Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

И.С. Дорошкевич, М.А. Морозова

АНГЛИЙСКИЙ ЯЗЫК. СБОРНИК ТЕКСТОВ ДЛЯ СТУДЕНТОВ ФАКУЛЬТЕТА ИНФОРМАЦИОННЫХ СИСТЕМ И ГЕОТЕХНОЛОГИЙ

Санкт-Петербург

РГГМУ
2018

УДК [811.111:004](076.6+075.8)
ББК 81.2Англ-923
Д69

Рецензенты:
Алешин А.С., к.ф.н., доцент кафедры иностранных языков для ГФ СПбГУТ;
Федорова Н.Ю., к.п.н., доцент кафедры связей с общественностью РГГМУ.

Д69 **Дорошкевич И.С., Морозова М.А.** Английский язык. Сборник текстов для студентов факультета информационных систем и геотехнологий. – СПб.: РГГМУ, 2018. – 60 с.

Сборник текстов предназначен для студентов дневного отделения, обучающихся на факультете информационных систем и геотехнологий РГГМУ. Применение в учебном процессе материалов, представленных в сборнике, ориентировано на развитие умений перевода, а также подробного и краткого пересказа текстов профессиональной направленности. Тематика подобранного текстового материала относится к сфере информационных технологий и геотехнологий, знакомит студентов с основными понятиями данной профессиональной области.

Doroshkevich I., Morozova M. English Texts for the Students of the Faculty of Information Systems and Geotechnology. – St. Petersburg: RSHU Publishers, 2018. – 60 pp.

The textbook is designed for full-time students studying Information Systems and Geotechnology at Russian State Hydrometeorological University. It focuses on developing reading comprehension and translation skills, as well as skills of rendering and detailed representation of IT texts into Russian. The topics of the texts cover the broad areas of IT sciences. Students are introduced to basic concepts of their future professional field. The materials are adopted from the original English-language sources – books, articles, and the Internet.

УДК [811.111:004](076.6+075.8)
ББК 81.2Англ-923

© Дорошкевич И.С., 2018
© Морозова М.А., 2018
© Российский государственный гидрометеорологический университет (РГГМУ), 2018

Предисловие

Сборник текстов предназначен для студентов дневного отделения, обучающихся на факультете информационных систем и геотехнологий РГГМУ.

Данный сборник представляет собой набор научно-технических текстов на английском языке. Применение в учебном процессе материалов, представленных в сборнике, ориентировано на развитие и совершенствование умений перевода, а также подробного и краткого пересказа текстов профессиональной направленности. Кроме того, подобранный текстовый материал знакомит студентов с основными понятиями их будущей профессиональной области, что может служить основной для дальнейшего более глубокого ознакомления с англоязычными источниками, затрагивающими проблематику сферы информационных технологий и геоинформатики.

Тексты с 1 по 16 затрагивают общие вопросы области информационных технологий, тексты 17–25 касаются вопросов геоинформатики и геоинформационных систем, в частности.

Сборник может представлять интерес и для широкого круга читателей, имеющих дело с научной литературой на английском языке в данной профессиональной области.

Тексты рассчитаны на средний уровень владения английским языком. В конце каждого текста дается пояснение терминам, которые могут представлять наибольшую сложность при переводе.

Материалы сборника заимствованы из оригинальных англоязычных источников – научных статей и монографий.

Introduction

The collection of texts in English is designed for full-time students studying Information Systems and Geotechnology at Russian State Hydrometeorological University. The main purpose of the use of these texts in the learning process is to develop students' skills of reading and rendering IT texts into Russian. Students are also introduced to the basic concepts of their future professional field. Students can use them as a basis for the search of other English professional texts to learn more about particular issues of Information Technology and Geographic Information Science.

The texts 1–16 relate to the issues of Information Technology. The issues of Geographic Information Science are concerned in the texts 17–25.

The texts are meant for intermediate level students of English. Each text is provided with a list of terms that may cause difficulties when translating.

The materials presented are built on the authentic books and articles.

The texts can be of interest to a wide range of readers who are interested in IT and Geographic Information Science.

1. Ethical Issues in Information Technology

Computers have had a powerful impact on our world and are destined to shape our future. This observation, now commonplace, is the starting point for any discussion of professionalism and ethics in computing.

Computers, digital data, and telecommunications have changed work, travel, education, business, entertainment, government, and manufacturing. Technological advances can antiquate laws, concepts, and traditions, compelling us to reinterpret and create new laws, concepts, and moral notions. Our attitudes about work and play, our values, and our laws and customs are deeply involved in technological change. When it comes to the social–ethical issues surrounding computers, some have argued that the issues are not unique. All of the ethical issues raised by computer technology can, it is said, be classified and worked out using traditional *moral* concepts, distinctions, and theories.

On the other hand, those who argue for the uniqueness of the issues point to the fundamental ways in which computers have changed so many human activities, such as manufacturing, record keeping, banking, international trade, education, and communication. Taken together, these changes are so radical, it is claimed, that traditional moral concepts, distinctions, and theories, if not abandoned, must be significantly reinterpreted and extended. For example, they must be extended to computer-mediated relationships, computer software, computer art, data mining, virtual systems, and so on.

The uniqueness of the ethical issues surrounding computers can be argued in a variety of ways. Computer technology makes possible a scale of activities not possible before. This includes a larger scale of record keeping of personal information, as well as larger-scale calculations which, in turn, allow us to build and do things not possible before, such as undertaking space travel and operating a global communication system. Among other things, the increased scale means finer-grained personal information collection and more precise data matching and data mining. In addition to scale, computer technology has involved the creation of new kinds of entities for which no rules initially existed: entities such as computer files, computer programs, the Internet, Web browsers, cookies, and so on.

The uniqueness argument can also be made in terms of the power and pervasiveness of computer technology. Computers and information technology seem to be bringing about a magnitude of change comparable to that which took place during the Industrial Revolution, transforming our social, economic, and political institutions; our understanding of what it means to be human; and the distribution of power in the world. Hence, it would seem that the issues are at least special, if not unique.

Most ethical issues arising around computers occur in contexts in which there are already social, ethical, and legal norms. In this respect, the issues are not new or unique. At the same time, the situation may have special features because of the involvement of computers – features that have not yet been addressed by prevailing norms. For example, although property rights and even intellectual property rights had been worked out long before the creation of software, when software first appeared, it raised a new form of property issue. Should the arrangement of icons appearing on the screen of a user interface be own able? Is there anything intrinsically wrong in copying software? Software has features that make the distinction between idea and expression almost incoherent. As well, software has features that make standard intellectual property laws difficult to enforce. Hence, questions about what should be owned when it comes to software and how to evaluate violations of software ownership rights are not new in the sense that they are property rights issues, but they are new in the sense that nothing with the characteristics of software had been addressed before. We have, then, a new species of traditional property rights.

Similarly, although our understanding of rights and responsibilities in the employer–employee relationship has been evolving for centuries, never before have employers had the capacity to monitor their workers electronically, keeping track of every keystroke, and recording and reviewing all work done by an employee (covertly or with prior consent). When we evaluate this new monitoring capability and ask whether employers should use it, we are working on an issue that has never arisen before, although many other issues involving employer–employee rights have. We must address a new species of the tension between employer–employee rights and interests.

The social–ethical issues posed by computer technology are significant in their own right, but they are of special interest here because computer and engineering professionals bear responsibility for this technology. We propose, in parallel with our previous genus–species account, that the professional ethics issues arising for computer scientists and engineers are species of generic issues of professional ethics.

Ethical issue – этические проблемы.
Technological advance – технологическое достижение; технологический прорыв.
Computer scientist – программист; компьютерный специалист.

2. Professional Ethics in Information Technology

Ethics is not just a matter for individuals as individuals. We all occupy a variety of social roles that involve special responsibilities and privileges. Being a professional is often distinguished from merely having an occupation, because a professional makes a different sort of commitment. Being a professional means more than just having a job. The difference is commitment to doing the right thing because you are a member of a group that has taken on responsibility for a domain of activity. The group is accountable to society for this domain, and for this reason, professionals must behave in ways that are worthy of public trust.

The professional group promises to self-regulate and practice its profession in ways that are beneficial to society, that is, to promote safety, health, and welfare. Members of professions implicitly, if not explicitly, agree among themselves to adhere to certain standards because this elevates the level of activity. If all computer scientists and engineers, for example, agreed never to release software that has not met certain testing standards, this would prevent market pressures from driving down the quality of software being produced. The special responsibilities of professionals are grounded in what members of a professional group owe to one another: they owe it to one another to live up to agreed-upon rules and standards.

Moreover, the special responsibilities of professionals can be based on ordinary morality. For example, the engineer's responsibility for safety derives from the ordinary moral edict *do no harm*. Because engineers are in a position to do greater harm than others, engineers have a special responsibility in their work to take greater care.

In the case of computing professionals, responsibilities are not always well articulated because of several factors. Computing is a relatively new field. There is no single unifying professional association that controls membership, specifies standards of practice, and defines what it means to be a member of the profession. Another difficulty in the role of computing professional is the diversity of the field. Computing professionals are employed in a wide variety of contexts, have a wide variety of kinds of expertise, and come from diverse educational backgrounds.

However, there are many several professional codes of conduct, and expectations for professional practice. Professional codes play an important role in articulating a collective sense of both the ideal of the profession and the minimum standards required.

Computer scientists may find themselves in situations in which their responsibility as professionals to protect the public comes into conflict with loyalty to their employer. Such a situation might arise, for example, when the computer professional believes that a piece of software has not been tested enough but her employer wants to deliver the software on time and within the allocated budget (which means immediate release and no more resources being spent on the project). Whether to blow the whistle is one of the most difficult decisions computer engineers and scientists may have to face.

Whistle blowing can be avoided when companies adopt mechanisms that give employees the opportunity to express their concerns without fear of repercussions. The need to blow the whistle can also be diminished when professional societies maintain hotlines that professionals can call for advice on how to get their concerns addressed.

Another important professional ethics issue that often arises is directly tied to the importance of being worthy of client (and, indirectly, public) trust. Professionals can find themselves in situations in which they have (or are likely to have) a conflict of interest. A conflict-of-interest situation is one in which the professional is hired to perform work for a client and the professional has some personal or professional interest that may (or may appear to) interfere with his or her judgment on behalf of the client. For example, suppose a computer professional is hired by a company to evaluate its needs and recommend hardware and software that will best suit the company. The computer professional does precisely what is requested, but fails to mention being a silent partner in a company that manufactures the hardware and software that has been recommended. In other words, the professional has a personal interest — financial benefit — in the company's buying certain equipment. If the company were told this upfront, it might expect the computer professional to favor his own company's equipment; however, if the company finds out about the affiliation later on, it might rightly think that it had been deceived. The professional was hired to evaluate the needs of the company and to determine how best to meet those needs, and in so doing to have the best interests of the company fully in mind. Now, the company suspects that the professional's judgment was biased. The professional had an interest that might have interfered with his judgment on behalf of the company.

There are a number of strategies that professions use to avoid these situations. A code of conduct may, for example, specify that professionals reveal all relevant interests to their clients before they accept a job. Or the code might specify that members never work in a situation where there is even the appearance of a conflict of interest. This brings us to the special character of computer technology and the effects that the work of computer professionals can have on the shape of the world.

To meet testing standard – пройти стандартный, установленный набор испытаний.
Computing professional – специалист по вычислительной технике.
Professional code of conduct – профессиональный кодекс чести.
To blow the whistle (перен.) – бить тревогу.
Silent partner – мало известный сотрудник, активно участвующий в деле.

3. Characteristics of the Internet

The computing and networking environment from which today's Internet evolved had its origins in a cooperative research and education culture. When the ARPANET, the predecessor to the Internet, was first implemented, the main goal was to share resources among groups of researchers in different geographical locations. The groups had compatible goals and worked toward sharing both computing resources and data. Access to the network was restricted to members of the group, so there was no need to be concerned about security at the time.

The culture of sharing among researchers and academicians that was born in and nurtured by the ARPANET lasted well into the 1990s, and there are still vestiges of it today. It included the notion of making information as available as possible, and that tradition still exists in the form of the World Wide Web, where content of all kinds is being provided, almost free of charge, to hundreds of millions of people around the world. It was a strong culture, and it was responsible in large part for why the Internet has grown to such an enormous size today. The early Internet was based on trust; the community of users trusted each other implicitly to work for the common good. As the Internet has broadened its reach and included more and more people with diverse interests and objectives, the trust model has become insufficient.

One of the major challenges for today's Internet is to develop a new trust model that is realistic, easy to implement, and effective in its application. The Internet is different from earlier communications systems in a variety of ways, but several are particularly important. Some differences are best understood when compared to the public switched telephone network (PSTN) that is used worldwide on a daily basis.

The Internet is based upon a model of information transmission called *packet switching*. Every time information is transmitted over the Internet, it is broken up into packets of binary data. The packets are encoded and sent independently over the network, possibly by different routes, and the information is reassembled at the receiving end. This mode of transmission is called packet switching, as opposed to circuit switching.

The Internet is «stupid» in that all it knows how to do is to deliver packets from an origin connected to the network to a destination connected to the network. All services originate at the edge, or the boundary, of the Internet in the computers attached to it. This is in contrast to the PSTN where the intelligence is at the center of the network (at the switch), and the user instruments at the edge have little functionality other than being used for speaking and listening.

The Internet is global. It connects many countries, and information generally flows freely across national borders. This characteristic raises interesting policy concerns not necessarily directly concerned with security. The PSTN is also global, but the methods of accessing phones in different countries are not as opaque as they are with the Internet. The user knows that he is dialing a foreign country, for example, whereas he may access a website without knowing where the servers are located.

The Internet is open. Formally defined as a network of networks, any network that conforms to a family of protocols known as TCP/IP (Transmission Control Protocol / Internet Protocol) can connect successfully with it and become a part of it. The standards defining this family of protocols come from the work of the Internet Engineering Task Force (IETF), an informal technical body based on technical meritocracy and the creation of implementable consensual standards.

The Internet is decentralized. There are no system-wide gatekeepers. If you obey the «rules of the road,» i. e. the TCP/IP standards, you can connect your computer or your network to the Internet.

The Internet is abundant. The barriers to entry are low and the amount of bandwidth, (i.e. how fast you can transmit data through it) depends upon the carrying capacity of the copper wires, fiber links, or satellite channels that are in the path.

The Internet is by and large user-controlled. In many countries, you can choose whether your messages and other transmissions should be encrypted or not. In addition, filtering of messages for whatever reason is under your control, although you may wish to have an external source do it for you, such as instructing your Internet Service Provider (ISP) to filter out spam messages according to rules that you set up.

The Internet is interactive. You can move quickly and easily between access to multiple content providers and sending and receiving electronic mail with many people. While waiting times for on-line services depend upon the size or bandwidth of your connection to the Internet, it is often possible to get response times that support your activities.

The Internet can be vulnerable. Based initially upon a concept of providing services to a relatively homogeneous, cooperative group of people, certain aspects of trust were assumed rather than required to undergo strict verification.

Based on the above characteristics, you should be getting a picture of an Internet that is supportive and permissive of many kinds of activity, rather than one that is restrictive and controlled. This openness strongly reflects the academic and research roots of the Internet, and is responsible for its usefulness for all of us. The Internet was not designed to maximize security, but instead to maximize the fruits of collaborative work; such a degree of openness has provided opportunities for some people to misuse the network in ways that are harmful to others. We need to understand what those misuses are and guard our networks against them.

To work toward – работать над.

Packets witching – коммутация пакетов (метод передачи данных, при котором информация делится на дискретные фрагменты – пакеты; пакеты передаются последовательно, один за другим).

Circuits witching – коммутация каналов (вид телекоммуникационной связи, при которой между двумя узлами сети должно быть установлено соединение, прежде чем начнется обмен информацией).

Receiving end – приемный конец, приемник.

Intelligence	– сообщение.
Gatekeeper	– контроллер шлюза.
Dial-upline	– коммутируемая линия (связи).

4. Information Security Issues

The concepts of computer, network, and data security in cyberspace are similar to issues in the real world; however, the mechanisms are different. For example, in place of keys (physical or electronic), we have passwords to accounts that allow access to information and services. In place of sealed envelopes, we are able to encrypt information so that it is not readable by others who cannot unlock that information with the right key. In comparing the real world with cyberspace, we also observe some of the same violations of trust and confidentiality. In both worlds, it is possible to forge a false return address and even a false signature. In both worlds, it is possible to provide misleading or erroneous information. In both worlds, it is possible to deluge someone with information, either accidentally or deliberately, making it impossible to determine which information is important and relevant. And in both worlds, it is possible to gain access to confidential information and use it in unintended or illegal ways.

There are, however, three important differences. First, violations of security of all types of cyberspace can take place very rapidly. That means that by the time you understand what is happening to your information assets, it may be too late to prevent damage. Of course, not all violations occur quickly; some attacks are observable as they occur and take time to execute. The lesson to draw from this is that preventive measures taken to protect against violations are far superior to detecting a violation while it is happening or after it has been completed.

Consider the following account of the 'Slammer worm', which severely disrupted the Internet early in 2003. All continents and many countries were affected, including many developing countries. «Slammer (sometimes called Sapphire) was the fastest computer worm in history. As it began spreading throughout the Internet, the worm infected more than 90 % of vulnerable hosts within 10 minutes, causing significant disruption to financial, transportation, and government institutions, and precluding any human-based response …» «Slammer began to infect hosts slightly before the 25th January, 2003, by exploiting a buffer-overflow vulnerability in computers on the Internet running Microsoft's SQL server or Microsoft SQL server Desktop Engine (MSDE) 2000. David Litchfield of Next Generation Security Software discovered this underlying indexing service weakness in July 2002. Microsoft released a patch for the vulnerability before the vulnerability was publicly disclosed. Exploiting this vulnerability, the worm infected at least 75,000 hosts, perhaps considerably more, and caused network outages and unforeseen consequences such as cancelled airline flights, interference with elections, and ATM failures».

Second, you do not have to be physically present at a location, or even in the same country, to commit a security violation in cyberspace. This means that someone in Europe, for example, can probe the security of computers in India just as easily as a person located across the street from the target. In cyberspace, the threat can come from anywhere on the network. It may be directed at a known target, the target may have been selected at random, or it may have been chosen because its Internet address was in a range of addresses being probed as a unit. This omnipresent threat should change the way in which we think about security and the profile of our possible adversaries. It is worth noting that the Digital Millennium Copyright Act makes it illegal to design software that decrypts encryption software; national and global copyright regimes on this and other matters related to copyright and data protection are in active development at the present time.

Third, cyberspace provides a powerful but complex environment, in which the responsibility for security is divided among multiple players. If you are a user of computing and network services, there are a number of ways to protect yourself and your personal computer. However, you cannot control your ISP's security policy or its implementation. Nor can you control your client's software, even if you are closely linked with their systems. Thus you need to assume a protective stance over your own assets, while being aware that the connections you are making with the outside world prevent you from eliminating all vulnerabilities on the network.

Slammer worm – Интернет-червь, использующий уязвимости в Microsoft SQL и создающий большие помехи в Интернете.

Buffer-overflow – программная атака переполнением буфера (попытка взлома защиты компьютерной системы с использованием вирусов и сетевых червей, вызывающих переполнение буфера); переполнение буфера.

Indexing services – «поисковики», служба поиска.

ATM (*Asynchronous Transfer Mode* — асинхронный способ передачи данных) — сетевая высокопроизводительная технология коммутации и мультиплексирования, основанная на передаче данных в виде ячеек фиксированного размера (53 байта), из которых 5 байтов используется под заголовок.

5. Failures and Fault Tolerance

Fault tolerance is the ability of a system to continue correct operation after the occurrence of hardware or software failures or operator errors. The intended system application is what determines the system reliability requirement. Since computers are used in a vast variety of applications, reliability requirements differ tremendously.

For very low-cost systems, such as digital watches, calculators, games, or cell phones, there are minimal requirements: the products must work initially and should continue to operate for a reasonable time after purchase. Failures of these systems are easily discovered by the user. Any repair may be uneconomical.

At the opposite extreme are systems in which errors can cause loss of human life. Examples are nuclear power plants and active control systems for civilian aircraft. The reliability requirement for the computer system on an aircraft is specified to be a probability of error less than 10−9 per hour. More typical reliability requirements are those associated with commercial computer installations. For such systems, the emphasis is on designing system features to permit rapid and easy recovery from failures. Major factors influencing this design philosophy are the reduced cost of commercial off-the shelf (COTS) hardware and software components, the increasing cost and difficulty of obtaining skilled maintenance personnel, and applications of computers in banking, on-line reservations, networking, and also in harsh environments such as automobiles, industrial environments with noise sources, nuclear power plants, medical facilities, and space applications. For applications such as banking, on-line reservations, or e-commerce, the economic impact of computer system outages is significant. For applications such as space missions or satellites, computer failures can have a huge economic impact and cause missed opportunities to record valuable data that may be available for only a short period of time. Computer failures in industrial environments, automobiles, and nuclear power plants can cause serious health hazards or loss of human life.

Fault tolerance generally includes detection of a system malfunction followed by identification of the faulty unit or units and recovery of the system from the failure. In some cases, especially applications with very short mission times or real-time applications (e. g., control systems used during aircraft landing), a fault-tolerant system requires correct outputs during the short period of mission time. After the mission is completed, the failed part of the system is identified and the system is repaired. Failures that cause a system to stop or crash are much easier to detect than failures that corrupt the data being processed by the system without any apparent irregularities in the system behavior. Techniques to recover a system from failure include system reboot, reloading the correct system state, and repairing or replacing a faulty module in the system.

Any deviation from the expected behavior is a failure. Failures can happen due to incorrect functions of one or more physical components, incorrect hardware or software design, or incorrect human interaction. Physical failures can be either permanent or temporary. A permanent failure is a failure that is always present. Permanent failures are caused by a component that breaks due to a mechanical rupture or some wear out mechanism, such as metal electromigration, oxide defects, corrosion, time-dependent dielectric breakdown, or hot carriers. Usually, permanent failures are localized. The occurrence of permanent failures can be minimized by careful design, reliability verification, careful manufacture, and screening tests. They cannot be eliminated completely.

A temporary failure is a failure that is not present all the time for all operating conditions. Temporary failures can be either transient or intermittent. Examples of causes of transient failures include externally induced signal perturbation (usually due to electromagnetic interference), power-supply disturbances, and radiation due to alpha particles from packaging material and neutrons from cosmic rays.

An intermittent failure causes a part to produce incorrect outputs under certain operating conditions and occurs when there is a weak component in the system. For example, some circuit parameter may degrade so that the resistance of a wire increases or drive capability decreases. Incorrect signals occur when particular patterns occur at internal leads. Intermittent failures are generally sensitive to the temperature, voltage, and frequency at which the part operates. Often, intermittent failures cause the part to produce incorrect outputs at the rated frequency for a particular operating condition but to produce correct outputs when operated at a lower frequency of operation. Not all intermittent failures may be due to inaccuracies in manufacture. Intermittent failures can be caused by design practices leading to incorrect or marginal designs. This category includes cross-talk failures caused by capacitive coupling between signal lines and failures caused by excessive power-supply voltage drop. The occurrence of intermittent failures is minimized by careful design, reliability verification, and stress testing of chips to eliminate weak parts.

Major causes of software failures include incorrect software design (referred to as design bugs) and incorrect resource utilization such as references to undefined memory locations in data structures, inappropriate allocation of memory resources and incorrect management of data structures, especially those involving dynamic structures such as pointers. Bugs are also common in hardware designs, and can involve millions of dollars when big semiconductor giants are involved.

Failures due to human error involve maintenance personnel and operators and are caused by incorrect inputs through operator–machine interfaces. Incorrect documentation in specification documents and user’s manuals, and complex and confusing interfaces are some potential causes of human errors.

Commercial off-the-shelf software – стандартное программное обеспечение; тиражное программное обеспечение; программное обеспечение, готовое к немедленному использованию.
Fault tolerance – отказоустойчивость, сохранение работоспособности при отказе отдельных элементов.
Cross-talk failure – сбой, вызванный помехами, вмешательствами на линии.

6. Fault Models

The function of a computing system is to produce data or control signals. An error is said to have occurred when incorrect data or control signals are produced. When a failure occurs in a system, the effect may be to cause an error or to make operation of the system impossible. In some cases, the failure may be benign and have no effect on the system operation.

A fault model is the representation of the effect of a failure by means of the change produced in the system signals. The usefulness of a fault model is determined by the following factors: effectiveness in detecting failures, accuracy with which the model represents the effects of failures, tractability of design tools that use the fault model.

Fault models often represent compromises between these frequently conflicting objectives of accuracy and tractability. A variety of models are used. The choice of a model or models depends on the failures expected for the particular technology and the system design; it also depends on design tools available for the various models.

The most common fault model is the single stuck-at fault. Exactly one of the signal lines in a circuit described as a network of elementary gates is assumed to have its value fixed at either a logical 1 or a logical 0, independent of the logical values on the other signal lines. The single stuck-at fault model has gained wide acceptance in connection with manufacturing test. It has been shown that although the single stuck-at fault model is not a very accurate representation of manufacturing defects, test patterns generated using this fault model are very effective in detecting defective chips. Radiation from cosmic rays that cause transient errors on signal lines have effects similar to a transient stuck-at fault which persists for one clock cycle.

Another variation of the stuck-at fault model is the unidirectional fault model. In this model, it is assumed that one or more stuck-at faults may be present, but all the stuck signals have the same logical value. This model is used in connection with storage media whose failures are appropriately represented by such a fault. Special error-detecting codes are used in such situations.

More complex fault models are the multiple stuck-at and bridging fault models. In a bridging fault model, two or more distinct signal lines in a logic network are unintentionally shorted together to create wired logic. A manufacturing defect can convert a combinational circuit into a sequential circuit. This can happen due to either stuck-open faults created by any failure that leaves one or more of the transistors of CMOS logic gates in a non-conducting state, or feedback bridging faults in which two signal lines, one dependent on the other, are shorted so that wired logic occurs.

The previous fault models all involve signals having incorrect logic values. A different class of fault occurs when a signal does not assume an incorrect value, but instead fails to change value soon enough. These faults are called delay faults. Some of the delay fault models are the transition fault model, gate delay fault model, and path delay fault model.

The previously discussed fault models also have the convenient property that their effects are independent of the signals present in the circuit. Not all failure modes are adequately modeled by such faults. For example, pattern sensitive faults are used in the context of random–access memory (RAM) testing, in which the effect of a fault depends on the contents of RAM cells in the vicinity of the RAM cell to be tested. Failures occurring due to cross-talk effects also belong to this category.

The fault models described until now generally involve signal lines or gates in a logic network. Many situations occur in which a less detailed fault model at a higher level of abstraction is the most effective choice. In designing a fault-tolerant system, the most useful fault model may be one in which it is assumed that any single module can fail in an arbitrary fashion. This is called a single module fault. The only restriction placed on the fault is the assumption that at most one module will be bad at any given time.

There have been several attempts to develop fault models for software failures. However, there is no clear consensus about the effectiveness of any of these models — or even whether fault models can be developed for software failures at all.

Fault model – модель неисправности.
Single stuck at-fault model – модель одиночных константных неисправностей.
Bridging fault – неисправность типа перемычки.
Stuck-open fault – константная неисправность типа обрыва или постоянно отключенный затвор.
Feedback bridging faults – неисправность типа короткое замыкание, образующая дополнительную обратную связь в схеме.
Delay fault – неисправность типа неверная величина задержки.
Pattern sensitive faults – кодочувствительная ошибка.

7. Virtual Reality

Virtual reality is a new paradigm in computer–human interaction, in which three-dimensional computer-generated worlds are created which have the effect of containing objects that have their own location in three-dimensional space. The user's perception of this computer-generated world is similar to the perception of the real world.

User perception in virtual reality can be via a variety of senses, including sight, sound, touch, and force. Virtual environments are often, but not necessarily, immersive, providing the effect of surrounding the user with virtual objects.

The user interacts with the virtual environment using several interaction techniques. In order to create the effect of interactive three-dimensional objects, the virtual environment must be processed and presented at a near-real-time rate of 10 frames/s or greater. The three dimensional perception and interaction in the virtual environment, its real-world-like interface.

More precisely, we define virtual reality as the use of computer systems and interfaces to create the effect of an interactive three-dimensional environment, called the virtual environment, which contains objects which have spatial presence. By spatial presence, we mean that objects in the environment effectively have the property of spatial location relative to and independent of the user in three-dimensional space.

One of the guiding principles of virtual reality is to provide the same types of information about the virtual environment as are available for the real world. Thus there are several sensual modalities used to present the virtual environment, including visual, auditory, and haptic (touch and force) displays.

Visual display is one of the most important components of a virtual reality system. There are a number of quality considerations which arise in the selection of a visual display for virtual reality. In addition to the usual display quality considerations from conventional graphics such as color, contrast, brightness, and refresh rate, the following issues arise in virtual reality displays.

Resolution. Defined as the angle subtended by a pixel as viewed by the user.

Pixel spacing. Many screen technologies, such as shadow-masked color cathode-ray tubes and liquid crystal displays, have significant space between the pixels. This space is magnified by wide-field optics and can significantly deteriorate the image.

Field of view. A strong determinant of the immersiveness of a display. The field of view is determined by the physical screen size and the optics.

Optical quality. Because head-coupled displays are very close to the eyes, they require optics to provide a focused image. These optics also determine the field of view and may induce strong optical distortion.

There are a variety of display technologies available which are appropriate to virtual reality. They fall roughly into two classes: head-coupled, which move with the user's head as the user moves about, and stationary, which do not. Head-coupled displays place a screen in front of each of the user's eyes, and are often head-mounted, rigidly attached to the user's head. Head-coupled displays provide a strong sense of immersion, because the user sees the virtual scene no matter which way the user's head is turned. Augmented reality displays often use a half-silvered mirror for display, allowing the virtual scene to be overlaid on the real world.

A significant issue for displays is ergonomics, particularly user comfort. Head-mounted displays can be uncomfortable, leading to rejection by the user community. Stationary displays provide both enhanced comfort and a sharable experience. Which display is chosen will generally depend on the application requirements.

Sound output is an important modality of display in virtual reality. There are roughly two levels of sophistication of sound output: non-spatially-localized (possibly stereo) and three-dimensional spatially localized sound display.

Non-spatially-localized sound display involves using conventional sound rendering techniques, such as sampling and waveform synthesis, to provide nonspatial sound cues such as those found in conventional graphics applications. Nonspatial sounds are typically used to provide feedback as to the occurrence of events such as object collision in the virtual environment and (potentially) data display by varying the characteristics of a continuous sound.

Three-dimensional spatial sound uses various techniques to render sound whose source has a perceived location in three-dimensional space. This sound source location may or may not correspond to a visual or haptic object at the same location in three-dimensional space. The simplest method of providing three-dimensional spatially located sound is to surround the user with an array of speakers. A more sophisticated method provides spatially localized sound via headphones and appropriate signal processing of the sound source.

Haptics refers to the senses of force and touch. Haptic displays use various types of hardware to provide force or touch feedback when the user encounters an object in the virtual environment. Nowadays, haptic displays are highly experimental, with few commercial products available. The primary purpose of haptic displays is to give the user the effect of «touching» objects in the virtual environment. The experience of touching an object in the real world is extremely complex, involving the texture of, temperature of, vibration (if any) of, and forces exerted by the object. Reflecting this complexity, haptic displays fall into two classes: *Surface displays* including texture, vibration, and temperature displays; *Deep displays* including force displays such as pneumatic robotics, compressed air actuators, and «memory metal» devices.

Immersive – иммерсивный, обеспечивающий полный эффект присутствия.
Frame rate – частота смены кадров.

Wide-field optics – физический термин: оптика широкого поля зрения.

Head-coupled display – дисплей, фиксированный на голове.

Folded optics – оптическая система с изломанной оптической осью.

Augmented reality – дополненная реальность, расширенная реальность.

Head-tracking technology – технологии, связанные со слежением за положением головы пользователя.

8. Task analysis

Task analysis is the process of understanding the user's task thoroughly enough to help design a computer system that will effectively support users in doing the task. By task is meant the user's job or work activities, what the user is attempting to accomplish. By analysis is meant a relatively systematic approach to understanding the user's task that goes beyond unaided intuitions or speculations, and attempts to document and describe exactly what the task involves. The design of functionality is a stage of the design of computer systems in which the user-accessible functions of the computer system are chosen and specified.

The basic thesis is that the successful design of functionality requires a task analysis early enough in the system design to enable the developers to create a system that effectively supports the user's task. Thus, the proper goal of the design of functionality is to choose functions that are useful in the user's task, and which, together with a good user interface, result in a system that is usable, that is, easy to learn and easy to use.

The user's task is not just to interact with the computer, but to get a job done. Thus, understanding the user's task involves understanding the user's task domain and the user's larger job goals. Many systems are designed for ordinary people, who presumably lack specialized knowledge, so the designers might believe that they understand the user's task adequately well without any further consideration. This belief is often incorrect; the tasks of even ordinary people are often complex and poorly understood by developers. In contrast, many economically significant systems are intended for expert users, and understanding their tasks is absolutely critical. For example, a system to assist a petroleum geologist must be based on an understanding of the knowledge and goals of the petroleum geologist. To be useful, such a system will require functions that produce information useful to the geologist; to be usable, the system must provide these functions in a way that the frequent and most important activities of the geologist are well supported. Thus, for success, the developer must design not just the user interface, but also the functionality behind the interface.

The purpose of every task analysis training is to provide some background and beginning how-to information about conducting a task analysis and approaching the design of functionality. It should be discussed there why task analysis and the design of functionality are critical stages in software development, and how typical development processes interfere with these stages. Then should be presented some background on methods for human–machine system design that have developed in the field of human factors over the last few decades, including newer methods that attempt to identify the more cognitive components of tasks.

Task analysis – анализ рабочих заданий.
Document – задокументировать, зарегистрировать.
Developer – девелопер, разработчик программного обеспечения.

9. The Critical Role of Task Analysis and Design of Functionality

In many software development organizations, some group, such as a marketing department or government procurement agents, prepares a list of requirements for the system to be developed. Such requirements specify the system functions. The designers and developers then further specify the functions, possibly adding or deleting functions from the list, and then begin to design an implementation for them.

Typically, only at this point or later is the user interface design begun. Ideally, the design of the user interface will use appropriate techniques to arrive at a usable design, but these techniques normally are based on whatever conception of the user and the user's tasks have already been determined, and the interface is designed in terms of the functions that have already been specified.

Thus, the focus of usability methods tends to be on relatively low-level questions (such as menu structure) and on how to conduct usability tests to identify usability problems; the problem is posed as developing a usable interface to functions which have already been chosen.

If the initial requirements and system functions are poorly chosen, the rest of the development will probably fail to produce a usable product. It is a truism in human–computer interaction that if customers need the functionality, they will buy and use even a clumsy, hard-to-use product;

if the functionality is poorly chosen, no amount of effort spent on user interface design will result in a usable system — and it might not even be useful at all. This is by no means a rare occurrence; it is easy to find cases of poorly chosen functionality that undermine whatever usability properties the system otherwise possesses.

The interface is often not the problem. An important case studies on this issue involve failures of functionality design masquerading as usability problems. The most painful is a business organization's database system that was considered too difficult to use. The interface was improved to make the system reasonably easy to use, but then it became clear that nobody in the organization needed the data provided by the system! Apparently, the original system development did not include an analysis of the needs of the organization or the system users. The best way to improve the usability of the system would have been simply to remove it.

Half a loaf is worse than none. The second major version of an otherwise easy-to-use basic word processing application included a multiple-column feature; however, it was not possible to mix the number of columns on a page. Documents having a uniform layout of two or three columns throughout do not exist in the real world; rather, real multi-column documents always mix the number of columns on at least one page. For example, a common pattern is a title page with a single column for the title that spans the page, followed by the body of the document in two-column format. The application could produce such a document only if two separate documents were prepared, printed, and then physically cut and pasted together!

In other words, the multiple-column feature of this second version was essentially useless for preparing real documents. A proper task analysis would have determined the kinds and structures of multiple-column documents that users would be likely to prepare. Using this information during product development would have led either to more useful functionality (like the basic page-layout features in the third major release of the product) or to a decision not to waste resources on a premature implementation of incomplete functionality.

Why doesn't it do that? A first-generation handheld «digital diary» device provided calendar and date book functions equivalent to paper calendar books, but included no clock, no alarm, and no awareness of the current date, although such functions would have been minor additions to the hardware. In addition, there was no facility for scheduling repeating meetings, making such scheduling remarkably tedious. The only short cut was to use a rather clumsy copy–paste function, but it did not work for the meeting time field in the meeting information. A task analysis of typical user's needs would have identified all of these as highly desirable functions. Including them would have made the first generation of these devices much more viable.

Progress is not necessarily monotonic. The second version of a personal digital assistant also had a problem with recurring meetings. In the first version, a single interface dialog was used for specifying recurring meetings, and it was possible to select multiple days per week for the repeating meeting. Thus, the user could easily specify a weekly repeating meeting schedule of the sort common in academics, for example, scheduling a class that meets at the same time every Monday, Wednesday, and Friday for a semester. However, in the second version, which attempted many interface improvements, this facility moved down a couple of menu levels and became both invisible in the interface (unless a certain option were selected) and undocumented in the user manual. If any task analysis was done in connection with either the original or the second interface design, it did not take into account the type of repeating meeting patterns needed in academic settings — a major segment of the user population which includes many valuable early adopter customers.

Usability — юзабилити, научно-прикладная дисциплина, занимающаяся повышением продуктивности, эффективности и удобства пользования каким-либо инструментом.

Failures of functionality design masquerading as — выдавать ошибки, связанные с разработкой функциональных возможностей за…

Half a loaf — часть английской пословицы *Half a loaf is better than no bread* — «Лучше полбуханки, чем совсем ничего» (русский эквивалент — «Лучше синица в руке, чем журавль в небе»).

Personal digital assistant — карманный компьютер, используемый в качестве записной книжки.

10. The Role of Task Analysis in Development

Problems with misdefined functionality arise because first, there is a tendency to assume that the requirements specifications for a piece of software can and should contain all that is necessary to design and implement the software, and second, the common processes for preparing these specifications often fail to include a real task analysis. Usually, the requirements are simply a list of desirable features or functions, chosen haphazardly, and without critical examination of how they will fit together to support the user.

Simply mentioning a user need in the requirements does not mean that the final system will include the right functions to make the system either useful or usable for meeting that need. The result is often a serious waste: pointless implementation effort and unused functionality. Thus, understanding the user's task is the most important step in system and interface design.

The results of task analyses can be used in several different phases in the development process:

Development of requirements. A task analysis should be conducted before developing the system requirements to guide the choice and design of the system functionality; the ultimate usability of the product is actually determined at this stage. The goal of task analysis at this point is to find out what the user needs to accomplish, so that the functionality of the system can be designed so that the user can accomplish the required tasks easily. Although some later revision is likely to be required, these critical choices can be made before the system implementation or user interface is designed.

User interface design and evaluation. Task analysis results are needed during interface design to design and evaluate the user interface effectively. The usability process itself and user testing both require information about the user's tasks. Task analysis results can be used to choose benchmark tasks for user testing that will represent important uses of the system. Usage scenarios valuable during interface design can be chosen that are properly representative of user activities. A task analysis will help to identify the portions of the interface that are most important for the user's tasks. Once an interface is designed and is undergoing evaluation, the original task analysis can be supplemented with an additional analysis of how the task would be done with the proposed interface. This can suggest usability improvements either by modifying the interface or by improving the fit of the functionality to the more specific form of the user's task entailed by a proposed interface. In fact, some task analysis methods are very similar to user testing. The difference is that in user testing, one seeks to identify problems that the users have with an interface while performing selected tasks; in task analysis, one tries to understand how users will perform their tasks given a specific interface. Thus, a task analysis might identify usability problems, but task analysis does not necessarily require user testing.

Follow-up after installation. Task analysis can be conducted on fielded or in-place systems to compare systems or to identify potential problems or improvements. When a fully implemented system is in place, it is possible to conduct a fully detailed task analysis. The results could be used to compare the demands of different systems, identify problems that should be corrected in a new system, or determine properties of the task that should be preserved in a new system.

Follow-up – комплекс действий по сопровождению выполнения задачи.
Field – вводить в эксплуатацию.
Misdefine – неправильно определять.

11. The Changing IT Landscape

The last decade has seen a significant change in the information technology (IT) landscape, resulting in a fundamental shift in the activities and skills required of IT professionals and teams. While still demanding deep technical expertise, the work of IT individuals and teams is becoming more interdisciplinary and interpersonal in nature, requiring a broader range of skills to meet end user needs.

Today’s IT professionals are no longer solely technical specialists; they are also educators, facilitators, and consultants, working as teams in conjunction with end users to solve business needs. Amidst these new demands and roles, IT teams are under increasing pressure to create and deliver products and services that are on time, within budget, and of high quality.

These realities force a reexamination of the factors influencing the work of IT teams from a human dynamics perspective. Today’s setting requires IT professionals and teams to possess a wide range of communication and interpersonal skills, skills that have not always been taught and supported in the technically focused environment of IT.

The advent of ubiquitous computing and the birth of e-terms (e.g., e-mail, e-business) are small indicators of the degree of change that an information-hungry culture has experienced over the past 10 to 15 years. For many end users, IT is now an enabler, rather than an enigma — and has become the province of abroad user base, rather than a select group of technical gurus. This rate of technological advance reflects the extensive study and development of both technical and process dynamics and principles. Despite this progress, system development efforts still continue to face failure at a high rate. The CHAOS 2000 Report estimated that 23 % of application development efforts failed between 1994 and 2000;

an additional 49 % were «challenged» (i. e., completed over budget, past deadline, and with fewer features).

What factors contribute to this situation? Even with our best development methodologies and technological sophistication, the human dimension of systems and software development remains the key element to success. For example, the CHAOS 2000 Report concludes that «user involvement» is the second most important criterion for project success, falling only behind «executive support».

At the most basic level, the output from any development project emerges from the conversations and collaboration among many individuals working together over time. As such, success may ultimately be influenced by the development team's ability to manage the following types of human dynamics.

Team technical diversity. The range of talents required of an IT development team today is both broad and deep. Ten or fifteen years ago, a development team might have consisted primarily of programmers; today, the team is also likely to include functional experts, analysts, architects, writers, network and systems administrators, and perhaps even a facilitator.

In fact, in a recent study across 36 IT-focused organizations, only 12 % of IT professionals surveyed reported their role as being programmer or developer. The result of this trend is that IT teams, which once shared at least a common technical base for building relationships, are now coming together from separate specialties and backgrounds within their own fields. This diversity creates profound opportunities for collaboration, but it can also lead to miscommunication if not managed effectively.

Team collaboration. With decreased development times and increased focus on product integration over development, information technology activities demand that IT team members are able to transfer knowledge internally and communicate effectively. Gartner, a famous IT theorist, notes, «teamwork is key to software life cycle planning» and recommends that managers «establish cross-functional teams to enable consistent and constant communication across multiple groups.»

A study of the Microsoft NT operating system development effort concluded that the *team* was the most vital operating level of that organization. «So rapid are technological developments that the core of the corporation is now the team, the only unit small enough to maintain its intellectual edge».

Interaction with user. The emergence of the Internet and ubiquitous computing has resulted in a new type of end user — one who is more IT savvy than in the past and who faces uncertain and highly volatile requirements as business needs shift and demands grow. This has led to development efforts that are more connected to the user, and more dependent on social and technical interactions as systems are iterated through collaborative development efforts.

End user	конечный пользователь, то есть пользователь, не работающий непосредственно с системой, но применяющий результат ее функционирования.
Human Dynamics	раздел исследований, посвященных сложным системам (complex systems) в области статистической физики, или способ понимания и описания того, как люди обрабатывают информацию.
IT-savvy	подкованный в области информационных технологий, хорошо разбирающийся в них.
Ubiquitous computing	глобальные вычисления.

12. Human Dynamics Research

Research in human factors and human–computer interaction has predominantly focused on interface design, object-oriented techniques, and other issues related to a person's interaction with a computer or system. In the world of computer science and engineering, human factors often avoid the personal — and profoundly messy — areas of human-to-human interaction inherent in the system development process.

This work has been primarily the realm of industrial psychology and organizational development, often difficult to translate to the unique setting of the system development experience. One of the books first connecting the fields of psychology and computer science even acknowledged that «the idea of the programmer as a human being is not going to appeal to some people».

During that period, the field of software psychology brought new focus to the programmer's interaction with computing problems and solutions. Still, with some exceptions, early research in this area primarily involved the programmer's interaction with the computing environment—focusing on cognitive problem solving skills, rather than the ability to work within a team or with others.

Recent research suggests that the focus has shifted to IT management and process, viewing the problems of system development from a project or management science perspective, rather than a personal one.

When IT professionalism is discussed, it usually includes the context of technical or human resource–related issues, such as skills development, recruiting, and retention incentives. Still, while some of these works note the need for effective teamwork and communication among IT workers and users, they rarely describe how these dynamics can be assessed, taught, or developed.

It is to be noted that there is the need for more empirical studies of computer professionals working in the field. This type of research poses unique challenges, because the range of variables and potential interactions impacting the human process carries a complexity that is difficult to manage. As a result, such studies are best approached from an epidemiological viewpoint, rather than as an experimental study based on controllable variables.

Many IT writers describe the computing landscape changing. The history and evolution of the personal computer (PC) is steeped in both folklore and competition. Many trace the origins of today's PCs to the Xerox Alto, but not all agree. For example, the history of IBM's PC, especially the XT and its spinoffs, have little or nothing to do with research done by Xerox on the Alto.

Many believe that the first true personal computer was the Altair, introduced in 1975. The story of the Altair coincides with the emergence of two key people in system development: Bill Gates and Paul Allen, who started Microsoft Corporation and developed programs for the Altair.

In 1977, Steven Jobs and Stephen Wozniak introduced the Apple II personal computer, the flagship for the newly introduced Apple Computer, Inc. For the first time, personal computers became affordable and accessible to the general population. Apple also introduced the intuitive graphical interface, opening the world of computing to millions.

Before the early 1980s, Digital Research, the developers of the CP/M operating system, had yet to build a CP/M version that would work with Intel's 16-bit central processing units. A small hardware vendor named Seattle Computer, with programmer Tim Paterson, offered an operating system much like CP/M, but which used Intel's 8086 processor. Under tremendous time pressure to get it to market, Paterson wrote the code hastily. Subsequently, the 8086-DOS operating system was dubbed QDOS, for *quick and dirty operating system*.

Bill Gates, aware of IBM's problems obtaining an Intel-compatible operating system, acquired a license for and later bought QDOS. He then licensed it to IBM, where it became known as PC-DOS. The PC was a market success, and third-party applications developers began developing a wealth of products that would run on this platform.

Despite the rich, colorful, and very human history of the PC, most computing historians agree that current desktop configurations stabilized around 1990. During this time, the workstation became part of most professional business environments and appeared in many homes. Between 1993 and 1998, the percentage of U.S. households with a computer jumped from 23 % to 42 %; by 2000, the percentage had reached 51 %.

The 1990s' explosion of PC usage would forever change the human dynamics of computing, as IT professionals faced a redefinition of the term *user*. The number and variety of users in the PC world had become far greater than those of the earlier mainframe or minicomputer world. Users became much more sophisticated in the ways of computing, as they started thinking of computers more as interactive tools to meet business needs and less as sophisticated filing cabinets. This led to an increase in user demands from their systems and to heightened expectations of the programmers designing their applications. The ability to code was no longer the sole priority for an IT professional. Other key talents were now required: the ability to design effective user interfaces, an understanding of the client's business setting and requirements, and the ability to communicate effectively with IT-savvy clients.

XT (сокращение от *extended technology*) — элемент аббревиатуры IBMPC/XT, указывающей на семейство персональных компьютеров корпорации IBM.

Spinoff — спин-офф, продукт, часто художественное произведение (комикс, фильм, игра), являющееся производным по отношению к другому уже существующему популярному произведению.

CP/M (*Control Program / Monitor* или *Control Programs for Microcomputers*) — операционная система, написанная в 1973 году, предназначенная для 8-разрядных компьютеров.

DOS (Disk Operating System) — семейство операционных систем для персональных компьютеров, ориентированных на использование дисковых накопителей.

13. Operating System Features in Early Computers

Operating systems have undergone enormous changes over the years. The changes have been driven primarily by hardware facilities and their cost, and secondarily by the applications that users have wanted to run on the computers. The earliest computers were massive, extremely expensive, and difficult to use. Users would sign up for blocks of time during which they were allowed «hands-on» exclusive use of the computer. The user would repeatedly load a program into the computer through a device such as a card reader, watch the results, and then decide what to do next.

A typical session on the IBM 1620, a computer in use around 1960, involved several steps in order to compile and execute a program. First, the user would load the first pass of the Fortran compiler. This operation involved clearing main store by typing a cryptic instruction on the console typewriter; putting the compiler, a 10-inch stack of punched cards, in the card reader; placing the program to be compiled after the compiler in the card reader; and then pressing the «load» button on the reader. The output would be a set of punched cards called «intermediate output.» If there were any compilation errors, alight would flash on the console, and error messages would appear on the console typewriter. If everything had gone well so far, the next step would be to load the second pass of the Fortran compiler just like the firstpass, putting the intermediate output in the card reader as well. If the second pass succeeded, the output was a second set of punched cards called the «executable deck.» The third step was to shuffle the executable deck slightly, load it along with a massive subroutine library (another 10 inches of cards), and observe the program as it ran.

The facilities for observing the results were limited: console lights, output on a typewriter, punched cards, and line-printer output. Frequently, the output was wrong. Debugging often took the form of peeking directly into main store and even patching the executable program using console switches. If there was not enough time to finish, a frustrated user might get a line-printer dump of main store to puzzle over at leisure. If the user finished before the end of the allotted time, the machine might sit idle until the next reserved block of time.

The IBM 1620 was quite small, slow, and expensive by our standards. It came in three models, ranging from 20K to 60K digits of memory (each digit was represented by 4 bits). Memory was built from magnetic cores, which required approximately 10 microseconds for a read or a write. The machine cost hundreds of thousands of dollars and was physically fairly large, covering about 20 square feet. The economics of massive mainframe computers made idle time very expensive. In an effort to avoid such idleness, installation managers instituted several modifications to the open shop mechanism just outlined.

Fortran compiler — оптимизирующий компилятор Фортрана, разрабатываемый компанией Intel для своих процессоров.
Main store — оперативная память.
Stack of punch cards — массив перфокарт.

14. Operating System Development

As early computers were bulky, massive and slowly working, there have been developed some advancements. For instance, an operator was hired to perform the repetitive tasks of loading jobs, starting the computer, and collecting the output. The operator was often much faster than ordinary users at such chores as mounting cards and magnetic tapes, so the setup time between job steps was reduced. If the program failed, the operator could have the computer produce a dump. It was no longer feasible for users to inspect main store or patch programs directly. Instead, users would submit their runs, and the operator would run them as soon as possible. Each user was charged only for the amount of time the job required.

The operator often reduced setup time by batching similar job steps. For example, the operator could run the first pass of the Fortran compiler for several jobs, save all the intermediate output, then load the second pass and run it across all the intermediate output that had been collected.

In addition, the operator could run jobs out of order, perhaps charging more for giving some jobs priority over others. Jobs that were known to require a long time could be delayed until night. The operator could always stop a job that was taking too long.

The operator-driven shop organization prevented users from fiddling with console switches to debug and patch their programs. This stage of operating system development introduced the long-lived tradition of the users' room, which had long tables often overflowing with oversized fanfold paper and a quietly desperate group of users debugging their programs until late at night.

The next stage of development was to automate the mechanical aspects of the operator's job. First, input to jobs was collected offline by a separate computer (sometimes called a «satellite») whose only task was the transfer from cards to tape. Once the tape was full, the operator mounted it on the main computer. Reading jobs from tape is much faster than reading cards, so less time was occupied with input/output.

When the computer finished the jobs on one tape, the operator would mount the next one. Similarly, output was generated onto tape, an activity that is much faster than punching cards. This output tape was converted to line-printer listings offline.

A small resident monitor program, which remained in main store while jobs were executing, reset the machine after each job was completed and loaded the next one. Conventions were established for control cards to separate jobs and specify their requirements. These conventions were the beginnings of command languages. For example, one convention was to place an asterisk in the first column of control cards, to distinguish them from data cards. The resident monitor had several duties, including: Interpret the command language; Perform rudimentary accounting; Provide device-independent input and output by substituting tapes for cards and line printers.

This last duty is an early example of information hiding and abstraction: programmers would direct output to cards or line printers but, in fact, the output would go elsewhere. Programs called subroutines provided by the resident monitor for input/output to both logical devices (cards, printers) and physical devices (actual tape drives). The early operating systems for the IBM 360 series of computer used this style of control.

Large IBM360 installations could cost millions of dollars, so it was important not to let the computer sit idle. Computer architecture advanced throughout the 1960s. Input/output units were designed to run at the same time the computer was computing. They generated an interrupt when they finished reading or writing a record instead of requiring the resident monitor to track their progress. As mentioned, an interrupt causes the computer to save some critical information and to branch to a location specific to the kind of interrupt.

Device-service routines, known as device drivers, were added to the resident monitor to deal with these interrupts. Drums and, later, disks were introduced as a secondary storage medium. Now the computer could be computing one job while reading another onto the drum and printing the results of a third from the drum.

Unlike a tape, a drum allows programs to be stored anywhere, so there was no need for the computer to execute jobs in the same order in which they were entered. A primitive scheduler was added to the resident monitor to sort jobs based on priority and amount of time needed, both specified on control cards.

This mode of running a computer was known as a spooling system, and its resident monitor was the start of modern operating systems. One of the first spooling systems was HASP (the Houston Automatic Spooling Program), an add-on to OS/360 for the IBM 360 computer family. As computers became less expensive, the time cost of switching from one process to another became insignificant. Idle time also became unimportant. Instead, the goal became helping users get their work done efficiently. This goal led to new software developments, enabled by improved hardware.

Patch — патч, программное средство, предназначенная для установки программы или устранения проблем в программном обеспечении.
Resident monitor — резидентный монитор, программа, постоянно находящаяся в оперативной памяти и контролирующая операции, которые производятся с диском и оперативной памятью.
Spooling — спулинг, метод работы с внешними устройствами вывода в многозадачной операционной системе, при котором задачам создается иллюзия одновременного доступа к устройству.

15. Cloud Computing

Cloud computing can be defined as a new style of computing in which dynamically scalable and often virtualized resources are provided as a services over the Internet. Cloud computing has become a significant technology trend, and many experts expect that cloud computing will reshape information technology (IT) processes and the IT marketplace. With the cloud computing technology, users use a variety of devices, including PCs, laptops, smartphones, and PDAs to access programs, storage, and application-development platforms over the Internet, via services offered by cloud computing providers. Advantages of the cloud computing technology include cost savings, high availability, and easy scalability.

Cloud computing can be viewed as a collection of services, which can be presented as a layered cloud computing architecture. The services offered through cloud computing usually include IT services referred as to SaaS (Software-as-a-Service). SaaS allows users to run applications remotely from the cloud.

Infrastructure-as-a-service (IaaS) refers to computing resources as a service. This includes virtualized computers with guaranteed processing power and reserved bandwidth for storage and Internet access. Platform-as-a-Service (PaaS) is similar to IaaS, but also includes operating systems and required services for a particular application. In other words, PaaS is IaaS with a custom software stack for the given application. The data-Storage-as-a-Service (dSaaS) provides storage that the consumer is used including bandwidth requirements for the storage.

There are three categories of cloud services. The first one is the cloud service SaaS, where the entire application is running in the cloud. The client contains a simple browser to access the application. The second one is another type of cloud services, where the application runs on the client; however it accesses useful functions and services provided in the cloud. An example of this type of cloud services on the desktop is Apple's iTunes. The desktop application plays music, while the cloud service is used to purchase a new audio and video content. An enterprise example of this cloud service is Microsoft Exchange Hosted Services. On-premises Exchange Server is using added services from the cloud including spam filtering, archiving, and other functions. Finally, the last one is a cloud platform for creating applications, which is used by developers. The application developers create a new SaaS application using the cloud platform.

There are three types of cloud computing: (a) public cloud, (b) private cloud, and (c) hybrid cloud. In the public cloud (or external cloud) computing resources are dynamically provisioned over the Internet via Web applications or Web services from an off-site third-party provider. Public clouds are run by third parties, and applications from different customers are likely to be mixed together on the cloud's servers, storage systems, and networks. Private cloud (or internal cloud) refers to cloud computing on private networks. Private clouds are built for the exclusive use of one client, providing full control over data, security, and quality of service. Private clouds can be built and managed by a company's own IT organization or by a cloud provider. A hybrid cloud environment combines multiple public and private cloud models. Hybrid clouds introduce the complexity of determining how to distribute applications across both a public and private cloud.

One of the critical issues in implementing cloud computing is taking virtual machines, which contain critical applications and sensitive data, to public and shared cloud environments. Therefore, potential cloud computing users are concerned about cloud computing security.

The security requirements for cloud computing providers begins with the same techniques and tools as for traditional data centers, which includes the application of a strong network security perimeter. However, physical segmentation and hardware based security cannot protect against attacks between virtual machines on the same server. Cloud computing servers use the same operating systems, enterprise and Web applications as localized virtual machines and physical servers. Therefore, an attacker can remotely exploit vulnerabilities in these systems and applications.

Cloud computing — облачные вычисления; концепция, согласно которой программы запускаются и выдают результаты работы в окно стандартного веб-браузера на локальном ПК, при этом все приложения и их данные, необходимые для работы, находятся на удаленном сервере в Интернете.

SaaS (*software as a service* — программное обеспечение как услуга) — бизнес-модель продажи и использования программного обеспечения, при которой поставщик разрабатывает веб-приложение и самостоятельно управляет им, предоставляя заказчику доступ к программному обеспечению через Интернет.

IaaS (Infrastructure-as-a-Service) — облачная модель, которая подразумевает под собой использование виртуальных серверов за арендную плату.

16. Issues in Wearable Computing

Wearable computers range from proof of concept to customer-driven systems design based on a task specification to visionary design predicting the form and functionality of wearable computers of the future. For pervasive computing to reach its potential, the average person should be able to take advantage of the information on or off the job. Even while at work, many people do not have desks or spend a large portion of their time away from a desk. Thus, mobile access is the technology required to make information available at any place and at any time.

Today's computer systems distract a user in many explicit and implicit ways, thereby reducing their effectiveness. The systems can also overwhelm users with data leading to information overload. The challenge for human computer interaction design is to use advances in technology to preserve human attention and to avoid information saturation.

There are four design principles for mobile systems.

1. User interface models — what is the appropriate set of metaphors for providing mobile access to information (i. e., what is the next desktop or spreadsheet)? These metaphors typically take over a decade to develop (i. e., the desktop metaphor started in early 1970sat Xerox Palo Alto Research Center (PARC) and required over a decade before it was widely available to consumers). Extensive experimentation working with end users is required to define and refine these user interface models.

2. Input / output modalities — although several modalities mimicking the input / output capabilities of the human brain have been the subject of computer science research for decades, the accuracy and ease of use(i.e., many current modalities require extensive training periods) are not yet acceptable. Inaccuracies produce user frustrations. In addition,

most of these modalities require extensive computing resources, which will not be available in lightweight, low-power wearable computers. There is room for new, easy-to-use input devices.

3. Matched capability with application requirements — many mobile systems attempt to pack as much capacity and performance in as small a package as possible. However, these capabilities are often unnecessary to complete an application. Enhancements such as full-color graphics not only require substantial resources, but also may compromise ease of use by generating information overload for the user. Interface design and evaluation should focus on the most effective means for information access and resist the temptation to provide extra capabilities simply because they are available.

4. Quick interface evaluation methodology — current approaches to evaluate a human computer interface require elaborate procedures with scores of subjects. Such an evaluation may take months and is not appropriate for reference during interface design. These evaluation techniques should focus on decreasing human errors and frustration.

Over the past 12 years, there have been built wearable computers for over a dozen clients in diverse application areas. We have observed several functionalities that have proven useful across multiple applications. These functionalities form the basis for four user interface models, each with their unique user interface, input/output modality, and capability:

1. Procedures — text and graphics — maintenance and plant operation applications are characterized by a large volume of information that varies slowly over time. For example, even simple aircraft may have over a hundred thousand manual pages. But due to operational changes and upgrades, half of these pages are obsolete every six months for even mature aircraft. Rather than distribute CD-ROMs for each maintenance person and running the risk of a maintenance procedure being performed on obsolete information, maintenance facilities usually maintain a centralized database to which maintenance personnel make inquiries for the relevant manual sections on demand. A typical request consists of approximately ten pages of text and schematic drawings. Changes to the centralized information base can occur on a weekly basis.

2. Master / apprentice help desk — there are times, however, when an individual requires assistance from experienced personnel. Historically, an apprenticeship program, wherein a novice observes and works with an experienced worker, has provided this assistance. A simple example of this is the help desk, wherein an experienced person is contacted for audio and visual assistance in solving a problem.

3. Team maintenance / collaboration — the help desk can service many people in the field simultaneously. Today, with downsizing and productivity improvement goals, teams of people are geographically distributed, yet are expected to pool their knowledge to solve immediate problems. An extension of the help desk is a team of personnel such as field service engineers, police, and firefighters, who are joining together to resolve an emergency situation. Information can be expected to change on a minute-by-minute and sometimes even second-by-second basis.

Wearable computer — носимый компьютер; компьютер, который можно носить на теле.
Proof-of-concept — пробный, экспериментальный.
Pervasive computing — сетевые технологии.

17. Geographic Information Science: Definition and Challenges

GIS, the acronym for Geographic Information Systems, has been around since the 1980s. Although one can impute a number of characteristics from the use of this acronym, at the heart of the term «systems» lies a computer software package for storing, displaying, and analyzing spatial data. Consequently, the use of the term GIS implies an object or tool which one can use for exploring and analyzing data that are recorded for specific locations in geographical space. Conversely, Geographic Information Science or GI Science, or more simply GISc, represents a much broader framework or *modus operandi* for analyzing spatial data. The term GI Science emphasizes more the methodology behind the analysis of spatial data. Indeed, one could define GI Science as: *any aspect of the capture, storage, integration, management, retrieval, display, analysis, and modeling of spatial data.* Synonyms of GI Science include Geocomputation, GeoInformatics, and GeoProcessing.

Under this definition, GI Science is clearly an extremely broad subject and captures any aspect connected with the process of obtaining information from spatial data. At the top level, GI Science is concerned with the collection or capture of spatial data by such methods as satellite remotely sensed images, GPS, surveys of people and/or land, Light Detection And Ranging (LiDAR), aerial photographs, and spatially encoded digital video. The key element here is to capture not only attribute information but also accurate information on the location of each measurement of that attribute. For instance, we might ask people some information on themselves during a survey but we would also like to record some aspect of the location of that individual – this might be the location at which the survey took place, or the person's usual residence or their workplace or some other location. Similarly, if we measure some attribute such as the elevation above sea level and/or the precipitation at a set of points, we also need to know the locations of these points, otherwise the elevation and precipitation measurements are useless.

Once spatial data have been captured, they need to be stored and transmitted. This can create challenges as some spatial data sets can be extremely large. Satellite imagery of the Earth's surface can generate terabytes of data and the move towards global data sets can lead to even larger data sets. Consequently, spatial data sets can be extremely large and finding ways to store, process, and transmit such large data sets efficiently is a major challenge in GI Science.

In many operations, large volumes of spatial data are being collected and a major challenge in GI Science is to turn these data into useful information. Most organizations simply do not have the resources (measured in terms of personnel, knowledge, and/or software) to be able to make full use of all the data they routinely gather. There is a growing need for techniques that allow users to make sense out of their spatial data sets. This mirrors the general transformation of society from one dominated by the industrial revolution with its origins in the eighteenth and nineteenth centuries, to one dominated by the information revolution with its origins in the computer age of the late twentieth century.

Two main sets of techniques exist to turn data into information: visualization and statistical/mathematical modeling. There is a vast array of techniques that have been developed for visualizing spatial data and it remains a very intense and fruitful area of research. Spatial data lend themselves to visualization because the data are geocoded and can therefore be represented easily on maps and map-like objects. Simply mapping spatial data can shed so much more light on what is being studied than if the data are presented in tabular form. The development of algorithms for continuous cartograms, software to create pseudo-3D virtual reality environments, and hardware that allows digital video to be linked to a GPS is providing us with the means towards much more sophisticated visualization of spatial data than traditional 2D maps and there are great advances in this area yet to come. Because spatial data contain attribute and locational information, the data can be shown together as on a simple map of the distribution of an attribute or they can be displayed separately in different windows. The use of multiple windows for displaying spatial data is now commonplace and it can sometimes provide useful information to display a map in one window and a non-spatial display of the data (such as a histogram or scatterplot for example) in another window and to provide a link between the two.

The other set of methods to turn spatial data into information are those involving statistical analysis and mathematical modeling. Specialized statistical techniques have been developed specifically for use with spatial data and a great deal more research is needed in this area. More recently, and probably related to the recent explosion of data availability, «exploratory» statistical techniques have increased in popularity. With these, the emphasis is more on developing hypotheses from the data rather than on testing hypotheses. That is, the data are manipulated in various ways, often resulting in a visualization of the data, so that possible relationships between variables may be revealed or exceptions to general trends can be displayed to highlight an area or areas where relationships appear to be substantially different from those in the remainder of the study region.

Spatial data can be found in most areas of study and include many different types of data, such as:

- Geodetic — coordinate reference systems for locating objects in space;
- Elevation — recording heights of objects above mean sea level;
- Bathymetric — recording the depth of water bodies;
- Orthoimagery — georeferenced images of the earth's surface;
- Hydrography — data on streams, rivers and other water bodies;
- Transportation networks — roads, railways, and canals;
- Communication networks — the transmission of ideas and data across space;
- Cadastral — precise positioning of property boundaries;
- Utilities — the locations of pipes, wires and access points;
- Boundaries — electoral, administrative, school and health districts;
- Medical — the location of incidents of disease and patients with respect to the location of services;
- Crime — the location of police incidents;
- Environmental — habitats, pollution, and the impacts of natural disasters;
- Urban — the location of areas of high priority for social and economic intervention;
- Planning — the spatial impacts of locational decisions;

- Retailing — the location of consumers with respect to the location of services;
- Biogeography — the location of one species with respect to the location of one or more others.

Attribute — атрибут, реквизит. В геоинформатике «атрибут» — это свойство, качественный или количественный признак, характеризующий пространственный объект (но не связанный с его местоуказанием) и ассоциированный с его уникальным номером (идентификатором). Множество атрибутов пространственного объекта образует атрибутивные данные.
Orthoimagery — ортографическая визуализация/проекция.

18. Geographic Information Systems

A geographic information system (GIS) is a computerized system that facilitates the phases of data entry, data analysis and data presentation. The domain of GIS concerns georeferenced data, plus integration and analysis procedures, that function to transform the raw data into meaningful information that can support management decisions. In any environmental GIS, after defining the nature of the problem, the initial activity will be to measure aspects of the variable or natural process including both spatial and temporal perspectives. Variables or processes will have three types of properties that need recording: (1) features; (2) attributes; and (3) relationships. GIS will have the ability to store and access digital details of these measurements from a computer database. Then, measurements will be linked to features on a digital map. Aspects of the features will be able to be digitally mapped as points, lines, and polygons (vector) as well as pixels and voxels (raster). The analysis of collected measurements as well as the application of numerical manipulations or modelling algorithms may produce additional data. The combined analysis of several datasets in a GIS environment provides meaningful information for natural processes and it is the core of the GIS technique. The depiction of the analysed data in some type of display (maps, graphs, lists, reports or summary statistics) provides for the communication media of GIS results or output.

Geographic Information Systems (GIS) were ‘born’ on land; they are around 35 to 40 years old, but only about 15 years ago did they ‘migrate’ to the sea. There are important differences between marine and terrestrial GIS applications. The static, terrestrial-based GIS developments consist of certain functions such as overlaying, buffering, reclassification and Boolean operators. Terrestrial objects are very suitable for such operations and output results with a high accuracy. Marine problems have the characteristics of the fuzzy boundary, dynamics, and a full three dimensions. It is not completely suitable for the land-based GIS to apply fully in the marine environment. In the marine context, GIS development enters into a highly dynamic environment where almost everything moves or changes. Marine GIS is called upon to describe the intimate relations among the wind and sea currents that trigger certain oceanographic processes and explain the impact of these processes to the behaviour of marine organisms, taking species biology and ecology into consideration, as well.

The researchers Wright and Bartlett identified the important contribution of GIS in coastal and oceanographic research by opening new ways of georeferenced data processing. They underlined the migration of the early ocean GIS applications that were simple data collection and display tools to complex integrated modeling and visualization tools. They also pointed out the primitive stage at which volumetric GIS analysis and 3D GIS visualization is today, underlining that marine GIS must first adapt to the characteristics of the marine world and marine data and then output results that describe the dynamic relations among the various components of the marine environment.

Marine GIS, as a general term, includes a wide area of applications. Depending on the nature of questions that a marine GIS is called upon to answer and on the spatial extent they cover, applications in this field may be categorized as coastal, oceanographic, and fisheries GIS, with a good deal of overlap among all the three main kinds of marine GIS. A coastal fisheries GIS dealing with how oceanographic processes, such as upwelling, affect fish populations and production is a common example of the overlapping of marine disciplines in marine GIS applications. In such applications, GIS developers are called to cooperate with scientists from a variety of disciplines in order to design an application in such a way that specific spatiotemporal questions could be answered. Thus, marine GIS development is a multidisciplinary procedure and scientists from many disciplines are invited to participate. Marine and fisheries biologists as well as physical and biological oceanographers participate in the development process and check GIS input, analysis, and output for accuracy.

The generation of information-based management proposals is one of the main goals of a marine GIS development, which adds policy makers to the GIS development process. The ability of GIS to map integrations among a variety of datasets is unique for the identification of conflicts between current management policies and marine objects dynamics. GIS contribute to the enhancement of natural resource monitoring and test the efficiency of currently applied management policies by presenting integrated products that describe the relations among biotic and abiotic resources and their current management schemes. The mapping of possible conflicts will lead to new information-based management proposals and schemes, which, depending on the case, will prove to be effective to the preservation, conservation, and sustainable management of marine resource dynamics.

In respect of monitoring ocean and fisheries dynamics and generating new information-based management schemes, GIS technology is closely related to several other technologies, such as GPS, RS, modelling, image processing, spatial statistics as well as the Internet. Cross-disciplinary integration is essential in gripping with the complexity of contemporary environmental problems (global climate change, human impacts on environment, mitigation of environmental hazards, etc.) using the substantial powers of computation for data analysis, process simulation, and decision aid.

Today, there is an increasing number of GIS consortia and organisations that promote communication among scientists from different disciplines towards establishing cross-disciplinary integration in GIS and facilitating the resolution of GIS-related issues worldwide.

Georeferenced data — геоданные; данные с привязкой к местности; пространственно-временные данные.
Voxel — воксел(ь). Элемент объёмного изображения, содержащий значение элемента растра в трёхмерном пространстве, трехмерный пиксель.
Boolean operator — булев оператор, логический оператор.
Decision aid — поддержка принятия решений.

19. Spatial databases

Spatial databases are a specific type of database. They store representations of geographic phenomena in the real world to be used in a GIS. They are special in the sense that they use other techniques than tables to store these representations. This is because it is not easy to represent geographic phenomena using tables.

A spatial database is not the same thing as a GIS, though they have a number of common characteristics. A spatial database as any other database focuses on the following functions: concurrency, storage, integrity, and querying, especially, but not only, spatial data. A GIS, on the other hand, focuses on operating on spatial data with what we might call a 'deeper understanding' of geographic space. It knows about spatial reference systems, and functionality like distance and area computations, spatial interpolations, digital elevation models et cetera. Obviously, a GIS must also store its data, and for this it provided relatively rudimentary facilities. More and more, we see GIS applications that use the GIS for the spatial analysis, and a separate spatial database for the data storage.

The assumption for the design of a spatial database schema is that the relevant spatial phenomena exist in a two- or three-dimensional Euclidean space. Euclidean space can be informally defined as a model of space in which locations are represented as coordinates — (x, y) in 2D; (x, y, z) in 3D — and notions like distance and direction have been defined, with the usual formulas. In 2D, we also talk about the Euclidean plane.

The phenomena that we want to store representations for in a spatial database may have point, line, area or image characteristics. Different storage techniques exist for each of them. An important choice in the design of a spatial database application is whether some geographic phenomenon is better represented as a point, as a line or as an area. Currently, the support for image data exists but is not impressive. Some GIS applications may even be more demanding and require point representations in certain cases, and area representation in other cases. Cities on a map may have to be represented as points or as areas, depending on the scale of the map.

To support this, the database must store representations of geographic phenomena (spatial features) in a scaleless and seamless manner. Scaleless means that all coordinates are world coordinates given in units that are normally used to reference features in the real world (using a spatial reference system). From such values, calculations can be easily performed and any (useful) scale can be chosen for visualization. A seamless database does not show map sheet boundaries or other partitions of the geographic space other than imposed by the spatial features themselves. This may seem a trivial remark, but early GIS applications had map production as their prime purpose, and considered map sheet boundaries as important spatial features.

All geographic phenomena have various relationships among each other and possess spatial (geometric), thematic and temporal attributes (they exist in space and time). Phenomena are classified into thematic data layers depending on the purpose of the database. This is usually described by a qualification of the database as, for example, a cadastral, topographic, land use, or soil database. A spatial database not only serves to store the data and manipulate it, as it should also allow the users to carry out simple forms of spatial analysis.

Spatial analysis involves questions about the data that relate topological and other relationships. Such questions may involve neighbourhood, distance, direction, incidence, disjointness and a few more characteristics that may exist among geographic phenomena. In the El Ni˜no case, for example, we may want to find out where is epicentre of warm water or where is the steepest gradient in water temperature.

A database, like a GIS, is a software package capable of storing and manipulating data. This begs the question when to use which, or possibly when to use both. Historically, these systems have different strengths, and the distinction remains until this day.

Databases are good at storing large quantities of data, they can deal with multiple users at the same time, they support data integrity and system crash recovery, and they have a high-level, easy to use data manipulation language.

GISs are not very good at any of this.

GIS, however, is tailored to operate on spatial data, and allows all sorts of analysis that are inherently geographic in nature. This is probably GIS's main stronghold: combining in various ways the representations of geographic phenomena.

GIS packages, moreover, nowadays have wonderful, highly flexible tools for map production, of the paper and the digital type. GIS have an embedded 'understanding' of geographic space. Databases mostly lack this type of understanding.

The two, however, are growing towards each other. All good GIS packages allow to store the base data in a database, and to extract it from there when needed for GIS operation. This can be achieved with some simple settings and/or program statements inside the GIS. Databases, likewise, have moved towards GIS and many of them nowadays allow to store spatial data also in different ways. Previously, they in principle were capable of storing such data, but the techniques were fairly inefficient.

In summary, one might conclude that small research projects can probably be carried out without the use of a real database. GIS have rudimentary database facilities on board; the user should be aware they are really rudimentary.

Mid-sized projects use a database/GIS tandem for data storage and manipulation. Larger projects, long-term projects and institutional projects organize their spatial data processing around a spatial database, not around a GIS. They use the GIS mostly for spatial analysis and output presentation.

Spatial feature — пространственный объект.

20. Applying GIS to the Coast

Since the late 1980s there has been an upsurge in the application of proprietary GIS packages to the coast. Initially this tended to be based on software running on mainframe computers but, increasingly, workstations and desktop / personal machines are being used, as the performance of these increases and a greater range and amount of suitable software appears on the market. Many GIS vendors and products are now involved. At the same time, the application to typical coastal problems of most current proprietary GIS is still limited and the drive to extend GIS functionality further into the marine and shore environments is a concern of many, if not most, major system developers. For the end-user, the application of proprietary software has both costs and benefits. The advantages include:

- freedom from the need to write, develop, maintain in-house software;
- improved access to data, since a growing amount of data is being collected and made available using standard interchange formats, many of which are based on the data formats of particular vendors;
- easier vertical (within a particular project) and horizontal (between projects) integration of data sets and applications;
- easier implementation of data standards for data quality and exchange;
- the user-base and pool of expertise of proprietary software is invariably going to be more extensive than that for systems developed in-house. This in turn can lead to reduced need for system-specific training, improved learning curves for new personnel, and greater access to assistance and advice in case of difficulty.

Against this, the disadvantages of applying proprietary GIS packages to the coast also have to be considered. In particular, most commercial GIS were developed for land-based applications, and are built around cartographic metaphors, data models and fundamental paradigms optimised for conditions found in terrestrial (including on-shore social and economic) environments. These paradigms and models frequently are poorly suited to coastal data, where boundaries between key variables are less easy to define; where a much greater range of spatial scales and resolutions have to be considered; where there is a greater need to work in three spatial dimensions; and where the temporal dimension is also fundamental to many analyses. Notwithstanding the many advances in technology, coastal zone GIS remains comparatively immature and is situated firmly at the cutting edge of geoinformation science and technology.

One of the most important lessons to be learned from collective experience in the field of coastal GIS, both published and unpublished, is the importance of rigorous data modelling before work starts on implementing a GIS database. Any GIS database is a model of reality, and it is important to establish a thorough and rigorous conceptual model, in order to successfully identify and represent all those features of the real world considered relevant to the study. Unfortunately, where the coast is the focus of study, this modelling is made particularly difficult, partly due to the multiplicity of professions, stake holders and other groups with interests at the shore; and partly due to the inherently fuzzy and indeterminate nature of so many coastal entities and phenomena. There is a multiplicity of different sectors, public and private, with interests at the coast. It is essential to establish the user-base for whom the coastal zone GIS is intended, and the view of the coast and its component elements that is relevant to this constituency, since this operational context will have a direct bearing on the more technologically-oriented questions of hardware and software selection, and the overall architecture of the system. It will also define the information products expected from the system, and hence the types of data, and the processes performed on these, that are required in order to produce the desired output. One useful approach to deriving this conceptual model, where time and resources permit, is for the database designer to pay one or more field visits to the coast in the company of those scientists, administrators or other professionals for whom the system is being developed. During this field visit, the prospective end-users are invited to describe, in purely verbal terms, using the language and terminology of their discipline, the coastal zone that they are seeing. By examining the elements, attributes and relationships between these that they identify, the data modeller can thereby derive a first-order impression of what is important and what is of lesser concern to these users within the coastal system. While the results of this modelling will be *ad hoc* and unstructured, they do provide a useful basis for subsequent more formal entity-relationship modeling based on this provisional schema.

Proprietary GIS packages — коммерческие ГИС-пакеты, коммерческие программные средства ГИС.
Cutting edge — зд. передовое направление; передний край, авангард.
In-house software — программное обеспечение для внутреннего пользования.
User-base — база пользователей.
To have a direct bearing on — иметь непосредственное отношение к.
Ad hoc — спонтанный, случайный, произвольный.

21. Software Issues and Packages for Marine GIS Development

There is a great number of GIS packages on the market today, which include commercial products from well-established software enterprises as well as some free products available from new companies willing to attract users and conquer a share in the world's GIS software market. Several products are already widely used for the development of marine GIS applications (e. g. ARC/INFO, ArcView, MapInfo, Imagine, GRASS, MGE, IDRISI, ILWIS) and through the years, they established a main core of marine GIS developments that are based on these products. Of course, this does not limit the introduction of new GIS software on the market, since the GI sector is continually growing with fast GIS solutions needed and with software learning curves and friendliness of use playing important roles to the software selection process. Specifically for the development of marine GIS software packages, the market is widely open because as of today, there is not a GIS software, which offers in one package a 3D database, 3D data integration and 3D visualisation capabilities. On the other hand, the well-established software enterprises are trying to confront with GIS interoperability issues aiming to produce products that satisfy the exchange of applications that are based on different software solutions.

Throughout the GIS software selection process, several issues become important on all application levels, including single user desktop, departmental, local authority and governmental levels. Software solutions should be examined for their learning curve, ease of use, hardware platforms, network capabilities, the amount and formats of data that they can process and their connection abilities with other GIS external packages (e. g. statistical or visualization software). For example, today most high-level GIS networks use powerful GIS packages as main data servers and a series of compatible satellite GIS software as peripheral tools for interaction with the main data servers. A practice of 'what others already use' is common to the selection process, mainly due to compatibility issues in similar disciplines.

Designs of new GIS installations for high-level networks should look in the future. The selected GIS packages should be capable to interchange data and output with other widely used packages while confronting interoperability specifications on database design, metadata documentation, data analysis, and Internet dissemination. In parallel, established and new GIS software enterprises should focus on the development of slightly different versions of their products based on the target discipline. This becomes essential in disciplines like Oceanography and Fisheries, which need extensive 3D GIS operations for the modelling of real-world dynamics.

Following, a small reference list of commercial and freely available 'lighter' GIS packages are presented, which all are worth trying for their specific level of development and purpose of use. AGIS is a simple mapping and GIS software package for Windows based applications. Data are inserted in the form of ASCII text files and maps can be animated and projected to several map projections. A built in scripting language allows the creation of map animations and link to other applications such as database queries. CARIS GIS provides a marine GIS for retrieval of geographic and textual information from various databases, based on selection criteria, such as source, scale, area, depth, etc. ERDAS Imagine is an image processing and GIS package, which includes many raster and vector analytical operations as well as an extensive modeling module. ESRI products include a variety of widely used GIS packages with build in extensive analytical tools, conversion among different data formats based on specific analysis needs and customisation languages that allow integration of both raster and vector datasets. GRASS GIS (Geographic Resources Analysis Support System) is an open source GIS with raster, topological vector, image processing, and graphics production functionality that operates on various platforms. Data are inserted in the form of 2D and 3D raster files including a variety of formats. GRASS is mainly used as spatial analysis, map generation, and data visualization tool. MapInfo products include a variety of widely used GIS module applications with integration capabilities of both raster and vector datasets. PCRaster is a raster-based GIS where data cells can store more than one attribute. GIS operations can be performed between attributes on one cell location or between cells. SADA (Spatial Analysis and Decision Assistance) is a freeware that incorporates tools from environmental assessment fields into an effective problem-solving environment. The software includes modules for visualization, geospatial and statistical analysis, cost/benefit analysis, sampling design, and decision analysis. SPRING is a GIS and RS image processing system. It features an object-oriented data model under which both raster and vector data representations can be integrated. ProGIS among other products, offers WinGIS, a mapping software tool with capabilities such as layer grouping and object selection that also accepts several common data formats. IDRISI offers exceptional raster analytical functionality for both GIS and RS needs and includes tools for database query, spatial modelling, image enhancement and classification. ILWIS is also an RS and GIS software, which integrates image, vector and thematic data in one powerful package that delivers a wide range of capabilities, such as data import and export, digitising, editing, analysis and display of data and production of quality maps. Last but not least, TNT products include a variety of applications for both GIS and RS with many capabilities, such as vector and raster operations, attribute and database operations including a customisation language (SML, Spatial Manipulation Language). Most of the enterprises that develop these products offer fully functional trial versions or reduced functionality freeware products, often with extensive documentation.

Software solutions — программные решения.

Learning curve — кривая обучения, кривая роста производительности. Отражает ситуацию, при которой с каждым повтором операции либо увеличением выпущенной продукции значение среднего затраченного времени на единицу продукции сокращается на определенный процент.

A built in scripting language — встроенный скриптовый язык программирования.

Customisation language — специализированный язык программирования. Язык программирования, разработанный или адаптированный для выполнения специфических для какой-либо сферы деятельности задач.

RS (remote sensing / technology) — дистанционное обследование, дистанционное зондирование.

22. What is ECDIS?

Electronic Chart Display and Information Systems (ECDIS) is an advanced navigation information system for use in ships. It has been developed to lighten considerably the navigation workload, freeing the mariner for other important navigation-related tasks such as maintaining a safe lookout and for collision avoidance. It is a real-time decision aid, which provides the navigator with accurate and reliable information about a ship's position and its intended movements in relation to charted navigational features.

ECDIS combines satellite position fixing, ship's sensors and other data with a sophisticated electronic database containing chart information. The electronic chart database is known as an electronic navigational chart or ENC. An ENC is much more than a computer copy of a paper chart, though it can look very similar when displayed on ECDIS equipment. ENCs are sophisticated and strictly controlled vector navigational chart data bases containing high levels of textual, spatial and graphical data representing not only the material already shown on a paper chart, but also additional data and information drawn from other publications and from source survey data. ENCs are only produced by or on the authority of Government-authorised hydrographic offices or relevant government-authorised organisations.

ENCs are divided up into manageable geographic areas called «cells». Each cell is rectangular and defined by two geographic parallels (latitude) and two meridians (longitude). Individual cell boundaries are chosen by compiling authorities to best meet their requirements. Cells with the same navigational purpose cannot contain overlapping data, however cells of different purpose may overlap. Cells may not be more than 5Mb in size.

The fact that chart information is held in a data base rather than as a fixed image means that it can be analysed, manipulated and compared with other available information to provide a powerful decision making tool on the bridge of a ship. ECDIS continually analyses the ENC database and compares it with a ship's position and its manoeuvring characteristics to give timely warning of approaching dangers or notable events in the navigational plan. For example, ECDIS can display and respond to a nominated safety depth contour based on a ship's draft and give advanced and automatic warning of the approach to potentially dangerous situations.

ECDIS provides the mariner with a continuous, unified presentation of relevant navigation information, allowing him or her to decide what to do rather than spending valuable time determining whether action is required. ECDIS allows the level of chart detail and various day / night colour schemes to be selected (within strictly defined limits) by the operator.

ECDIS provides many other navigation and safety features including continuous voyage data recording and playback. The mariner can also supplement ENC data with additional information such as personal notes and warnings. These can be input manually and used to raise additional alarms and warnings.

The ship's radar signal can also be incorporated into an ECDIS and the radar image or contact data displayed on screen as an overlay. This helps provide a comprehensive and fully integrated appreciation of the navigational situation and brings together the charted information (which is essentially a record of expected conditions) with a record of the current circumstances as seen by radar.

ECDIS and particularly the requirement to create and publish ENCs is leading to major changes in the way hydrographic organisations will handle their data holdings in future. More and more data is being captured and processed digitally and exchange formats are being standardised. Within some hydrographic organisations' sophisticated digital data assessment, processing and production facilities are also under development. These rely on GIS technology and offer the capability not only to more easily identify what hydrographic data is available, but also to provide tailored and sorted data sets in appropriate formats according to a user's requirements.

Electronic Chart Display and Information Systems (ECDIS) — электронно-картографическая навигационно-информационная система (ЭКНИС).

23. The Two Principal Data Models

Practitioners of GIS have for years grouped geographic data into two classes of models: raster and vector data models. The choice of raster or vector data models for a GIS has usually been presented as an either-or question and sometimes as a debate over the merits of the two data models. More recently, however, practitioners have realized that their needs may incorporate multiple models for spatial data, and it has become unnecessary to view the process as an either-or choice. Rather, you need to consider your needs for modeling your portion of the real world. Many, but not all, GIS software systems can move data back and forth between these two data models, and you may discover that certain tasks are easier to perform in one or the other of the data models. But many users find that one or the other of the data models meets their needs so that one is best for their applications. Organizations that require high-quality mapped output from their GIS will find that only the vector data model allows that. Users who will be doing lots of overlaying of different geographic features will find out that the raster data model is more efficient for that type of task.

When you are trying to develop a model of a real-world object, it is vital to keep several questions in mind. The main questions are: 1) *What is the purpose of this model?* 2) *Which data model, raster or vector, better suits my purpose, or is a combination appropriate?*

Most GISs are able to handle both raster/image data and vector information; so the correct question is what is the proper data model for particular features rather than what is the proper data model for the entire GIS.

If the purpose of the model is solely to be a digital representation of the feature for mapping, the design issues are rather simple. You will need to decide what kind of digital feature you will use to represent the real-world feature and how to symbolize it on the map. In the case of a land parcel, if the sole purpose is mapping of the land parcel for visual representation, the parcel will consist of a set of lines with symbolization and a point inside these lines where some unique identification number is attached. The point symbol may not be visible at all because its only purpose is to serve as an anchor for some text. The digital line, point, and text features may exist separately in your GIS because the map reader does the linking between them visually. The presence of a number inside a set of bounding lines informs the reader that that number identifies the polygon represented by the bounding lines. This is a common way to visually represent land parcels using CAD tools.

If the model is to serve multiple purposes, design concerns become more complex. Using the land parcel example, let us say that in addition to simple mapping you want to be able to link the assessor's information so that this information can be included in a report that provides a quick printout of the important information along with a map of the parcel and adjoining parcels. In this case, the land parcel can no long exist only as a set of lines but must be modeled as a polygon or area in your GIS. In addition, the unique identifier (parcel identification number, or PIN) must be in the GIS database and directly linked to the parcel so that it can be connected with the assessor's information. The system, not having the ability to see the PIN inside the lines bounding the parcel, must know the parcel's PIN and relate it to that parcel and only that parcel.

The representation or model of a real-world feature may be different depending on the purpose of the model. Although a land parcel in the real world is a polygon, there may be some value in representing that polygon as a point in some databases. The fire department may need to have the buildings on land parcel represented as a digital image so that they can see the buildings and where they are relative to each other when planning or evaluating a fire response. But when they want to know what hazardous materials are stored in that building and where in the building they are stored, the building may need only to be a record in a database linked to an address or a point. A business responsible for distributing goods over a wide area may be able to find good locations for a warehouse by modeling their regional demand and supply facilities as points along a network of major highways, but when they have to actually move goods from a particular warehouse to a set of locations, they will need a larger-scale model of the transportation network. So the same real-world feature may exist in your GIS database several times for the different purposes. The need to display or analyze geographic data at very different spatial scales also results in this apparent duplication.

Either-or choice — выбор между двумя вариантами.
Mapped output — зд. картографические данные.
Feature — объект, особенность.

24. Geospatial Data Integration

The problem of integrating geospatial data is ubiquitous since there is so much geospatial data available and such a variety of geospatial formats. The commercial world has numerous products that allow one to combine data that is represented in the myriad of geospatial formats and perform the conversion between products in these various formats. In an effort to improve the sharing and interoperability of geospatial information, the Open GIS Consortium has created the Geographic Markup Language (GML) to support the sharing of geographic information. GML provides an agreed-upon representation for publishing and using the various types of spatial data (for example, maps, vectors, etc.). As the interest in and use of geospatial data continues to grow, these efforts will be critical in exploiting the geospatial data that is available.

However, even in a world where geospatial products can be readily converted between different formats and geospatial information is published using agreed standards for interoperability, the problem is not fully solved. First, there may be sources that are not represented in any geospatial data format. For example, there are libraries of maps on the web that contain maps with only a textual description of the map and no metadata about the location or scale of the map. Second, there are many sources of online information that can be placed in a geospatial context, but the information is only available on a website as HTML pages, not in any standard geospatial format. Finally, even sources that may be available in one of the many standard formats may be difficult to integrate in a meaningful way due to differences in the resolution of the products, differences in the algorithms used to orthorectify the products, or just the lack of metadata on the products.

Let's consider recent work on the extraction and integration of geospatial and geospatial-related data.

(1) Extracting Data from Online Sources. Beyond the traditional types of geospatial data sources, including satellite imagery, maps, vector data, elevation data, and gazetteers, there are many other sources of information available on the web that can be placed in a geospatial context. This includes sources such as property tax sites, telephone books, train, and bus schedules. The amount of such information is large and continues to grow at a rapid rate. The challenge is how to make effective use of all of this information and how to place it in a geospatial context. The first step is to turn the online sources that were intended for browsing by people into sources that can be effectively integrated with the more traditional types of geospatial sources.

To address this challenge, researchers have developed machine learning techniques for rapidly converting online web sources into sources that can be queried as if they were databases. These techniques greatly simplify the problem of turning web pages into structured data. The user provides examples of the information to be extracted and the system learns a wrapper that can dynamically extract data from an online source or convert an online source into a database. A wrapper is defined by a set of extraction rules that are specific to extracting the data from a particular website. The extraction rules specify how to locate specific types of information from a page and these rules must work over the potentially large number of pages available on a given website. The machine learning techniques developed for this problem are designed to produce highly accurate extraction rules with a minimum number of training examples. The extraction rules for many websites can be learned with just a few examples.

(2) Integrating Vector Data and Imagery. When combining different geospatial data sources, such as vector data and satellite imagery, a critical problem is that the products do not correctly align. This problem is caused by the fact that geospatial data obtained from various sources may use different projections, may have different accuracy levels, and may have been corrected in different ways. The applications that integrate information from various geospatial data sources must be able to overcome these inconsistencies accurately and for large regions.

Traditionally, this problem has been solved in the domain of image processing and Geographic Information Systems (GIS). The focus of image processing techniques has been on automatic identification of objects in the image in order to resolve vector-image inconsistencies. However, these techniques require significant CPU time to process an image in its entirety and still may result in inaccurate results. The primary approach used in most GIS is to require a user to manually identify a set of control point pairs and then to use a technique called conflation to align two geospatial data sets. The need to manually identify control point pairs means that this approach does not scale up to large regions.

To address this problem, GIS engineers have developed a technique to efficiently and automatically integrate vector data with satellite or aerial imagery. The approach is based on two important observations. First, there is a great deal of information that is known about a given location beyond the data that is to be integrated. For example, most road-network vector data sets also contain the road direction, road width and even location of the road intersections. Second, rather than processing each source of information in isolation, it is much more effective to apply what is known about these sources to help in the integration of two sources. For example, the information about road directions and road width can be used to help locate the corresponding intersections in the satellite imagery.

Wrapper — зд. оболочка, интерфейс.

To orthorectify — ортотрансформировать (совершать геометрическую коррекцию изображения, во время которой исправляются существенные геометрические неточности, которые могут быть обусловлены топографией, геометрией камеры и ошибками сенсора).

Machine learning technique — метод машинного обучения.

Extraction rules — правило извлечения (данных).

Conflation — соединение, конфляция.

Aerial *imagery* — аэрофотосъемка.

25. Virtual Reality in Geographic Information Systems

Virtual reality (VR) emerged as part of the extension of computing into graphics which began to accelerate with the advent of the microprocessor and its widespread use in personnel computers, workstations, and supercomputers. Sometime in the mid-1980s, Jaron Lanier coined the term to describe the kinds of virtual environments popping up everywhere in which users had begun to hook themselves into computers in such as way that what they saw, heard, and touched was a product of the machine's simulation. The visual was the most important of these sensory experiences, and, nowadays, the typical portrait of VR is still the 3D world in which the user is immersed and able to navigate amongst movable objects. But as computers have fused with networks, the concept of VR has blurred, and no longer does the term imply total immersion in the machine's simulation. A more general definition from the computer and internet encyclopedia *Webopedia* is «an artificial environment created with computer hardware and software and presented to the user in such a way that it appears and feels like a real environment». Today the term is used to describe many experiences in environments which exist on everything from the desktop to the net.

We need to be a little more specific about how we are beginning to use the term in GI Science because the complete panoply of environments, media, and electronic peripherals used to generate as many sensory experiences as possible (fully-fledged VR) have not yet been widely exploited in our field. In fact GI Science is largely graphical, rooted in the 2D space, the map, extending into 3D, and even into the temporal dimension, but with little use of more esoteric virtual environment hardware and media. The main distinction we must make at the outset is one of abstraction: VR technologies can be and are being exploited in two directions, first with respect to how we interact with analytical conceptions of space, with surfaces and their properties in terms of their statistics, and, second, with respect to much more realistic conceptions of space in terms of landscapes where the quest for simulated realism is to the fore. In a sense, this is the difference between *geographic* and *geometrical analysis*, the difference between navigating in mathematical space and its various transforms in contrast to interacting with the Euclidean space of the three-dimensional world in which we live. Of course, GI Science spans the route between both conceptions of space, but the development of VR along this continuum marks out rather difference styles and methods.

In essence, VR here is limited to that constellation of users and technologies that enable interactions within a visual environment. Indeed, a good working definition of VR is «visualization + interaction + motion» where motion implies that user moves and consciously manipulates the environment in some way. Interaction is more than passive in that users continuously respond to cues within the environment which is configured for some particular task ranging from entertainment to scientific research.

In this sense, interactivity is the watchword of VR, and it is not surprising that this domain has been dominated by the development of computer-aided environments oriented to design and problem-solving. In this sense, spatial, particularly fine-scale design, such as computer-aided architecture design, has figured prominently in the development of VR.

The 3D geometric environment – rooms, buildings, cities, natural landscapes, and so on — represent the quintessential examples of VR. Many definitions of the term reflect the fact that the third dimension is all important in enabling users to truly immerse themselves in the digital environment. Since the late 1990s, 3D extrusion from the map to the cityscape or landscape has become possible, thus providing geometric frameworks in which spatial data can be queried and displayed in 3D, just as such techniques have become the workhorse of routine 2D GIS applications. In fact, there are extremely rapid developments in 3D GIS at present. There is now little doubt that the functionality that such systems offer in building such environments is considerably richer than the rather vacuous CAD methods that emphasize rendering rather than geometric database construction to which spatial attributes can be associated.

In a sense, the 3D graphic environments that can now be produced easily using GIS do not constitute virtual reality. They may be the basis for constructing virtual realities but such realities in the narrow sense depend upon how the user is immersed in the environment and this depends on the hardware and software used to make such environments work. In a broad sense of course, flying through a 3D model on the desktop might be considered part of VR (and increasingly is so as the definition continues to broaden) but it is worth keeping such graphics separate from the media of communication and interaction. In the past, there have been quite severe limits on what 3D GIS can handle, although these are fast being resolved. In particular, while moving through the model, detailed rendering has either not been possible or has been only primitive whether on-the-fly or using predetermined flight paths. In current generations of software, all these limits are being relaxed while the size of the environment that can be handled is being massively increased through better hardware and software. Embedding other kinds of multimedia into such environments such as panoramas has become possible and for the first time, it looks at through 3D GIS is becoming the preferred medium for such model construction.

GI Science (или *GI Sci — Geographic Information Science*) — геоинформатика.
Esoteric — зд. редкий, малораспространенный.
Transform — матем. преобразование.
3D extrusion — 3D экструзия. Создание 3D пространства.
Queried — зд. детализированный.
CAD (Computer-Aided Design) — автоматизированное проектирование.

Содержание

Учебное пособие

Ирина Сергеевна Дорошкевич,
Мария Александровна Морозова

АНГЛИЙСКИЙ ЯЗЫК. СБОРНИК ТЕКСТОВ ДЛЯ СТУДЕНТОВ ФАКУЛЬТЕТА ИНФОРМАЦИОННЫХ СИСТЕМ И ГЕОТЕХНОЛОГИЙ

Начальник РИО А.В. Ляхтейнен
Редактор Л.Ю. Кладова
Верстка М.В. Ивановой

Подписано в печать 03.09.18. Формат 60×90 $^1/_{16}$. Гарнитура Times New Roman.
Печать цифровая. Усл. печ. л. 3,75. Тираж 150 экз. Заказ № 693.
РГГМУ, 195196, Санкт-Петербург, Малоохтинский пр., 98.