proved to be a promising trend in designing operating systems. Limited firmware implementation of software functions has been termed firmware support. The present article proposes an approach to firmware support of the functions of controlling parallel YeS operating system processes based on what is termed logical factoring of the function and on the principles of designing microprimitive support systems. Because the control units of the YeS operating system are not standardized from the standpoint of size or the links between them, a large number of uncoordinated supervisor subprograms would be required for direct firmware implementation of the functions of the YeS operating system. Having a large number of such components would reduce the operating system's reliability and thus makes direct firmware implementation of the YeS operating system impossible. One approach that surmounts this problem is at least partial control of parallel processes. The essence of this approach lies in the "logical factoring out" of the functions of controlling processes beyond the scope of the YeS operating system. A multiprocessing supervisor is used for this purpose. A multiprocessing supervisor is a set of software and firmware intended to expand and develop the supervisory functions of the YeS operating system with respect to controlling parallel processes and exchange with peripherals. The multiprocessing supervisor can be organized as a set of software and firmware or solely as firmware. The latter is preferable because it permits the most optimal distribution of functions between the software and firmware levels. When this approach is used, the multiprocessor is organized in the form of a set of microprimitives implementing the most useful multiprocessing supervisor functions. The system of microprimitives is best designed on the basis of factoring out the basic functions. The multiprocessing supervisor can be addressed by user programs by using special instructions, and its operation is based on the use of multiple-link lists. The capability of performing extensive parallel processes in conjunction with the optimization capabilities of the multiprocessor page access method has created the prerequisites for using multiprocessing supervisors for the data processing tasks performed within the framework of ASU, data base managers, and automated design systems. Trials of experimental multiprocessing supervisor setups have demonstrated that the new method dramatically reduces the time required to perform such tasks. For example, it was possible to reduce the time required to decode an incoming document stream based on external dictionaries from 1 hour 15 minutes to 3.5 minutes by subdividing the task into 100 parallel processes operating with common dictionaries. Figures 1; references 9: 5 Russian, 4 Western.

Modeling of the Nonstrict Human Discussions in Familiar Situations

[Article by O. G. Kovrizhkin]

UDC 519.8.62-50

[Abstract] A model is constructed of the decision making process involving a person who is familiar with the situation. Such models are expedient when the time or skills of the decision maker are limited. A linguistic variable (e.g., supply) is assigned a value (deficit, sufficient, surplus) or range of values, and these are linked by a transitive preference relation (better-worse, more-less). The distance between values is relative and may never be scaled as an absolute value. The solution is reached by joining the parameters and linguistic parameter values. Factors affecting the decision are also evaluated (the extent of their effect is determined) and considered. The model also handles cases where a clear cut decision cannot be reached and compromises must be made. The extent of the compromise is evaluated and considered. Several examples are given of the workings of the model. The difficulties of determining the confidence of the decision are discussed, as well as the means the model uses to evaluate confidence levels. The model was implemented on an SM-4 computer in FORTRAN. Figures 2; tables 2; references 4.

Finite Automata Over the Direct Product of Free Semigroups, Groups, and Languages

[Article by P. V. Gorshkov and A. B. Stavrovskiy]

UDC 519.713.2

[Abstract] An algebraic theory has been developed to describe the formal languages used in automata. This work examines finite automata for finitely defined semigroups. These automata can assign formal languages to the semigroups. Unlike other works, the transversals and subsets of transversals of regular events are studied, rather than the closures. Here the transversals are isomorphic to a direct product of a free semigroup, and, consequently, of one or two free groups. Thus, one can construct a classification of formal languages and study the properties of the subclasses, based on a study of the structural limitations imposed by finite automata. Some theoretical set correlations are established between classes of languages defined by automata using various methods. Classes of recursive-countable and context-free languages can be described by this model. References 13.

One Application of the Theorem of Implicit Functions in Solving Controllability Problems for Nonstationary Discrete Systems

[Article by Vu Ngok Fat]

UDC 519.6

[Abstract] While sufficient conditions have been obtained for the local controllability of nonlinear systems with no restraints, the controllability of nonlinear systems with restraint in infinite-dimensional space remains relatively unstudied. This article describes one application of the theorem of implicit functions for multivalued representation to obtain new controllability conditions for some classes of nonstationary discrete systems. Conditions are obtained which are not only sufficient, but necessary. A general criterion is obtained for the local controllability of a set for convex processes. Several examples are provided in the article. References 12: 6 Russian, 6 Western.

Data Structures as Sets With Repeated Elements

[Article by Yu. G. Gostev]

UDC 51:681.3.001:007

[Abstract] This article presents a data structure model in which the data structure is represented by a set with repeated elements. The author establishes a parallel between the role of the data structure in programming and the set in mathematics, stating that each is a building block for more complex constructs. Sets with repeated elements are equated with extraordinary sets in set theory. This class of sets has been intractable, but the author shows that finite extraordinary sets can be dealt with if they are taken to be sets with repeated elements. Each element of the set acquires a label or selector that differentiates it from other elements of the same value. Two sets are equal if and only if they are elements of the same sets on the same selector. A set cannot be given by itself; it is always an element of some system of sets (called expanded sets). The article provides examples and illustrations. Figures 16; references 12: 9 Russian, 3 Western.

Implementation of Statistical Expert Systems

[Article by A. G. Vagis]

UDC 681.142/155

[Abstract] In order to eliminate the problem of incomplete or low confidence information in expert systems, ways must be found to automate the extraction of knowledge from statistical data or observations. The application software developed to do this is written in FORTRAN-5 and PL-1 for the YeS computer. It implements methods of multidimensional statistical analysis for continuous and discrete models of spatial and temporal series. The programs carry out a number of functions, including various forms of modeling and simulation. Statistical expert systems are noted for their ability to conduct factoral simulation of dynamic objects. A huge knowledge base is not required because only the information relevant to the current situation needs to be generated. One variant designed for econometric modeling performs the following operations: preliminary data processing, determination of the dependence of temporal series, construction of linear and nonlinear models, and factor and goal simulation. Several other variants use other simulation and modeling techniques. Applications components of the packets and system may be united to develop combined or new statistical expert systems. The system programs organize the computing process under the Yes operating system and the DISOD operating system. The latter supports interactive instruction. Some elements of the application program package can be used on the IBM PC/AT. Once the capabilities of this class of statistical expert systems are enhanced, a new generation of expert systems may be developed. In the future, these expert systems may be implemented as hardware. References 11: 10 Russian, 1 Western.

Possible Approaches to the Solution of Matching Pipelined Computing Processes

[Article by A. V. Palagin and N. B. Fesenko]

UDC 681.325(088.8)

[Abstract] This article develops the mechanism of matching pipelined computing processes on the basis of parallel-sequential arithmetic. Both parallel and sequential computing processes have the problem of synchronizing calculations. The synchronization problem is described, and two methods are given to decrease the misalignment of synchronization signals. The approach depends on the relation between the number of bits processed and the speed of the computer element. Delays in signal propagation due to the speed of the element and the layout of the circuits is considered. The propagation of synchronization signals among and within computer elements is discussed. Matching the speeds of operation in computing cells at the pipeline level suppresses the accumulation of synchronization signal misalignments, thus preventing breakdowns. This makes it possible to determine optimal synchronization rates. Figures 4; references 8: 5 Russian, 3 Western.

Efficiency of Assembly Programming

[Article by V. V. Lipayev]

UDC 681.3.06

[Abstract] There is a need for experimental evaluation of the efficiency of methods used to increase productivity in the development of software. There is also a need for theoretical analysis of how increases in efficiency are achieved. Some increase in efficiency is gained by automating non-original work, that is, using pre-existing software components. The use of reusable software components (assembly programming) becomes more worthwhile as the volume of the software product increases and as the frequency of use of the product increases. When 80 percent of the program product is "recycled" material, efficiency is raised by a factor of 2.5 - 3. The time needed to create the product is reduced by a factor of 1.5 - 2. Next, analysis was done to determine in which stages of software development time and labor were expended. A great deal of time is spent on programming and debugging, but a considerable amount is also used in the initial design process. When prototypes exist, labor may be reduced by a factor of 3 - 4. Worker productivity may be raised by a factor of 3 - 5 by using pre-existing software components. Figures 6; references 5: 4 Russian, 1 Western.

Tools for the Utilization of Software

[Article by N. A. Sidorov and A. N. Sherepa]

UDC 681.3.03

[Abstract] The use of pre-existing software components can reduce the labor and time needed to produce new software products. This article describes the "recycling" technology which has been developed for software, approaches to the creation of software components, and means of revising existing software by creating reusable components. The technology is divided into three subsystems. The first creates software components suitable for repeated use. Programs written in other languages are one source of sofware. The second creates reusable software on the basis of the products of the first subsystem. Analysis, classification, testing, and documentation occur in this subsystem. The third subsystem searches for and selects reusable software which meets the needs of the problem to be solved. The process of recycling software is broken down into phases which depend on how much work is needed to modify the existing software for repetitive use. The work may include in-depth analysis of the software, division into smaller components, and documentation. Tables 2; references 7: 5 Russian, 2 Western.

Facilities for the Expansion of the Input Languages of the Expanding Programming System TEREM and Their Application

[Article by N. M. Mishchenko]

UDC 681.3.068

[Abstract] This article examines pre-processors constructed using the TEREM system. The input languages of these pre-processors are constructed by expanding the basic languages which are the output languages of the pre-processors. The basic languages are implemented independent of the output languages. These pre-processors permit not only input languages but also other basic languages whose syntax may differ. This makes it possible to combine fragments of input text in different languages. The syntax of the language-expander yields a context-free grammar in a metalanguage. Semantics are assigned through a system of semantic objects asssociated with elements of the input grammer. Syntactic tables are constructed to describe the initial state of the pre-processor. Translation from the expansion language may occur on several levels. The metalanguage, basic language, and software to expand the input grammar of the pre-processor are the basic means of expanding their

input languages. Descriptions and examples of these means are provided. Examples are given of preprocessors which have been developed using this system. One implements a language to control asynchronous parallel calculations. Work is being conducted to create a version of TEREM for personal computers. References 4.

A Digital Computing System for Diagnosis and Control of Plasma Parameters in Real Time (Brief Communication)

917G0026F Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, May 90 (manuscript received 16 Feb 89) pp 102-106

[Article by A. N. Golovin, V. P. Grekov, A. V. Matviyenko, and A. I. Nebukin]

UDC 681.32

[Abstract] Control of processes in JT-60 and T-10 thermonuclear installations requires responses on the order of milliseconds. The system described in this article can be used for plasma processes and other highly dynamic processes in real time. The system performs data collection, primary and secondary analysis, and issues control commands. Data is collected by sensors with standardized analog outputs. The system is a modular multiprocessor problem-oriented computing system. Real time processing is achieved by parallel processing among problem-oriented processors and units within the processors themselves. The architecture of the system and the processors is problem-oriented. The response time of this system was found to be a factor of 3 - 4 faster than existing systems. Diagrams of the minimum and maximum configurations of the system are provided. The components of the configurations are described and specifications given. The system and application software and programming tools are briefly outlined. Figures 2.

Infrastructure of an Automated Technological Process Control System

917G0026G Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, May 90 (manuscript received 8 Jun 89) pp 125-128

[Article by L. M. Valyasniy]

UDC 681.012.011.56

[Abstract] Frequently, the resources devoted to the development and introduction of automated technological process control systems do not correspond to the results obtained. A number of factors involved in the introduction affect the efficiency of introduction and operation. These factors are called the infrastructure of automated technological process control systems. These factors include the users' need for a system, the users' ability to put the capabilities of the system to practical use, and the availability of the system. The efficiency of introduction and use of the system is affected not by the level of development of the infrastructure, but by how this level corresponds to the development of the system. A consideration of the real infrastructure makes it possible to more efficiently coordinate facilities or to develop components of the system and its infrastructure. Because the infrastructrue is vital to the system, and must be developed at the same time as the system, it should be funded as a necessary part of development. The infrastructure requires continued support as much as the system itself. References 4.

Integrated Automated System to Control a Shop Implemented Using a Distributed Data Base (Brief Communication)

917G0026H Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, May 90 (manuscript received 25 May 89) pp 128-130

[Article by V. Yu. Meytus]

UDC 681.3.06

[Abstract] The system was designed for an assembly shop. It was implemented with a single distributed data base created with Dbase III Plus and Clipper. All of the subsystems operate in a single information environment, which increases the efficiency of information processing, allows one to switch to a modular construction, and simplifies modification. The system runs on an IBM AT network. The system can be modified without disrupting its operation. Modules can be replaced without taking the system off line. The distributed data base makes use of the Clipper system and the RPTI (Ethernet) network. The system supports the MS DOS 3.30 and NETOS operating systems. MODULA-2, Turbo PASCAL, and other programming languages may be used. The system plans the output of orders with planning of the manufacture of each part. It also determines how long it will take to fill the order. The inflow and outflow of materials is traced, and shop operations are monitored on an operation-by-operation basis. Deviations from plans or schedules are analyzed. Changes can be introduced when plans change or when resources are not being used fully. The system consists of six workstations. The software occupies 450 kilobytes. Tables 1; references 2.

Foreign Broadcast Information Service (FBIS) and Joint Publications Research Service (JPRS) publications contain political, military, economic, environmental, and sociological news, commentary, and other information, as well as scientific and technical data and reports. All information has been obtained from foreign radio and television broadcasts, news agency transmissions, newspapers, books, and periodicals. Items generally are processed from the first or best available sources. It should not be inferred that they have been disseminated only in the medium, in the language, or to the area indicated. Items from foreign language sources are translated; those from English-language sources are transcribed. Except for excluding certain diacritics, FBIS renders personal and place-names in accordance with the romanization systems approved for U.S. Government publications by the U.S. Board of Geographic Names.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by FBIS/JPRS. Processing indicators such as [Text] or [Excerpts] in the first line of each item indicate how the information was processed from the original. Unfamiliar names rendered phonetically are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear from the original source but have been supplied as appropriate to the context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by the source. Passages in boldface or italics are as published.

SUBSCRIPTION/PROCUREMENT INFORMATION

The FBIS DAILY REPORT contains current news and information and is published Monday through Friday in eight volumes: China, East Europe, Soviet Union, East Asia, Near East & South Asia, Sub-Saharan Africa, Latin America, and West Europe. Supplements to the DAILY REPORTs may also be available periodically and will be distributed to regular DAILY REPORT subscribers. JPRS publications, which include approximately 50 regional, worldwide, and topical reports, generally contain less time-sensitive information and are published periodically.

Current DAILY REPORTs and JPRS publications are listed in *Government Reports Announcements* issued semimonthly by the National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, Virginia 22161 and the *Monthly Catalog of U.S. Government Publications* issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

The public may subscribe to either hardcover or microfiche versions of the DAILY REPORTs and JPRS publications through NTIS at the above address or by calling (703) 487-4630. Subscription rates will be provided by NTIS upon request. Subscriptions are available outside the United States from NTIS or appointed foreign dealers. New subscribers should expect a 30-day delay in receipt of the first issue.

U.S. Government offices may obtain subscriptions to the DAILY REPORTs or JPRS publications (hardcover or microfiche) at no charge through their sponsoring organizations. For additional information or assistance, call FBIS, (202) 338-6735,or write to P.O. Box 2604, Washington, D.C. 20013. Department of Defense consumers are required to submit requests through appropriate command validation channels to DIA, RTS-2C, Washington, D.C. 20301. (Telephone: (202) 373-3771, Autovon: 243-3771.)

Back issues or single copies of the DAILY REPORTs and JPRS publications are not available. Both the DAILY REPORTs and the JPRS publications are on file for public reference at the Library of Congress and at many Federal Depository Libraries. Reference copies may also be seen at many public and university libraries throughout the United States.