UDC 004.942
МРНТИ 50.51.19

DOI: https://doi.org/10.37788/2023-3/ 134-142

I.I. Lyashenko
Innovative University of Eurasia, Kazakhstan
*(e-mail: irinaL72@yandex.ru)

On the application of CASE-methodologies at the stage of analysis and design of software tools

Annotation

Main problem: the globalization of the information sphere has led to a significant increase in the volume of stored and processed information. To increase the processing speed and provide information to the recipient, it is logical to increase the power of the equipment on which the information is processed. But such a path also leads to an increase in financial costs for continuous modernization or complete updating of computer equipment and application software.

Purpose: an alternative solution to increasing the capacity of hardware and software may be to optimize the process of organizing data storage, and already at the stage of information analysis. At the initial stages of the task, a huge amount of information is collected and processed, which needs to be classified, to identify essential and non-essential information flows for the information system being developed, etc. The quality of functioning of the developed information system as a whole and its individual modules in particular will depend on how correctly and fully the structuring and systematization of data will be carried out.

Methods: modern methodologies of conceptual design of information systems shift the emphasis from software implementation to analysis and modeling of the production environment. The tools used in the process of conceptual modeling allow you to automate the main part of the processes, including the development of program code in the format of templates (patterns), including the preparation of documentation for the software product being developed. Since modern methodologies and tools are based on the use of powerful graphical tools, this significantly increases the visibility and simplifies the process of making adjustments to the information system project, actually using the "Drag-and-Drop" technology. This allows you to redistribute the time spent on the implementation of the first stages of information system development, allocating more time to the analysis and design of the software environment at an abstract, conceptual level.

Results and their significance: the stages of analysis and design do not require large financial costs and are more variable. Information at these stages is easier to adjust and change than, for example, at the stage of writing program code. In addition, it is at these stages that the most active interaction between the developer and the customer is carried out, where maximum visibility and variability of the project of the future information system is required. And the use of modern software design methodologies significantly increases the efficiency of the first stages of information system design.

Keywords: model, CASE tools, code generation, IDEF methodologies, UML, software engineering.

Introduction

There are many approaches to designing complex systems. Comparative analysis shows that with the existing differences, almost all approaches include two stages: analysis and design of the subject area of the software tool. This is due to the fact that at the initial stages of the task, a huge amount of information is collected and processed, which needs to be classified, significant and insignificant flows of information for the information system being developed, etc. The quality of functioning of the developed information system as a whole and its individual modules in particular will depend on how correctly and fully the structuring and systematization of data will be carried out.

Materials and methods

Globalization in the field of information technology has given significant advantages to a huge number of users, but has significantly increased the volume of stored and processed information. One of the ways to increase the speed of processing and providing information to the recipient can be considered to increase the power of the equipment on which the information is stored. But this way also leads to an increase in financial costs for the constant modernization or complete renewal of the

computer fleet, as well as system and application software. At the same time, in most cases, it is possible to solve the problem of optimizing data processing and storage on existing equipment and installed software due to the correct structuring of the data itself, their storage and processing sequence. Thus, modern methods of software design shift the emphasis from the stages of software development and debugging to the stages of data analysis.

Indeed, the stages of analysis and design do not require large financial costs and are more variable. Information at these stages is easier to adjust and change than, for example, at the stage of writing program code. In addition, it is at these stages that the most active interaction between the developer and the customer is carried out, where maximum visibility and variability of the project of the future information system is required. And the use of modern software design methodologies significantly increases the efficiency of the first stages of information system design.

Modern methodologies of conceptual design of information systems shift the emphasis from software implementation to analysis and modeling of the production environment. The software development tools used in the process of conceptual modeling allow you to automate the process of code generation and compilation of documentation for the software product being developed. In addition, modern methodologies and tools are based on the use of powerful graphical tools, which also increases visibility and simplifies the possibility of making changes to the designed information system and software product, actually using the "Drag-and-Drop" technology. This is what makes it possible to redistribute the time spent on the implementation of the first stages of the information system lifecycle, allocating more time to the analysis and design of the software environment from various points of view at the abstract, conceptual level. This, in turn, makes it possible to optimize the storage structure and algorithms for processing large data streams that make up the database of any information system, which ultimately leads to optimal use of technical resources and an increase in the time of use of the system.

Design can be defined as a description of an object that does not yet exist under specified conditions, as well as the choice of appropriate conditions for its functioning. [1]. Conceptual design is a composite stage performed at the initial stages of design, when the task is formed under conditions of uncertainty, that is, it has a research, creative character. Thus, conceptual design is more dynamic and variable compared to the stages of software development and implementation of an information system. And for the conceptual design stage, there are also methods and technologies for its implementation. Of the modern methodologies for designing information systems, the methods and technologies of the IDEF (Integrated computer-aided manufacturing Definition) family should be noted. IDEF methodologies allow you to build a model of an information system from various points of view, which significantly increases the efficiency of the developer's work at the first stages of design, allowing you to use powerful graphical tools when building a model consisting of a complex of diagrams.

For example, in the conceptual design of information systems, the following IDEF family methodologies are most often used:

- IDEF0 is a functional modeling method designed to simulate decisions, actions and activities of an organization or system [2].
- IDEF1X is an information modeling method designed for the development of relational databases. [2].
- IDEF3 is a dynamic modeling method that provides a description of the behavioral aspects of the system being developed. [2].

Thus, the functional component of the information system project can be represented in the notation of the IDEF0 model, the elements of the data processing system can be described in the format of an information model in the IDEF1X notation, and the methodology of dynamic modeling IDEF3 will describe the main organizational and business processes of the designed system.

There are quite a large number of tools for developing graphic objects. From this class, it is necessary to allocate tools containing a powerful set of graphical tools, as well as having functionality that allows you to automate part of the operations. Such tools belong to the class of CASE-tools. The concept of CASE (Computer Aided Software Engineering) is understood as software packages for automating the main stages of information systems development [3], including database design, code generation and testing.

Of the many popular tools that support IDEF family notation, the most famous is the AllFusion Process Modeler software package.

AllFusion Process Modeler (BPwin) refers to tools for visual modeling of business processes that do not require writing program code. The software package provides the developer to optimize the information system design process in order to eliminate unnecessary operations already at the first stages of development, increase the dynamism and efficiency of the information system.

For modeling the information environment, the most well-known CASE tools include the ERwin Data Modeler software package, which allows you to visually display complex data structures, as well as build a future database project in the IDEF1X methodology. In addition, the ERwin Data Modeler tool allows you to automate the process of generating a real database from a model to one of the well-known data management systems: MS Access, MySQL, SQL Server.

It should be noted that these tools currently have no updates for modern operating systems, but are still used for designing information systems in the environment of system analysts and designers. The most popular "descendant" of the IDEF methodology is the unified modeling language UML (Unified Modeling Language), which supports a powerful apparatus of graphical tools for object modeling of software systems and business processes. This language has become a successful generalization of the best achievements in the field of system analysis and system design, which are possessed, among others, by IDEF methodologies. At the same time, tools that support the notation of the UML language are very popular among developers, since they support all stages of the software lifecycle. It should be noted that in the free versions of such packages, access is limited only to domain modeling. The full package of tools requires the purchase of a license to use, which somewhat limits the range of users. On the other hand, large corporations specializing in the development of software products, as well as solving business process optimization problems, prefer to purchase a complete package with a wide range of functionality. This is due to the fact that the automation of code generation processes practically eliminates the risk of errors at the programming and debugging stage, since the tools allow the developer to correct them already at the design stage, most often together with the customer. For large IT companies, this approach provides the strongest time savings for the development of software products, which allows developers to focus on modeling a future information system that would not only have optimal resources in terms of information processing speed and the amount of memory occupied, but also solved the tasks stated by the customer as accurately as possible. It is these goals that are achieved at the stage of analysis and design of the information system.

Results

As a result of the analysis of CASE-methodologies and an example of their application as a means of conceptual design, the design of an information module of one of the many business processes of an educational institution is further considered, in particular, the process of accounting for the control of students' knowledge. The construction of the functional model of the information subsystem was carried out using the IDEF0 methodology. At the top level there is a contextual diagram describing the business process in the "first approximation".

Figure 1 shows the only functional block of the top level of the diagram, which includes the business process, as well as all information and material flows, control actions and mechanisms of the described process.

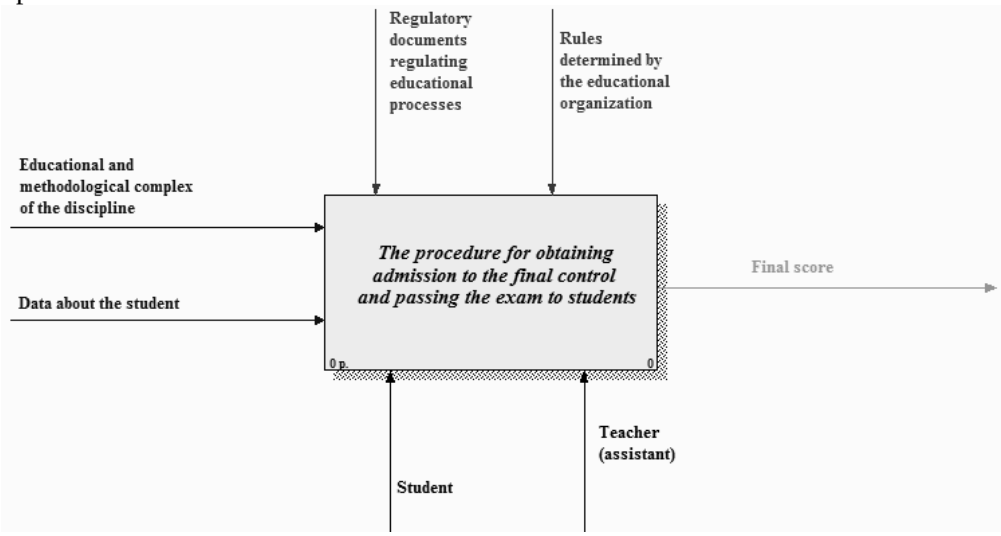

Figure 1 - IDEF0 context diagram (top level)

The lower level of the diagram allows you to describe the business process in more detail as a sequence of functional blocks connected by information or material flows. Figure 2 shows the lower level of the functional business process modeling diagram in the IDEF0 methodology.

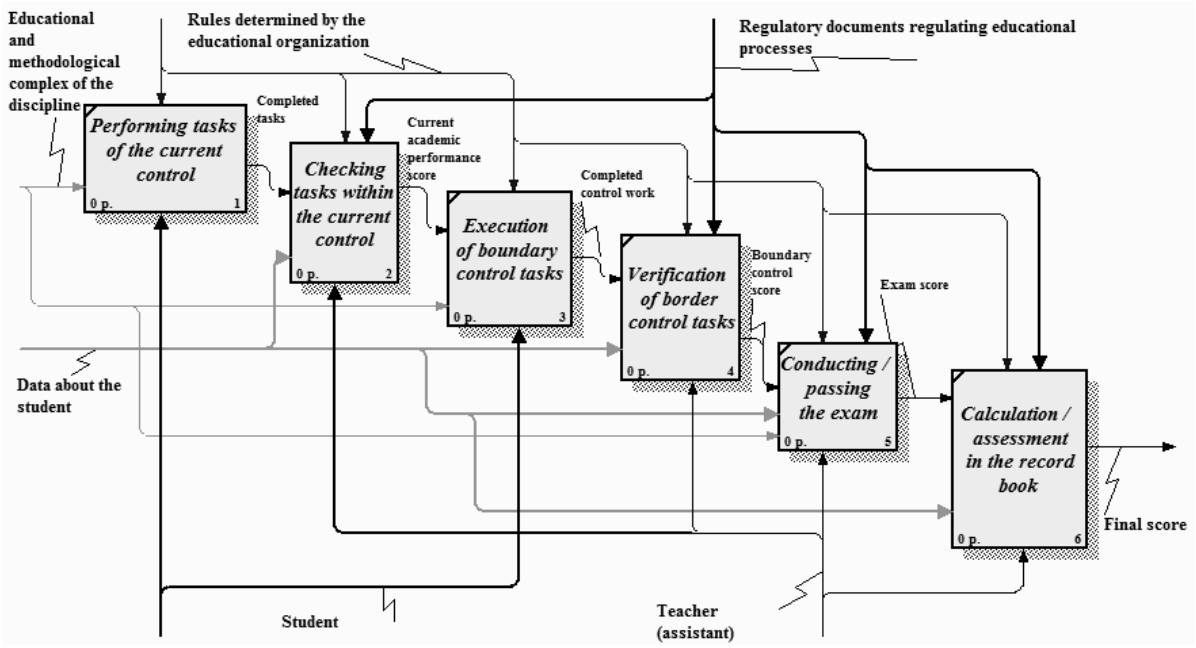

Figure 2 - IDEF0 context diagram (lower level)

Any functional block of the diagram can be decomposed at the lower level to increase the level of detail of understanding according to the principle "from a more general view to a more detailed one" [4]. An example of such a decomposition is made for the functional block "Checking tasks within the current control" and is shown in Figure 3.

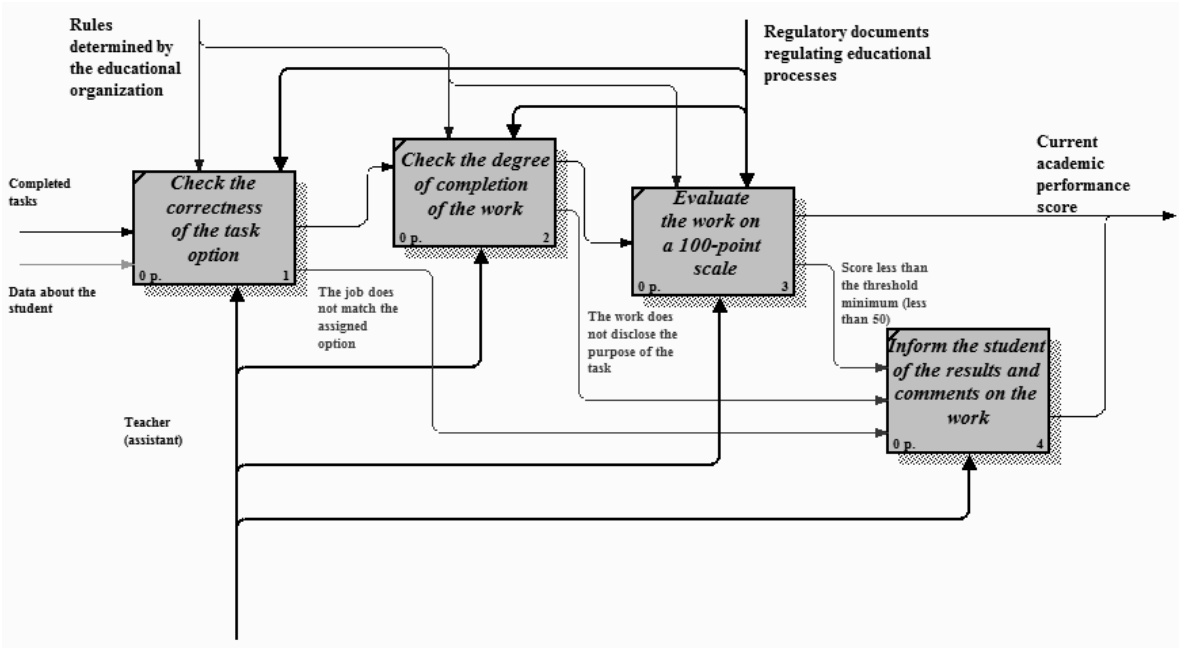

Figure 3 – Decomposition of the functional block "Checking tasks within the current control"

As can be seen in the diagram, the process was divided into separate blocks, in accordance with their sequence of execution, approved by regulatory documents and requirements of the higher educational institution. In addition, the diagram shows the information flows required for the initialization of each functional block, as well as the result that each block should end with.

Modeling of the data processing and storage system (database design) is performed in the IDEF1X notation. The ERwin Data Modeler program is selected as a tool. Figure 4 shows a logical database model in the methodology of ER diagrams, containing information about the student's progress "Record book".

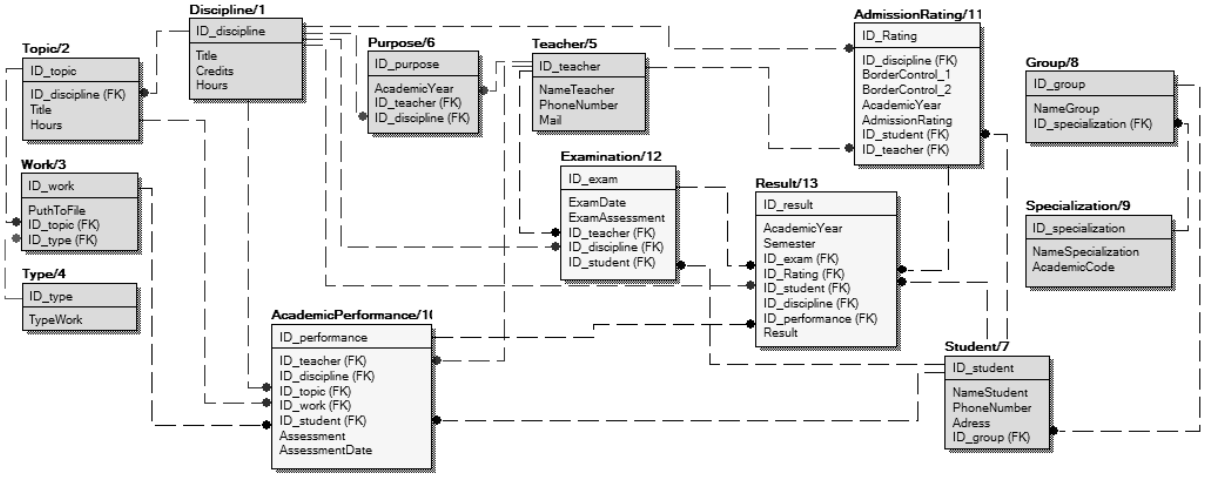

Figure 4 – IDEF1X-diagram of the information data model

The ERwin Data Modeler tool, in the environment of which a data diagram is built, allows you to demonstrate the automated process of creating a real database using the CASE-tool functionality. Figure 5 shows the result of automatic code generation of an IDEF1X diagram into MS Access format.

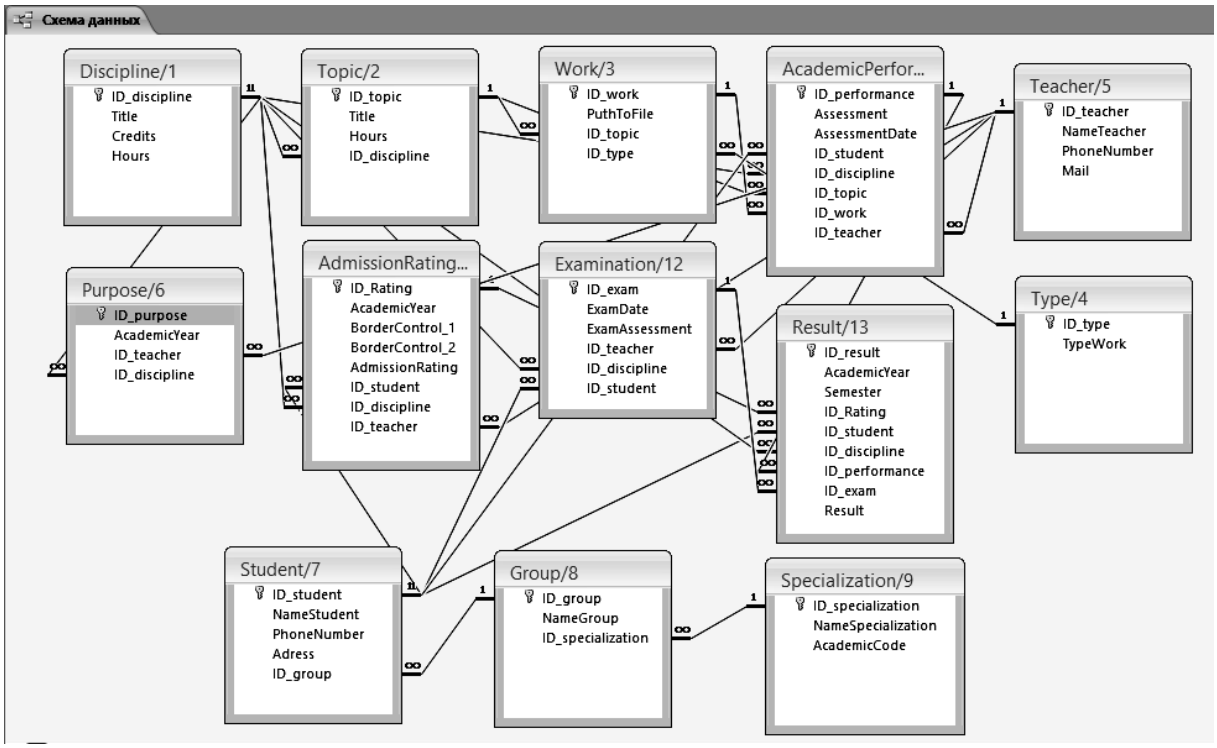

Figure 5 – The result of automatic code generation of the IDEF1X model into the MS Access database format

As can be seen in Figure 5, all tables, the relationships between them, as well as the types of relationships are stored strictly in accordance with the model. The algorithm of actions for generating a database into an MS SQL environment is almost identical, except that you need to choose an environment for exporting MS SQL. The structure of the database automatically generated from the model in MS SQL is shown in Figure 6.

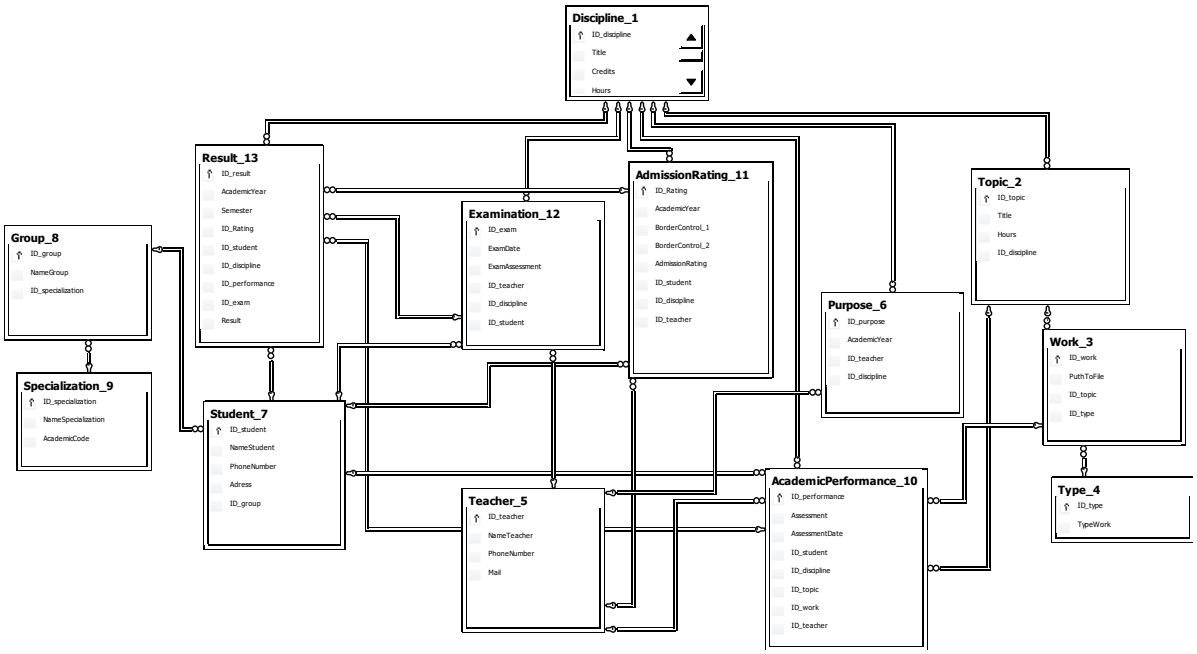

Figure 6 – The result of automatic code generation of the IDEF1X model into the MS SQL database format

Discussion

At the same time, the process of generating database tables takes a much shorter period of time compared to the "manual" creation of a database, which clearly demonstrates the advantages of using specialized tools for designing information systems. It can also be noted that the database model in the Erwin methodology is much clearer due to the wide graphical capabilities of the ERwin Data Modeler CASE tool [5].

Designing a dynamic model of an information system represents the process of describing and analyzing the system in terms of a sequence of actions (operations), taking into account many different "alternative" processes (branches) [6]. As an example of building such a model, Figure 6 shows a diagram of activities in the Visual Paradigm for UML Community Edition tool environment.

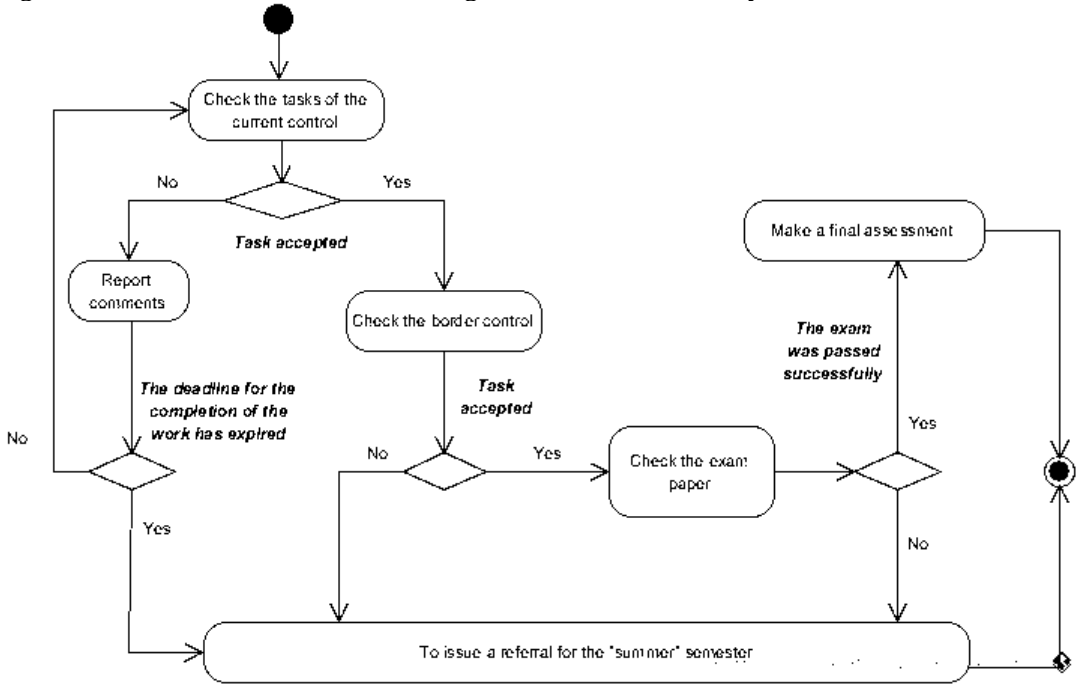

Figure 7 – Activity diagram in the UML notation of the process of keeping track of progress in the record book

The UML (Unified Modeling Language) notation includes a powerful apparatus for modeling the subject area from various points of view. It should be noted that the UML methodology allows you to design information systems in similar ICAM notations. So, for example, the Use Case diagram in UML notation corresponds to the IDEF0 diagram, and the activity diagram is similar in semantics to the IDEF3 diagram of the ICAM methodology.

The list of tools that support UML notation includes more than 300 software products. Full versions of CASE tools support functionality that includes model generation in real object-oriented programming and data management systems. For example, the Visual Paradigm for UML tool environment allows you to perform code generation in such object programming environments as Java, C++, PHP, Python.

Conclusion

The given example of building functional and information models of one of the business processes using instrumental CASE tools clearly demonstrates the advantages of automating the design process over the traditional approach of developing information products in the IT sphere. So, even at the stage of designing an information subsystem, graphical models of business processes allow designers to demonstrate not only the structure of the future information system, but also its functionality as clearly as possible. In addition, automation tools can significantly reduce the design time of an information system, which is especially important for systems with a large number of modules and subsystems. In addition, the process of designing information systems in the traditional way becomes more complicated if it is necessary to make changes to an existing database. The risk of such problems is minimized if modern CASE-methodologies and their supporting tools are used, which clearly proves the relevance of using modern methods in the process of designing and developing information systems.

THE LIST OF SOURCES

1 Андрейчиков А.В. Методы и интеллектуальные системы анализа и синтеза новых технических решений. – М.: РИОР, 2019. – 544 с.
2 Суркова Н.Е. Методология структурного проектирования информационных систем: монография. - Красноярск: Изд. НИЦ, 2014. – 190 с.
3 Таршхоева Ж.Т. CASE-технологии разработки программных систем // Молодой ученый. - 2021. - № 34. - С. 1-3.
4 Кара-Ушанов В.Ю. Функционально-структурное моделирование в системе RAMUS EDUCATIONAL. – Екатеринбург: Изд. УФУ, 2019. – 66 с.
5 Попова-Коварцева Д. А. Основы современных технологий баз данных. – Самара: СУ, 2019. – 92 с.
6 Каюмова А.В. Визуальное моделирование систем в StarUML. – Казань: Изд.-во Казанский федеральный университет, 2013. – 104 с.

REFERENCES

1 Andrejchikov, A.V. (2019). Metody i intellektual'nye sistemy analiza i sinteza novyh tekhnicheskih reshenij [Methods and intelligent systems for the analysis and synthesis of new technical solutions] M: RIOR [in Russian].
2 Surkova, N.E. (2014). Metodologiya strukturnogo proektirovaniya informacionnyh sistem: monografiya [Methodology of structural design of information systems: monograph] Krasnoyarsk: NIC [in Russian].
3 Tarshkhoeva, Zh.T. (2021). CASE-tekhnologii razrabotki programmnyh sistem [CASE-software system development technologies]. Molodoj uchenyj, 34, 1-3 [in Russian]
4 Kara-Ushanov, V.Yu. (2019). Funkcional'no-strukturnoe modelirovanie v sisteme RAMUS EDUCATIONAL [Functional and structural modeling in the RAMUS EDUCATIONAL system] Ekaterinburg: UFU [in Russian].
5 Popova-Kovarceva, D.A. (2019). Osnovy sovremennyh tekhnologij baz dannyh [Fundamentals of modern database technologies] Samara: SU [in Russian].
6 Kayumova, A.V. (2013). Vizual'noe modelirovanie sistem v StarUML [Visual modeling of systems in StarUML] Kazan': Kazanskij federal'nyj universitet [in Russian].

И.И. Ляшенко
Инновациялық Еуразия университеті, Қазақстан

Бағдарламалық құралдарды талдау және жобалау кезеңінде CASE-әдістемелерді қолдану туралы

Ақпараттық саланың жаһандануы сақталған және өңделетін ақпарат көлемінің айтарлықтай өсуіне әкелді. Өңдеу жылдамдығын арттыру және алушыға ақпарат беру үшін ақпарат өңделетін жабдықтың қуатын арттыру қисынды. Бірақ мұндай жол компьютерлік жабдықтар мен қолданбалы бағдарламалық жасақтаманы үнемі жаңартуға немесе толық жаңартуға қаржылық шығындардың артуына әкеледі.

Аппараттық және бағдарламалық жасақтаманың қуатын арттыруға балама шешім деректерді сақтауды ұйымдастыру үрдісін оңтайландыру және ақпаратты талдау кезеңінде болуы мүмкін. Тапсырманың бастапқы кезеңдерінде көптеген ақпаратты жинау және өңдеу жүзеге асырылады, оларды жіктеу керек, әзірленіп жатқан ақпараттық жүйе үшін маңызды және маңызды емес ақпарат ағындарын және т.б. бөліп көрсету керек. Тұтастай алғанда әзірленген ақпараттық жүйенің және оның жекелеген модульдерінің жұмыс істеу сапасы деректерді құрылымдау мен жүйелеудің қаншалықты дұрыс және толық орындалатынына байланысты болады.

Ақпараттық жүйелерді тұжырымдамалық жобалаудың заманауи әдістемелері бағдарламалық іске асырудан өндірістік ортаны талдау мен модельдеуге баса назар аударады. Тұжырымдамалық модельдеу үрдісінде қолданылатын құралдар процестердің негізгі бөлігін автоматтандыруға мүмкіндік береді, соның ішінде шаблондар (үлгілер) форматында бағдарламалық кодты әзірлеу, соның ішінде әзірленіп жатқан бағдарламалық өнім бойынша құжаттама жасау. Заманауи әдістемелер мен құралдар қуатты графикалық құралдарды қолдануға негізделгендіктен, бұл көрнекілікті едәуір арттырады және "Drag - and-Drop" технологиясын қолдана отырып, ақпараттық жүйенің жобасына түзетулер енгізу процесін жеңілдетеді. Бұл абстрактілі, тұжырымдамалық деңгейде бағдарламалық ортаны талдауға және жобалауға көбірек уақыт бөліп, ақпараттық жүйені дамытудың алғашқы кезеңдерін аяқтауға кететін уақытты қайта бөлуге мүмкіндік береді.

Талдау және жобалау кезеңдері үлкен қаржылық шығындарды қажет етпейді және өзгермелі. Осы кезеңдердегі ақпаратты, мысалы, бағдарламалық кодты жазу кезеңіне қарағанда, түзету және өзгерту оңайырақ. Сонымен қатар, дәл осы кезеңдерде әзірлеуші мен Тапсырыс берушінің ең белсенді өзара әрекеттесуі жүзеге асырылады, онда болашақ ақпараттық жүйе жобасының максималды көрнекілігі мен өзгергіштігі қажет. Бағдарламалық жасақтаманы жобалаудың заманауи әдістемелерін қолдану ақпараттық жүйені жобалаудың алғашқы кезеңдерінің тиімділігін едәуір арттырады.

Түйінді сөздер: модель, кейс-құралдар, кодогенерация, IDEF, UML, software engineering әдістемелері.

И.И. Ляшенко
Инновационный Евразийский университет, Казахстан

О применении CASE-методологий на этапе анализа и проектирования программных средств

Глобализация информационной сферы привела к значительному увеличению объемов хранимой и обрабатываемой информации. Для повышения скорости обработки и предоставления информации получателю логично увеличение мощности оборудования, на котором информация обрабатывается. Но такой путь приводит и к повышению финансовых затрат на постоянную модернизацию или полное обновление компьютерного оборудования и прикладного программного обеспечения.

Решением, альтернативным повышению мощностей аппаратного и программного обеспечения, может стать оптимизация процесса организации хранения данных, причем уже на этапе анализа информации. На начальных этапах поставленной задачи выполняется сбор и обработка огромного количества информации, которую нужно классифицировать, выделить

существенные и несущественные для разрабатываемой информационной системы потоки информации и т.п. От того, насколько корректно и полно будет выполнена структуризация и систематизация данных, будет зависеть качество функционирования разработанной информационной системы в целом и ее отдельных модулей в частности.

Современные методологии концептуального проектирования информационных систем переносят акцент с программной реализации на анализ и моделирование производственной среды. Инструментальные средства, используемые в процессе концептуального моделирования, позволяют автоматизировать основную часть процессов, включая разработку программного кода в формате шаблонов (паттернов), включая составление документации по разрабатываемому программному продукту. Поскольку современные методологии и инструментальные средства базируются на использовании мощных графических средств, это значительно повышает наглядность и упрощает процесс внесения корректировок в проект информационной системы, фактически используя технологию «Drag- and-Drop». Это позволяет перераспределить время, затрачиваемое на выполнение первых этапов разработки информационной системы, выделив больше времени на анализ и проектирование программной среды на абстрактном, концептуальном уровне.

Этапы анализа и проектирования не требуют больших финансовых затрат и более вариативны. Информация на этих этапах легче поддается корректировке и изменениям, чем, к примеру, на этапе написания программного кода. Кроме того, именно на этих этапах выполняется наиболее активное взаимодействие разработчика и заказчика, где требуется максимальная наглядность и вариативность проекта будущей информационной системы. Использование современных методологий проектирования программного обеспечения значительно повышает эффективность выполнения первых этапов проектирования информационной системы.

Ключевые слова: модель, CASE-средства, кодогенерация, методологии IDEF, UML, software engineering.

Date of receipt of the manuscript to the editor: 2023/09/03