

РЕДАКЦІЙНА КОЛЕГІЯ:

Головний редактор:

проф., д.т.н. Рудь В.Д. (м.Луцьк)

Перший заступник головного редактора:

проф., д.т.н. Делявський М.В. (м.Львів)

Другий заступник головного редактора:

доц., к.т.н. Герасимчук О.О. (м.Луцьк)

доц., к.пед.н. Панасюк Н.Л. (м.Луцьк)

Редактори-коректори:

доц., к.т.н. Мельник К.В. (м.Луцьк)

інженер Свиридчук К.А. (м.Луцьк)

Члени редакційної колегії:

проф., д.т.н. Мазур М.П. (м.Хмельницьк)

проф., д.т.н. Пальчевський Б.О. (м.Луцьк)

проф., д.т.н. Мельник А.О. (м.Київ)

проф., д.пед.н. Горбатюк Р.М. (м.Тернопіль)

проф., д.пед.н. Поліщук Ю.Й. (м.Тернопіль)

проф., д.пед.н. Романишина Л.М. (м.Київ)

проф., д.пед.н. Свистун В.І. (м.Київ)

проф., д.т.н. Сидорчук О.В. (м.Київ)

проф., д.т.н. Тарасенко В.П. (м.Київ)

проф, PhD. Мілош Марек (Польща, м.Люблін)

доц., к.т.н. Лотиш В.В. (м.Луцьк)

доц., к.т.н. Гуменюк Л.О. (м.Луцьк)

доц., к.т.н. Кузнецов Р.М. (м.Луцьк)

доц., к.т.н. Пех П.А. (м.Луцьк)

доц., к.т.н. Повстяной О.Ю. (м.Луцьк)

доц., к.пед.н. Потапук Л.М. (м.Луцьк)

доц., к.т.н. Решетило О.М. (м.Луцьк)

Адреса редколегії:

Луцький національний технічний університет,
кафедра комп'ютерної інженерії.

вул.Львівська 75, ауд.141

м.Луцьк, 43018

тел. (0332) 74-61-15

E-mail: cit@lntu.edu.ua,

ekaterinamelnik@gmail.com

сайт журналу: **ki.lutsk-ntu.com.ua**

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ **ТЕХНОЛОГІЇ:** **ОСВІТА, НАУКА, ВИРОБНИЦТВО**

№21 2015р.

Журнал засновано у грудні 2010 р.
Свідectво про реєстрацію KB № 16705-5277 Р.
Засновник: Луцький національний технічний університет
Рекомендовано до друку Вченою радою
Луцького національного технічного університету
(протокол №13 засідання від 23.06.2015)
Журнал рішенням президії ВАК України
від 30 березня 2011
№1-05/3 включено в перелік наукових фахових видань.
Журнал має російський індекс наукового цитування
(РІНЦ)

ISBN:978-617-672-040-9

ЗМІСТ

Інформатика та обчислювальна техніка

Максимович О.В., Мельник В.М., Бортник К.Я., Лавренчук С.В., Самарчук В.Ф. Pliability of faults in open stack versions.(Стійкість до збоїв версій відкритих стеків.)	5
Максимович О.В., Мельник К.В., Мельник В.М., Гадай О.С. Audio player with a library of Delphi. (Аудіоплеєр з фонотекою засобами мови Delphi.).	21
Бортник К.Я. Структура системи багатошарового перцептронну для мобільних пристроїв.	25
Гринюк С.В., Міскевич О.І. Автоматизована інформаційна система обліку та руху товарів на складі.	32
Губаль Г. М. Аналіз деяких команд мови створення математичних текстів L ^A T _E X.	37
Коцюба А.Ю., Лавренчук С.В., Остапчук Р.В. Методика побудови алгоритмів генерації варіантів головоломки Судоку.	42
Марченко О.І., Марченко О.О. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло.	51

Автоматика та управління

Андрущак І.Є. Мультиваріативний метод якісного аналізу моделі на основі класифікаційних правил.	58
Грінченко Л.Г. Поперечний згин тонкої ізотропної прямокутної плити, жорстко закріпленої на контурі.	64
Ісмілова Н.П. Твердотільне моделювання спряжених поверхонь «Евольвентний гелікоід – конус обертання» на базі параметричного кінематичного гвинта.	69
Каганюк А.К., Желобицкий Я.К. Внедрение инновационных технологий струнного транспорта Юницкого «Skyway» в существующую транспортную систему.	75
Каганюк О.К., Поліщук М.М. Використання інноваційних технологій в розпізнаванні транспортних засобів за допомогою радіолокаційних методів.	82
Каганюк О.К., Сопіжук Р.В. Розробка апаратно-програмного вимірювального комплексу для діагностики та дослідження технічного стану двигунів внутрішнього згорання за допомогою мікроконтролера ELM327.	87
Рудь Н.Т. Математичне моделювання процесів трансферу технологій.	94
Рязанцев О.І., Кардашук В.С., Бортник К. Система автоматизації керування технологічним процесом виробництва желатину.	100
Федік Л.Ю., Балинський А.О. Особливості конструкції і керування автомобілями амфібіями.	106

Педагогічні науки

Бондаренко Т.С. Система хмарного моніторингу якості професійної підготовки майбутніх кваліфікованих робітників.	114
Зубехіна Т. В. Аналіз результатів педагогічного експерименту щодо формування інформаційної культури бакалаврів з туризму.	118
Мельник В.М., Бортник К.Я., Чеб С.С., Міскевич О.І. Деякі перспективи впровадження комунікаційних інтернет-технологій в системі освіти України.	123
Ліщина Наталія. Аналіз сучасних технологічних платформ для створення системи електронного документообігу університету.	128
Панасюк Н. Conceptual approachs to managing the quality of education in higher technical educational institutions. (Концептуальні підходи до управління якістю освіти у вищих технічних навчальних закладах.)	133
Сидоров М.О., Жигаревич О.К. Систематичне дослідження літературних джерел щодо екосистем навчального програмного забезпечення.	137

UDK 004.254 (045)

Maksymovych O.V., Melnyk V.M., Bortnyk K.Ya., Lavrenchuk S.V., Samarchuk V.F.
Lutsk national technical university

PLIABILITY OF FAULTS IN OPEN STACK VERSIONS

Maksymovych O.V., Melnyk V.M., Bortnyk K.Ya., Lavrenchuk S.V., Samarchuk V.F. Pliability of faults in open stack versions. *Stacks of cloud-management have become an important elements in cloud computing. They are serving as the resource managers for cloud platforms. While their functionality has been constantly expanding, their fault resilience remains under-studied. This article presents a fault resilience systematic study of OpenStack – a popular cloud-management stack for the open source. We have built an example fault-injection framework directing service communications during the external requests processing, both among OpenStack services and between OpenStack and external services, and have thus far uncovered 23 bugs in two OpenStack versions. Our findings hut light on defects in the design and implementation of advanced cloud management stacks from a fault-resilience perspective.*

Keywords: reliability, fault-injection framework, fault-resilience perspective, cloud-management stack, open stack.

Максимович О.В., Мельник В.М., Бортник К.Я., Лавренчук С.В., Самарчук В.Ф. Стійкість до збоїв версій відкритих стеків. *Стеки управління cloud'ами стали важливим елементом cloud обчислень. Вони слугують менеджерами ресурсів для cloud-платформ. У той час як їхня функціональність постійно розширюється, їхня стійкість до збоїв не до кінця вивчена. Ця стаття презентує системне дослідження стійкості до збоїв Open Stack – популярного стеку керування cloud'ами на базі відкритого коду. Створено приклад структури направляючого комунікаційного сервісу для генерування помилок під час обробки зовнішніх запитів, для роботи як серед сервісів OpenStack, так і для роботи між OpenStack та зовнішніми сервісами, і, таким чином, виявлено на даний момент 23 необроблені помилки у двох версіях OpenStack. Пророблені пошуки просвітили погляд на недоліки в проектуванні і реалізації поширених стеків управління cloud'ами з точки зору перспективи стійкості до збоїв.*

Ключові слова: надійність, структура генерування помилок, перспектива стійкості до збоїв, стек управління cloud'ом, відкритий стек.

Максимович О.В., Мельник В.М., Бортник К.Я., Лавренчук С.В., Самарчук В.Ф. Устойчивость к сбоям версий открытых стеков. *Стеки управления cloud'ами стали важным элементом cloud-вычислений. Они служат менеджерами ресурсов для cloud-платформ. В то время как их функциональность постоянно расширяется, их устойчивость к сбоям не до конца изучена. Эта статья представляет системное исследование устойчивости к сбоям Open Stack – популярного стека управления cloud'ом на базе открытого кода. Создан пример структуры направляющего коммуникационного сервиса для генерации ошибок при обработке внешних запросов, для работы как среди сервисов OpenStack, так и для работы между OpenStack и внешними сервисами, и, таким образом, обнаружено на данный момент 23 необработанные ошибки в двух версиях OpenStack. Прделанные поиски сделали просвет на недостатки в проектировании и реализации распространенных стеков управления cloud'ами с точки зрения перспективы устойчивости к сбоям.*

Ключевые слова: надежность, структура генерации ошибок, перспектива устойчивости к сбоям, стек управления облаком, открытый стек.

Introduction

With the virtual machine (VM) technology development in both: software design and hardware support, cloud computing has become a major computing platform. In addition to public cloud services that have been available since the early stage of cloud platform deployment [1], there is an evolving demand for other types of cloud platforms, particularly, private and hybrid clouds. The demand leads to the role transition from cloud users to a cloud users and providers combination, broadening the scope of cloud providers from major IT companies to any size organizations. It has also prompted the cloud management stack research and development – a new software stack functioning as a high-level cloud operating system and remains key to resource management in cloud platforms.

In recent years an attention to cloud-management stacks from academy and industry has led to a rapid increase in the number of features. However, fault resilience of this layer, as an optional feature, is still regarded and remains under-studied, despite its importance demonstrated by real-world failures and its significant impact on managed cloud platforms [2, 3, 4]. Fault-resilience-related issues constantly stun the users of cloud-management stacks. For example, when faults occur, VM creation may fail or take some long time, and VMs may be marked as successfully created but lack critical resources (IP addresses), thus remaining unusable. A fault resilience investigation of cloud-management stacks is long overdue that demystifies the above issues.

In this paper, the first systematic study on the fault resilience of OpenStack is present, as a popular open source cloud-management stack. The conventional wisdom in the fault-injection literature as well as its application in fault-injection studies directing large-scale distributed systems [5, 6, 7], that studied the

OpenStack execution in the two common faults presence with the cloud environment: server crashes and network partitions. OpenStack is considered fault-resilient during the external request processing, if it maintains correct and steady states and behaviours, even in case of faults occurrence. As external requests are an important source of inputs to OpenStack and usually trigger state transitions, focusing on OpenStack's fault resilience during external request processing. We insert faults into inter-service communications during request processing, as they characterize service collaboration of which design and implementation is difficult to be fault resilient. Specifically, there are targeted communications among OpenStack's computing, image, identity services, as well as external services such as databases, hypervisors, and messaging.

An approach is taken in this fault-injection study, to expose high-level OpenStack semantics (service A sends a request R to service B via communication channel C) by supplementing its wrapper layer of communication libraries with our sorting and module of coordination. Exposing, instead of inferring, high-level semantics reduces the amount of logs, simplifies the extraction of communication patterns and facilitates efficient fault injection. This approach can also be easily integrated into OpenStack's notification mechanism. It closely mirrors OpenStack's existing logging infrastructure. To mean broader, this white-box approach is valuable to the DevOps integration current trend [8], allowing developers and operators to better understand the software functioning in realistic deployment environments. It facilitates the spectrum design of approaches to hardening cloud-management stacks, such as fault-injection studies, and online fault detection and analysis, which are planned to explore in future.

11 external APIs of OpenStack were been studied and for each API were been executed late request with identifying all fault injection cases, each corresponding to the combination of a fault type and location in the request execution path. A single-fault injections were conducted by re-executing the same request and iterating through the fault-injection cases, each time injecting a distinct fault into the execution flow. Upon completion of fault injection experiments, the results were checked against predefined specifications regarding the expected states and OpenStack behaviours. When specifications are violated, the execution of OpenStack and bugs identifying were manually investigated.

Two OpenStack versions were studied, namely, "essex" and "grizzly", the latter being the first version of the most recent release series, and identify in total 23 bugs. As in the preliminary work version [9], those bugs were been categorized into seven groups and performed an in-depth study for each category. several common fault-resilience issues were been identified then in OpenStack, such as permanent service blocking due to the timeout protection lack, irrecoverable inconsistent system states due to the lack of periodic checking and state stabilization, and misleading behaviours due to code checking of the incautious return. The major contributions of this paper are three-fold:

1. Applied fault-injection techniques to cloud management stacks and presented design and implementation for an operational prototype fault injection framework for this emerging software layer, using OpenStack as the study target.
2. Conducted the first systematic fault-resilience OpenStack study, identifying 23 bugs.
3. Categorized bugs, presenting deep analysis for each bug category, and discussing related fault resilience issues.

Cloud-Management Stack and OpenStack Background

It is briefly discussed cloud-management stacks and then provided background information about OpenStack, concentrating on its components, supporting services, communication mechanisms, and threading model. Cloud-management stacks are an emerging software layer in the cloud ecosystem. They are responsible for the cloud platforms formation and management. A cloud management stack manages cloud platforms via distributed services cooperation, which including an external API service for communicating with external users, an image service for managing VM images (registration and deployment), a computer service for managing VMs (creating and deleting VMs) on supporting hosts, a volume service for managing persistent storage used by VMs (providing block devices and object stores) and network services for managing networks used by VMs (creating and deleting networks, manipulating firewalls on supporting hosts). Its own service as a cloud-management stack requires external services to fulfil its functionality. In particular, it often relies on a hypervisor, described in [10] (Xen), [11] (KVM) or [12] (Hyper-V) for managing VMs.

OpenStack is a state-of-the-art open source cloud management stack [13], implemented in Python. It contains several common services, such as a computation service group, an image service group, network service, and several persistent storage services. Other OpenStack services in a typical cloud setting include an identity service for validating services and users and a dashboard service for graphical interface providing to users and administrators. OpenStack relies on hypervisors installed on computer nodes (where VMs run) for VM management and uses database service to store persistent states related to its managed cloud.

OpenStack employs two major communication mechanisms. Compute services use remote procedure calls (RPCs) compatible to the Advanced Message Queuing Protocol (AMQP) for internal communications within the service group. Other OpenStack services conform to the RE presentational State Transfer (REST) architecture and communicate with each other via the Web Server Gateway Interface (WSGI). OpenStack uses the SQLAlchemy library to communicate with database back ends, such as MySQL and SQLite. Interaction with hypervisors is inattentive to virtualization drivers. Specifically, OpenStack designs a common hypervisor-driver interface and implements drivers using common hypervisor APIs (libvirt and Xen). Its services are implemented as green threads via event let and green let libraries, which employ a user-level cooperative multithreading model: a thread runs non-pre-emptively until it surrenders control. Upon thread yielding, a hub thread turn out to be active, makes a scheduling decision and then transfers control to the scheduled thread. This model requires several standard Python libraries to be patched with green thread-compatible implementations in order to prevent I/O functions issued by one green thread from blocking the other in same process.

Project Scope, Design Principles, Components and Workflow Overview

This section presents the project scope, followed by our design principles discussion. It is present an overview of the components and the workflow of our fault-injection framework. The fault-resilience-related programming bugs are target in OpenStack. They affect OpenStack's inherent fault-resilience from its design and implementation perspective. Configuration bugs, in contrast, are considered faults in this article. For example, a mistaken configuration may lead to network partitions, which are used for fault injection in our framework. Bugs that can only be manifested by a sequence of faults are not in this paper scope, due to single-fault injections use.

Design builds on prior research in distributed systems tracing, fault injection, and specification checking. Instead of proposing a new fault-injection methodology, it is discussed experience in building an operational fault-injection prototype for OpenStack, following below design principles. Cloud management stacks rely on the services cooperation distributed to a cloud environment to fulfil their functionality. This cooperation requires fault-resilient communication mechanisms. Given the service communications importance and the fast advances of sophisticated single-process debugging techniques, fault injection prototype targets service communications in OpenStack.

Domain knowledge has proven valuable for debugging, monitoring, and analysing distributed systems. In [14] showed that developers of applications running in a distributed environment were willing to expose and exploit domain knowledge in a production-level tracing infrastructure designed for application transparency, despite the infrastructure's decent performance without such knowledge. In our prototype OpenStack's high-level semantics expose the fault-injection module and achieve high fault-injection efficiency by injecting faults to high-level communication flows but generic low-level events in runtime systems or operating systems. It is extremely difficult and costly to thoroughly investigate every aspect of the cloud-management stacks fault resilience. It is focused on common cases, injecting common faults during the processing of OpenStack's most commonly used external APIs. These faults are based on existing knowledge related in works [6, 7], and experience with large-scale production-level cloud systems. The APIs selection is based on experience with OpenStack several experimental deployments.

To simplify adoption of our framework, it is used building blocks that cloud-management stack developers are aware with. The choice between the high-level semantics exposure and inference is an exemplification of this principle, because developers have built logging and notification mechanisms exposing such information. Another example is that it is used hybrid approach to implement OpenStack specifications, combining imperative checking via generic Python scripts with declarative checking via the SQLAlchemy library, both of which are widely employed by developers of OpenStack.

Our fault-injection framework consists of a logging and coordination module, a fault injection module, and a specification-checking module. The logging and coordination module is responsible for logging communications among services during external request processing and coordinating the execution of OpenStack and a fault-injection controller. The fault injection module is conceptually composed of a fault injection controller running at a test server node and fault-injection stubs running with OpenStack. The fault injection controller manufactures information collected by the logging and coordination module, makes fault injection decisions, and demands fault-injection stubs to inject faults into OpenStack. The specification-checking module verifies whether the internal states and the externally visible behaviours (HTTP status code returned to external user) of OpenStack at the end of each fault-injection experiment comply with predefined specifications. Fig. 1 presents a system overview and a high-level workflow, latter discussed in next.

Table 1 – Communication log format

Attribute	Value/Explanation
Identifier	Unique for each communication
Tag	Unique for each external request
Timestamp	Physical time
Entity	OpenStack/external service name
Type	RPC cast/call/reply, REST request/response, database/hypervisor/ shell operation
Direction	Send/receive, call/return

The workflow consists fault-free execution, fault injection, and specification checking. For given external request, it starts with fault-free OpenStack execution resulting in successful request processing. The logs produced during the fault-free execution are fed to a parser to generate an execution graph (next detailed), describing communications among services. Combining the execution graph and a predefined fault specification, the framework generates a test plans collection, each consisting of a fault type from the fault specification and a fault-injection location in the execution graph. Fault-injection experiments are then conducted via logging and coordination module and the fault-injection module collaboration, with each experiment corresponding to a test plan. The results of experiment are checked against predefined state and behaviour specifications. It is manually identified experiment bugs causing specification violations.

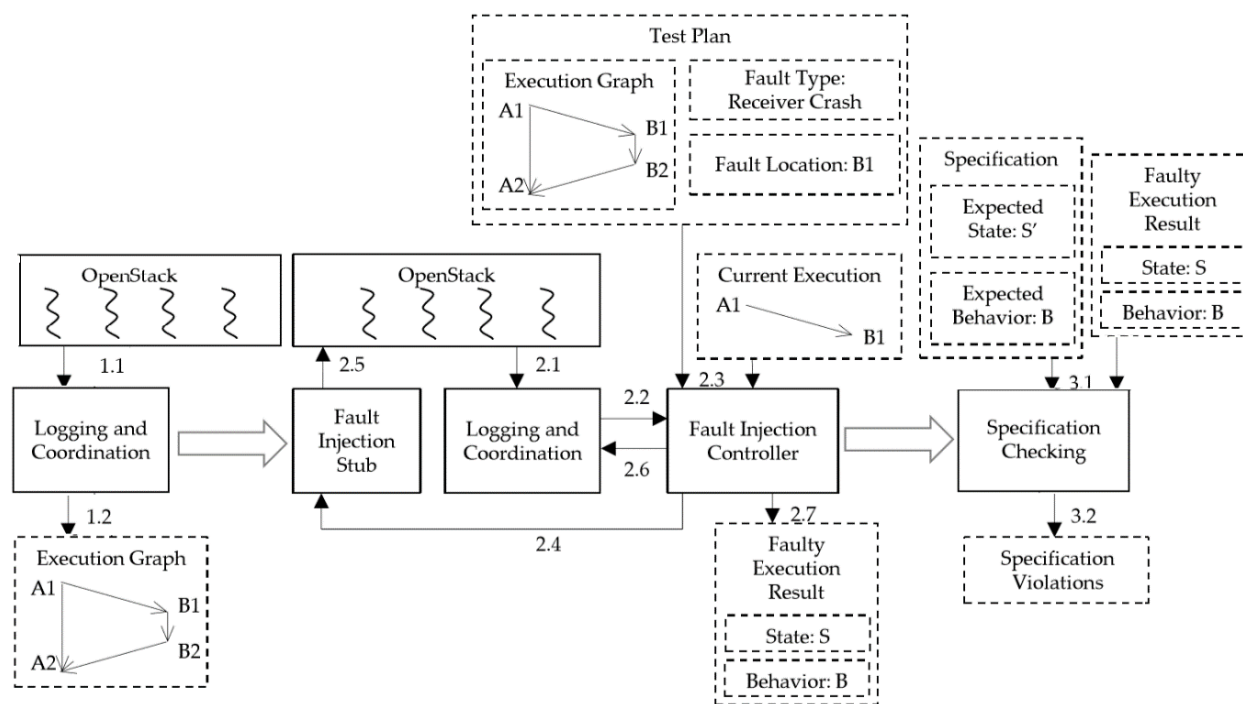


Fig. 1 – System overview and workflow

Logging and Coordination

After the framework overview we start an in depth discussion of its major components with the logging and coordination module. Following the domain specific information design principle exposing to the fault-injection controller, logging and coordination module openly maintains high-level several communications types' semantics in its logs, including RPC, REST, database, hypervisor and shell operations. Key attributes in communication log are enumerated in Table 1. A unique tag is created when

OpenStack receives an external request. The tag is then broadcast through OpenStack services along the request processing path. Recent OpenStack versions employ similar techniques for tracing the request processing within service groups. In contrast, the framework assigns a system-wide unique tag to each external request and traces its processing within the entire stack scope. Unique tags facilitate the log entries extraction related to a given external request. Otherwise, concurrent request processing would cause OpenStack to generate intertwined log entries and increase complexity of the log analysis. Although study currently boards fault injection during the single external request processing, the unique tag is still useful in that. It distinguishes the logs related to request processing from those generated by background tasks, such as periodic service liveness updates.

Boxes in fig. 1 with a solid border represent framework OpenStack and major components. Boxes with a dashed border represent key non-executable objects in the framework. Three stages: fault-free execution, fault injection and specification checking are separated by arrows: step «log OpenStack communications» in a fault-free execution, step «convert logs» to an execution graph, step «log communications» in a fault-injection experiment and pause communicating entities during logging, step «send logs» to fault-injection controller, step «make fault-injection decisions» according to a test plan, step «inform fault-injection stub» of the fault-injection decisions, step «inject faults», step «resume execution», step «collect results» from fault-injection experiments, step «check results» against specifications, step «report specification violations».

System-wide tag broadcast requires modifications to the communication mechanisms in OpenStack. Specifically, it is inserted a new field representing unique tags in both request contexts used

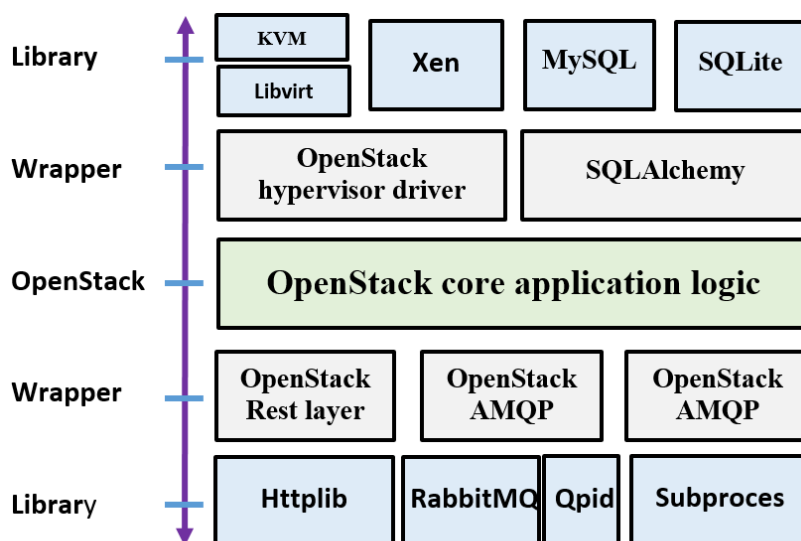


Fig. 2 – Illustration of logging instrumentation

by OpenStack services and thread-local storage for those services. When a green service thread is activated during the request processing, it updates the tag value in its thread-local storage with either the tag in the activating request if such a tag exists, or a newly initialized one. The thread associates this tag to all inter-service communications during its current activation. Framework cannot trace a unique tag once it broadcasts across the OpenStack boundary to external services. Consequently, if an OpenStack service communicates with an external service, which in turn communicates with another

OpenStack service, then our framework will treat the second communication independent from the first one. So far, it is not encountered such cases in this study, and the logging mechanism suffices for the use.

Implementation Pattern

There is the logging module implement by supplementing the communication layers between OpenStack and external services and libraries. On the whole, this module can be implemented at several layers along with communication paths: inside the logic of the OpenStack's core application where high-level communications are initiated, at OpenStack's wrapper layer of communication libraries, in communication libraries themselves, and in system libraries and interfaces. The logging arrangement resides at OpenStack's communication libraries wrapper layer as shown in Fig. 2. The advantages of logging at this layer are two-fold. First, semantics of the high-level can be precisely exposed instead of being incidental at this layer. Second, logging at this layer suffers minimum implementation effort, because it combines communications invented from and destined for OpenStack services. From one side, this layer is shared across OpenStack services and can use the same implementation to log communications from different services. On the other side, this layer is implemented at the communication category granularity (one implementation for all AMQP client libraries) and can use the

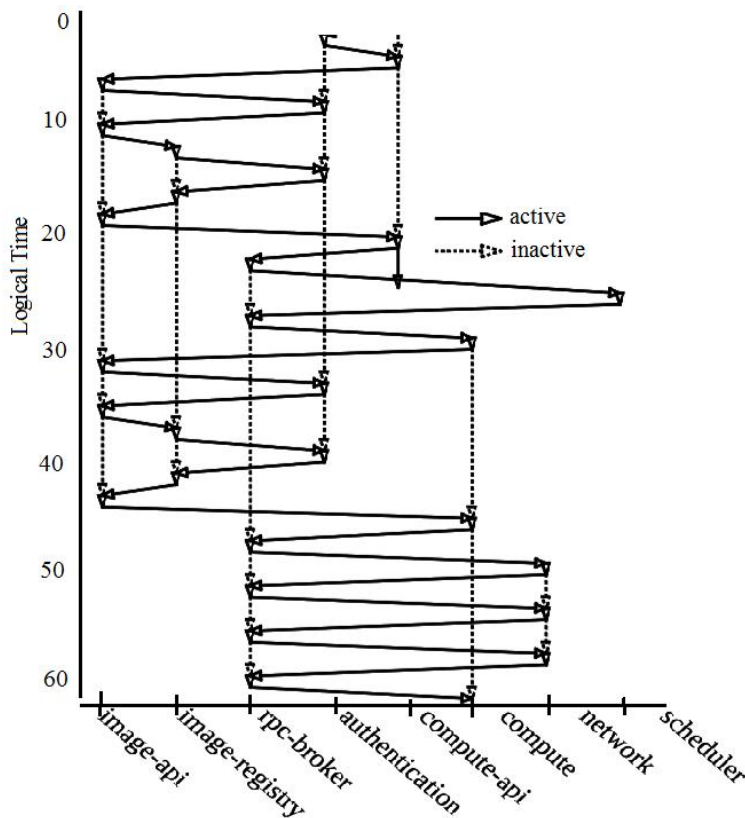


Fig. 3 - Execution graph illustration for VM creation, consisting of REST and RPC communication paths. Solid arrows represent active executions and dashed one represent idleness or waiting for the function call return

One drawback of this integrated user-level logging implementation is logging incompleteness. Compared to a system-level logging approach targeting a language level interface (Java SDK) or an operating system interface, the approach is incomplete in that it can only cover major communication mechanisms and is unaware to other channels (a customized socket communication). It need to submit that in a well-designed cloud management stack, the majority of inter-service communications are conducted via several well-defined interfaces, which were instrumented in the study on OpenStack. Additionally, system-level approaches usually lead to a significantly larger number of logs, humiliating system performance and necessitating the use of advanced log parsing and inference logic in the fault injection module. In the framework, there are traded logging completeness for simplicity in exposing high-level semantics and potentially high logging performance.

RPC communications within the compute service group is realised in part by modifying OpenStack's AMQP implementation. Such modifications cannot form a complete RPC communications picture, because the RPC caller and called (or producer and consumer in AMQP terminology) is decoupled by an AMQP broker. RPC cast from an OpenStack compute service is sent to AMQP exchange at the broker and then routed to a message queue. Another compute service subscribing to the message queue then receives the RPC, thus effecting the RPC cast. RPC calls are similar except that the return value goes through the AMQP broker as well.

For fine-grained fault injection control, it is intended to differentiate two stages of RPC message propagation – the first from the RPC caller to the AMQP broker and the second from the AMQP broker to the RPC callee. A direct solution would be to extend our logging module to the AMQP broker implementation (either Rabbit MQ or Qpid). This solution requires a general-purpose AMQP broker to include OpenStack-specific programming logic. Moreover, retrieving unique tags from RPC messages at an AMQP broker implies the abstraction layers elevation from the message transferring protocol

same implementation to log communications from supporting services in each category, conceptualizing away details related to individual services.

Whiteboxes with a solid border represent the instrumented layers for exposing high-level communications semantics between OpenStack core logic and supporting services and libraries. Logging snippets are placed in the WSGI implementation of OpenStack and the event let library, as well as several OpenStack client-side libraries for logging REST communications, and in the AMQP interface of OpenStack for logging RPC communications. For logging communications between OpenStack and hypervisors, we implement a logging driver acquiescent with OpenStack's hypervisor interface and use it to wrap the real drivers OpenStack selects to communicate with hypervisors. Communications between OpenStack and hypervisors are thus intercepted and recorded by the logging

(detailing packet formats) to a high-level RPC message with application semantics incurring significant implementation overhead.

The implementation leaves the AMQP broker intact and instead logs its activity via RPC trampolines – compute service proxies responsible for RPC forwarding. The created trampoline for each compute service is modifying OpenStack's client-side AMQP implementation so that RPCs addressed to a service are delivered instead to its trampoline. The trampoline records those RPCs and forwards them to the original destination. From the execution flow logging perspective, RPC trampolines represent the AMQP broker, thus completing the RPC communications picture.

Generating detailed logs with high-level semantics, the logging scraps also serve as coordination points, synchronizing the OpenStack execution and fault-injection servers. During fault-injection experiments, the logging module sends log messages to a fault-injection server and then blocks the logged OpenStack service. The server makes decisions of fault-injection, injects faults once necessary, and resumes the logged service execution by replying a «continue execution» message to the logging module. The logging module use for coordination is also one major difference between our implementation and the existing notification mechanisms in OpenStack.

Fault Injection

The fault-injection module is responsible for extracting execution graphs from logs, generating test plans, and injecting faults to OpenStack. An execution graph depicts the OpenStack execution during the external request processing. It is directed acyclic graph extracted from logs of fault free request processing procedure with each vertex representing in OpenStack a communication event. Each event is characterized by the communicating entity (an image-API service) and the communication type (REST request send operation). Edges represent causality among events. An edge connects two vertices: 1) if they form a sender-receiver pair or 2) if they belong to the same service and one precedes other. Fig. 3 shows a simplified execution graph related to a VM-creation request.

A test plan consists of three elements: an execution graph, a fault-injection location, and a fault type. Two types of faults are studied: servercrash6 and network partition. These fault types are common failure causes in cloud environments and are well-studied in the literature. Other fault types, such as invalid inputs [15] and performance degradation [16] are not considered here. Correlated faults are also common in real-world deployments but are not within the scope of this work, due to the limitation imposed by our current single-fault injection implementation.

Procedure1 Test Plan Generation

```
test_plans ← an empty list
for all node in exe graph do
  for all fault in fault specs do
    if fault can be injected to node then
      new_plan ← Test Plan (exe graph, node, fault)
      test_plans.append(new plan)
return test_plans
```

Procedure1 demonstrates the test plans generation. Iterating over an execution graph, the algorithm accounts for all fault types applicable to each vertex (a sender server crash targeting REST communications can only be inserted to the vertices performing REST request or response send operations) and makes accordingly test plans. This procedure provides an opportunity for global testing optimization: global because the fault-injection module has a view of the entire execution flow. For example, execution-graph vertices can be clustered by customized criteria, each cluster assigned with a testing priority. Vertices can then be selectively tested within each cluster to reduce overall testing cost. Given that a fault-injection experiment in the framework takes several minutes to complete and that an exhaustive set of test plans for one external request usually leads to hundreds of experiments, such a global optimization opportunity provided by an execution graph is valuable and worth further exploration. For test plan generation, a fault specification is used to define the faults types to be injected and the types of communications in which faults are injected. AS a test-case filter, the fault specification functions enabling the design of experiments set focusing only on a specific fault type (sender-server-crashes) and/or a specific communication type (REST communications). The specification format can be extended

to support other filters, such as confining fault injection to OpenStack services subset. A test plan is fulfilled via the test server cooperation and the logging and coordination module. The test server initializes the execution environment and then re-executes the external request to which the test plan resembles. Then, the test server employs the same log parsing logic for execution graph generation to analyse each log sent by the logging and coordination module. It tracks OpenStack's execution by using the execution graph in the test plan until the fault-injection location has been reached. A fault is then injected as plan specified. And OpenStack runs till the request processing is completed.

Faults of the server-crash are injected by killing relevant service processes via system. Configurations of system are modified such that when it stops the relevant services, a signal SIGKILL is sent, instead of the default signal SIGTERM. Forcing them to drop packets from each other, network-partition faults are injected by inserting IP-tables rules to service hosts that should be network-partitioned.

Specification Checking

The specification-checking module is responsible for verifying whether the results collected from OpenStack executions with injected faults comply with expectations on the states and OpenStack behaviors. Writing specifications for a large-scale complex distributed system is notoriously difficult, due to the numerous interactions and implicit inter-dependencies among various services and their execution environments. It is a key task for developing an effective specification checking module. In effect, the coverage and the states and behaviors granularity in the specifications determine the checking module ability to detect erroneous behaviors and target system states. A few approaches have been reported in the literature, including relying on developers to generate specifications [17], reusing system design specifications [6], and employing statistical methods [18]. To the best of our knowledge, OpenStack does not provide detailed and comprehensive specifications on system behaviors or state transitions during the external requests processing. The specifications that are used in this study are generated based on OpenStack understanding, existing knowledge in fault-resilient system design, and first principles, which mirrors the developer specification-generation approach. Specifically, specifications generated manually by inferring OpenStack developers' expectations on system states and behaviors. This process requires extensive reverse-engineering efforts, such as source-code reading and log analysis. Specifications generated in such a manner may require further debugging and refinements (similar to fixing incorrect expectations in [17]). Such specifications are best-efforts, with a coverage constrained by OpenStack understanding. Such specifications usefulness also is demonstrated by the bugs identification reported in this paper.

Specification 1 VM State Stabilization Specification

```
query = select VM from compute database
where VM.state in collection(VM unstable states)
if query.count() = 0 then
    return Pass
return Fail
```

Specification-Generation Guidelines listed below are usable specification-generation guidelines. "Do not block external users" announce that OpenStack shouldn't block external users due to faults during request processing. "Present clear error messages via well-defined interfaces" says that OpenStack should expose clear error states to external users via well-defined interfaces and avoid confusing information. "Stabilize system states eventually" informs that upon restoration of faulty services and with the quiescence of externally triggered activities, OpenStack should eventually stabilize inconsistent states caused by faults during request processing time.

Specification checking can be implemented via a general-purpose programming language or a specially designed specification language, using an imperative approach or a declarative approach. Following the using developer-familiar building blocks principle, a hybrid approach is adopted, applying declarative checking on persistent states stored in OpenStack's databases and imperative checking on the other states and OpenStack behaviors. Database-related checks are implemented via the SQL Alchemy library and others are implemented as generic scripts. This hybrid approach largely imitates OpenStack's existing implementation: Open Stack adopts the same approaches to controlling its states and behaviors.

Specifications implemented for OpenStack's states stored in databases and local filesystems and OpenStack's behaviors, such as the Http status code returned to an external user after processing a request. The specified expected states of cloud platforms are managed by OpenStack, such as the local hypervisors states on compute hosts and the Ethernet bridge configurations.

Specification 2 Ethernet Configuration Specification

```
if(VM.state = ACTIVE) and  
  ((VM.host.Ethernet not setup)  
   or (network controller.Ethernet not setup)) then  
  return Fail  
return Pass
```

Specification 3 Image Local Store Specification

```
query = select image from image database  
  where image.location is local  
if local image store.images = query.all() then  
  return Pass  
return Fail
```

Below was present three specification examples. Specification 1 indicates the VM state stabilization expectation. The VM state after the external request processing (VM creation) and a sufficient quiescence period should enter as table state (the ACTIVE state, indicating the VM running activity) instead of remaining in a transient state (the BUILD state, indicating the VM creation process). This is a using example declarative checking on database states.

Specification 2 requires VM actively running, the Ethernet bridges on the compute host where that VM resides and the host running the network controller service have been correctly set up. It is checked whether the bridges have been associated with the correct Ethernet interfaces dedicated to the subnet to which the VM belongs. It exemplifies the imperative checking on OpenStack-managed cloud platforms.

Specification 3 checks whether the database maintained states of OpenStack image service regarding the image store in the local filesystem are in accordance with the filesystem. It requires that if an image is uploaded to the local image store and thus exists in the filesystem of the image service host, the nits location attribute in the image database should be local, and vice versa. This specification shows a combined check on the database views and the service host filesystem.

Results

Here is discussing the bugs uncovered by our framework applied to OpenStack essex and grizzly versions, and find 23 bugs in total: 13 common to both versions, 9 unique to essex, and 1 unique to grizzly. The study covers three OpenStack service groups: the identity service (keystone), the image service (glance), and the compute service (nova). For external services, the framework supports the Qpid messaging service, the MySQL database service, the libvirt service for hypervisor interaction and the Apache Http server used as an image store for OpenStack.

The identity service is configured to use UUID tokens for authentication. Regarding the image service, it is configured to use a local filesystem or Http server as its backend store. As for the compute service, QEMU is used as the backend hypervisor, controlled by the libvirt interface. In essex, it is limited the reconnection from the Qpid client library to the backend broker equal 1.

OpenStack services run in VMs with 1 virtual CPU and 2GB memory. All OpenStack VMs run on an HP Blade System c7000 enclosure, each blade equipped with 2 AMD Opteron 2214HE (2.2GHz) processors and 8GB memory. For each fault-injection experiment, all services are deployed in one VM by default, each started as a single example. There are two exceptions to the above placement guideline. If a shell-operation fault should be injected to a compute service, then that service is placed in one VM and the other services are placed in another VM in order to prevent interference among similar shell operations of different services. If a network partition fault should be injected, then the pair of to be partitioned services are placed in two VMs and the other services are launched in a third VM.

There are tested 11 external OpenStack APIs, inject 3848 faults, implement 26 specifications, detect 1520 violations and identify 23 bugs. The results are summarized in Table 2. The table shows the bugs significance and issues discovered in our study, because they are manifested (I) in several frequently

used APIs and (II) in numerous locations along the execution paths of those APIs. The former observation is drawn from the fact that the bugs sum in the table exceeds by far the number of distinct bugs. The fact is that the ratio of specification violations to bugs is greater than 1 for each API, which is also shown in Fig. 4.

The bugs are classified into seven categories (Table 3) and present deep discussion of each category. Compared to our preliminary work [9], this article contains newly-identified bugs in OpenStack and also discusses the bugs evolution and fixes across the two OpenStack versions that is studying, demonstrating the OpenStack's fault resilience improvement and the outstanding issues.

Table 2: Summary of fault-injection results.

API	Faults				Specification Violations				Bugs			
	essex		grizzly		essex		grizzly		essex		grizzly	
	crash	part	crash	part	crash	part	crash	part	crash	part	crash	part
VM create	217	133	311	229	93	43	150	49	8	6	3	2
VM delete	79	61	102	82	51	15	45	23	9	5	5	2
VM pause	24	17	35	29	16	13	6	4	5	6	2	1
VM reboot	64	36	139	104	9	11	0	5	3	4	0	1
VM rebuild	159	106	242	183	103	67	0	13	5	5	0	1
VM image create (local)	142	119	171	150	59	106	90	79	4	2	3	3
VM image create (http)	107	92	171	150	24	84	79	71	3	2	3	3
VM image delete (local)	59	44	22	15	23	37	12	9	3	2	2	2
VM image delete (http)	59	44	22	15	23	37	10	8	2	2	1	1
Tenant create	7	6	7	6	0	6	0	6	0	1	0	1
User create	7	6	7	6	0	6	0	6	0	1	0	1
Role create	7	6	7	6	0	6	0	6	0	1	0	1
User-role create	9	8	10	9	0	8	0	9	0	1	0	1
Sum	940	678	1246	984	401	439	392	288	42	38	19	20

The results of image service APIs are broken down according to whether a local file system or an Http service is used as the image store. Specification violations related to an API are counted as the number of fault-injection experiments in which at least one violation is detected. The network-partition fault is shortened to "Part."

Table3: Bug Categories

Category	Count		
	common	essex only	grizzly only
Timeout	1	1	0
Periodic checking	6	4	0
State transition	1	0	1
Return code checking	4	1	0
Cross-layer coordination	0	1	0
Library interference	0	1	0
Miscellaneous	1	1	0
Total	13	9	1

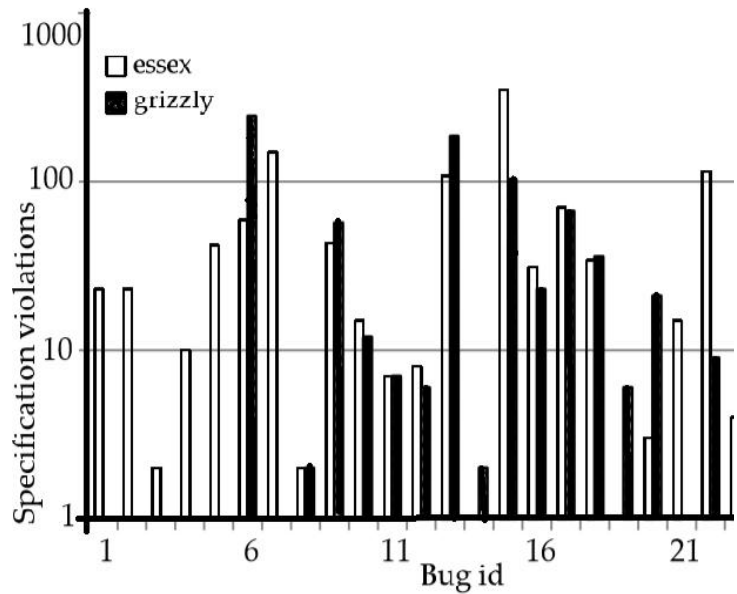


Fig. 4: Bug distribution. The Y-axis is the number of fault-injection experiments in which at least one specification violation leading to the given bug identification is detected

Timeout is a common mechanism in distributed systems to avoid one faulty service from indefinitely blocking other services and affecting system-wide functionality. OpenStack extensively uses the timeout mechanism with settings scattered across multiple service configurations to control various timeout behaviors of OpenStack services and external supporting services. Setting correct timeout values is known to be difficult. Given the numerous interactions and inter-dependencies among service groups and the deployment environments variety, it is very difficult if not impossible for OpenStack to provide a comprehensive set of timeout values covering all execution paths that can block the system.

For example, REST communications, one of the two major communication mechanisms used in OpenStack, fall out of the safety net of the timeout mechanism in the OpenStack essex version. Consequently, service is waiting for a response from another service via the REST mechanism may be indefinitely blocked if the two services become network partitioned after the request is sent but before the response is received. This also exemplifies the advantage of an execution-graph-based fault injection over coarser-grained approaches [19], such a bug can hardly be efficiently revealed by the latter due to the requirement on the synchronization between fault injections and send or receive operations.

This bug is fixed in the OpenStack grizzly version by supporting timeout values for REST communications and exposing such settings in its configurations. For instance, the image client library provides a default 600 seconds timeout. The identity client library, on the other hand, uses a default value of None, effectively disabling the timeout mechanism. While a system-wide default timeout value for REST communications might be a better solution, the fault OpenStack resilience has been clearly enhanced due to this critical timeout setting support.

Periodic checking is another critical mechanism for achieving fault resilience in OpenStack. In a cloud environment where faults are inevitable, periodic checking can monitor service liveness, resume interrupted executions, clean up garbage, and prevent resource leakage.

The fault-injection framework identifies several bugs caused by the lack of proper periodic checking mechanisms. Take the processing of a VM creation request as an example. During fault-free request processing, the VM state transits from None (non-existence) to BUILD (under construction) to ACTIVE (actively running). If the execution is interrupted after the VM has transited to BUILD, it is possible that, in essex, the VM indefinitely remains in that transient state. This is a bug because VM creation should cause the VM to enter a stable state (ACTIVE if the processing is successful but otherwise ERROR) in a timely manner, despite faults. This bug can be fixed by a periodic checking logic that converts VM's state from BUILD to ERROR if the related VM creation request is detected as failed. A bug fix has been integrated into grizzly, conducting such a transition based on a configurable VM

creation timeout value. This fix although effective in preventing the prolonged BUILD state, leads to a state transition bug.

Another bug in both OpenStack versions that can be fixed by periodic checking is related to database access in a fault-handling logic. OpenStack implements a decorator function wrap instance fault, injecting a fault record into a database table upon VM related faults detection. This fault-handling logic fails when the VM related fault that it intends to log is itself a database fault (database service crash fault during a VM creation). In such situation the fault record is discarded, making it difficult for post mortem fault analysis. This bug can be fixed by keeping a temporary log of the fault messages that cannot be stored in the designated database table at the time when they are generated. Then periodically checking the database service state and merging those temporary messages into the database when possible.

OpenStack maintains a large number of states in its databases, with complicated state-transition diagrams among them. A tool indicates that in faulty situations users may experience problematic state transitions. For example, grizzly employs a periodic task to convert a VM from BUILD to ERROR if it exceeds the maximum time that a VM is allowed to remain in BUILD. However, OpenStack does not cancel the VM creation request related to that VM. If a VM creation request experiences a transient fault, then a VM may transit from BUILD to ERROR and then to ACTIVE, because it can be created after the execution of the periodic task and the fault disappearance.

The state transition from ERROR to ACTIVE without external event triggering can be confusing and problematic. According to experience, upon VM creation error receipt an external user is likely to issue another VM creation request. The bug fix for the previous periodic-checking-related problem induces a new state-transition bug, potentially creating more VMs than needed for an external user. It suggests canceling an external request and negating its effect once it is believed to be erroneous in addition to the state stabilization employed by OpenStack.

Return code is commonly used as an indicator of the execution state from a function callee to its caller. Although thorough checking on return codes has long been established as a good programming practice, prior work has identified return code related bugs to be a major bug's source even in well-organized projects [20, 21]. Study on OpenStack confirms this observation. For example, during the processing of a VM creation request, when the identity service cannot authenticate a user-provided token passed from a compute API service due to an internal fault, it returns an error code to the compute API service correctly indicating the service fault. Due to a flawed return code checking logic the compute API service attributes such an error to the token invalidity generates a misleading error message accordingly and returns it to the external user.

Another example in this category is related to the execution of shell commands. OpenStack compute services implement a common function for executing shell commands allowing the caller to specify a list of expected return codes. If the return value falls out of that list, an exception is raised. A common bug pattern related to improper use of this function results from disabling its return code checking logic causing OpenStack when executing under faults, to deviate from expected execution flows without being detected. During the network setup procedure related to a VM creation, the *brctl addif* command is used to associate an Ethernet interface with a bridge on the compute host where the VM is placed. OpenStack assumes that the command can be fruitfully executed without checking its return code proceeds to start the VM. So, the VM may lose network connectivity if a fault occurs during execution of that command.

OpenStack relies on various supporting services to maintain its functionality and supports interaction with multiple services in each external service category via set of abstraction layers. Take the RPC messaging services as an example. OpenStack implements a unified layer for the AMQP protocol used for RPC communications, giving operations to a lower layer implemented for a specific AMQP broker. The lower layer is a client-side library wrapper provided by its corresponding broker such as RabbitMQ and Qpid. This client-side library comprises messaging implementation details and is responsible for the actual communication with the AMQP broker.

This valuable and well-designed multi-layer abstraction stack imposes stringent requirements on cross layer coordination. Incorrect interpretations of one layer behaviors may lead to subtle bugs in another layer. The Qpid client library is configured to automatically reconnect the AMQP broker after a connection disruption in *essex*. A threshold value is designed to control the maximum reconnections number. A connection maintained by the client library resides in a temporary erroneous state until the

threshold is reached at the time the connection enters a permanent erroneous state. The OpenStack client library wrapper does not coordinate properly with the client library regarding the temporary-permanent error state transition and causing the wrapper to vainly retry a connection that has been marked as irrecoverable by the client library.

The extensive external libraries use commonly found in large-scale software systems may lead to unexpected library interference. OpenStack uses a patched version of Python standard library functions to support cooperative thread scheduling. Subtle incompatibility in the patched functions, however, can engender bugs that are hard detected. Take the communication between OpenStack and Qpid broker again as an example. During a reconnection from the Qpid client library to a Qpid broker service the client internally uses a conventional consumer/producer synchronization via a select/read call pattern on a pipe. Due to incompatibility between the patched version of select and its counterpart in Python standard library, this Qpid client library, when invoked in essex, may perform a read on the pipe read-end that is not yet ready for reading, thus permanently blocking an entire compute service.

The framework also detects a simple implementation bug: in essex, when a fault disrupts a connection opening procedure in the Qpid wrapper layer, a subsequent open call is issued without first invoking close to clean up sour states in the connection object resulting from the previously failed open, causing all following retries to fail with an «already open» error message.

Comparing the results of the two versions, there are identified several interesting aspects in the OpenStack's fault resilience evolution. Using timeouts to limit the distributed operations execution is a well-known and important approach to fault-resilience improvement [22]. Use of timeout for REST communications effectively solves the indefinite sender blocking issue in essex. Systematically configuring timeouts also remains an open question. Different components in a REST communication flow (WSGI pipeline) have different default timeout values. The timeout settings of some important supporting services cannot be controlled by OpenStack. For example, OpenStack does not specify the timeout for SQL statement execution, thus causing long blocking time if the SQL statement issuing service and the database backend is network partitioned. These issues need to be properly addressed further improving the OpenStack fault resilience.

Carefully checking *return codes* enables prompt error detection. In grizzly, during the VM deletion request processing the compute service issues a RPC call instead of a RPC cast as in essex to the network service, demanding the latter to reclaim relevant network resources. This modification allows the compute service to detect errors in the network service and reacts accordingly (transiting the VM to the ERROR state), reducing the possibility of network resource leakage under faults.

By simplifying the *cross-layer coordination*, OpenStack reduces the bugs hiding chances between abstraction layers. By disabling automatic reconnection in the Qpid client library and reserving full control only in its wrapper layer OpenStack avoids the bug discussed before. Confining the decision-making logic regarding a specific aspect of cloud-management stack to a single layer, instead of coordinating the decision making in different layers, is considered a good design practice.

Discussions

The framework can be applied to solve fault-resilience issues related to cloud-management stacks in real-world deployments. One of Amazon's cascading failures [3] was caused by a memory leakage bug on storage servers, which was triggered by the data-collection server crash, the subsequent inability to connect to that failed server from storage servers, and the deficient memory reclamation for failed connections. This bug can be detected by the framework via the crash fault combination injected to the data-collection server and specification on the memory usage of the data reporting logic on storage servers. The framework needs to be extended for supporting multiple faults injection and scaling specification-checking logic from an individual request to multiple requests. Such improvements will enable the framework to handle complicated issues with multiple root causes [2].

One may consider the generality of this study in two aspects: the implementation reusability and findings applicability to other cloud-management stacks. As to implementation reusability, the fault-injection module and the specification-checking module of the framework are reusable in relevant studies for other cloud-management stacks. The logging and coordination module and the specifications used for checking the behaviors and states of OpenStack are domain-specific and require porting efforts. This also holds the study across two OpenStack versions. Such cross-version porting is straightforward. The logging and coordination module integrated in OpenStack contains some code lines, most of which are

reusable across the two versions. Specifications need to be moderately adjusted to handle with minor semantic (database schema changes) evolution.

Regarding the findings, the specific bugs and the related analysis presented in this work are OpenStack specific and cannot be sweeping to other cloud management stacks. The bug categories and the related fault-resilience issues are general. Despite the numerous differences in the detailed design and implementation of cloud-management stacks, many of them [23, 24] share common high-level scheme. They have similar service groups, rely on similar external supporting services, and employ similar communication mechanisms. The findings in this article have the potential to elucidate fault resilience in other cloud management stacks with a similar design.

The execution graphs and test plans use is optional. Instead of obtaining an execution graph related to request processing and generating test plans based on the graph before fault-injection experiments, an experiment could be started without prior knowledge and inject faults when a relevant communication event occurs.

Our choice of the execution graph use is mainly for future framework extensibility. On the one hand, an execution graph depicts the entire execution flow related to the external request processing and thus allows an intelligent test planner to conduct fault injection experiments with better test coverage and less resource consumption. On the other hand, execution graphs are useful for fault-resilience studies other than our current fault-injection framework, such as graph-comparison-based online fault detection [18].

In The framework is required that the external request processing during a fault-injection experiment match its execution graph up to the fault-injection location. The communication events observable by our framework in fault-free execution generally need to be deterministic. This requirement is satisfied in most of experiments. There are cases where additional care is required to accommodate nondeterminism. One source of nondeterminism is periodic tasks, which is resolved by deferring the tasks interfering with experiments. Another source is environmental inputs. The operations taken by a compute service for network setup on its local host depend on whether the compute service and network service are co-located on the same host. There are handled such cases by annotating related execution graphs so that all the paths observed in fault-free execution are marked as valid.

Related Work

Cloud-management stacks are type of distributed system. Our fault resilience study of this layer benefits from prior research on fault pliability of distributed systems in general and cloud systems. Here is compared the result of this work with existing fault resilience studies and execution path extraction and specification checking with similar techniques employed in distributed systems debugging and failure detection.

Fault injection is commonly used to study the cloud systems *fault resilience*. FATE is a fault-injection framework directing the recovery logic in cloud applications and exploring failure scenarios with multiple-failure injection [6]. In the work [5] crash faults were been injected to components on Hadoop's compute nodes and studied their effects on application performance. The framework targets cloud-management stacks and examines the recovery logic by conducting single-fault injection during the external requests processing. Similar to the target of FATE, it is studied the recovery logic functionality and correctness, which is difficult to be made correct.

Failure of Service (FaaS) is proposed as a new generic service to improve the cloud applications fault resilience in real deployments [25]. It is suggested that an alternative approach for cloud-management stacks, presenting an integrated fault-injection framework with domain knowledge. By combining them, cloud management stacks may enhance fault resilience by better balancing the cost and fault injection coverage.

Model checking in [7] is another common approach to examining the distributed systems fault resilience. Compared to our fault-injection-based approach, it checks the target system more thoroughly by exploring all possible execution paths instead of those observed by a fault-injection framework. This same thoroughness requires the extensive domain knowledge use in order to make it practical to check highly complicated operations in cloud-management stacks of the VM creation.

Execution path have been extensively used for workload modeling [26, 27], performance [28] and correctness debugging [18, 29], and evolution analysis [30] in distributed systems. Such information is exposed via special logging modules or inferred in sophisticated post-processing. Applying existing

knowledge to the framework, execution paths is extracted relating to external request processing via user-level logging, which explicitly exposes high-level semantics.

Prior research has explored various approaches to *specification checking* in distributed systems. Regarding specification expression, imperative approaches [31, 32] and declarative approaches [6, 17] have been studied. In our framework, a hybrid approach is employed in expressing specifications on the states and behaviors of cloud-management stacks, combining imperative and declarative checking. Similar combinations have been used to query and analyze distributed trace events [33]. Regarding specification generation, our and most existing approaches require developers to implement specifications. Recent advances in filesystem-checker testing leverage the characteristics in the checkers to automatically generate implicit specifications [34]. The applicability of similar approaches to cloud-management stacks remains an open question for today.

10 Conclusions

In this article, a systematic study is conducted on the fault resilience of OpenStack. It is designed and implemented a prototype fault-injection framework that injects faults during the external requests processing. Using this framework, it is uncovered 23 bugs in two OpenStack versions, classified them into seven categories, and presented deep discussion of the fault resilience issues, which must be addressed in order to build fault-resilient cloud-management stacks. In future for the research can be taken refining and automating specification generation logic in fault-resilience areas and exploring potential use of execution graphs.

1. Amazon. Amazon elastic compute cloud (Amazon EC2). <http://aws.amazon.com/ec2/>. Retrieved in September 2013.
2. Amazon. Summary of the Amazon EC2 and Amazon RDS service disruption in the US east region. <http://aws.amazon.com/message/65648/>. Retrieved in September 2013.
3. Amazon. Summary of the October 22, 2012 AWS service event in the US-east region. <http://aws.amazon.com/message/680342/>. Retrieved in September 2013.
4. Microsoft. Summary of Windows Azure service disruption on Feb 29th, 2012. <http://blogs.msdn.com/b/windowsazure/archive/2012/03/09/summary-of-windows-azure-service-disruption-on-feb-29th-2012.aspx>. Retr. in September 2013.
5. F. Dinu and T. E. Ng. Understanding the effects and implications of compute node related failures in Hadoop. In Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing, HPDC '12, pages 187–198, New York, NY, USA, 2012. ACM.
6. H. S. Gunawi, T. Do, P. Joshi, P. Alvaro, J. M. Hellerstein, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, K. Sen, and D. Borthakur. FATE and DESTINI: a framework for cloud recovery testing. In Proceedings of the 8th USENIX conference on Networked systems design and implementation, NSDI'11, Berkeley, CA, USA, 2011. USENIX Association.
7. J. Yang, T. Chen, M. Wu, Z. Xu, X. Liu, H. Lin, M. Yang, F. Long, L. Zhang, L. Zhou. Modist: transparent model checking of unmodified distributed systems. In Proceedings of the 6th USENIX symposium on Networked systems design and implementation, NSDI'09, pages 213–228, Berkeley, CA, USA, 2009. USENIX Association.
- A. Reddy. DevOps: The IBM approach. Technical report, IBM, 2013.
8. X. Ju, L. Soares, K. G. Shin, and K. D. Ryu. Towards a fault-resilient cloud management stack. In USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'13. USENIX Association, 2013.
9. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In Proceedings of the nineteenth ACM symposium on Operating systems principles, SOSP '03, pages 164 – 177, New York, NY, USA, 2003. ACM.
- A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the Linux virtual machine monitor. In Ottawa Linux Symposium, pages 225 – 230, 2007.
10. Microsoft. Microsoft Hyper-V server 2012. <http://www.microsoft.com/enus/server-cloud/hyper-v-server/default.aspx>. Retrieved in September 2013.
11. OpenStack. OpenStack open source cloud computing software. <http://www.openstack.org/>. Retrieved in September 2013.
12. B. H. Sigelman, L. A. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, C. Shanbhag. Dapper, a large-scale distributed systems tracing infrastructure. Technical report, Google, Inc., 2010.
13. B.P. Miller, L. Fredriksen, B. So. An empirical study of the reliability of UNIX utilities. Commun. ACM, 33(12):32 – 44, Dec. 1990.
14. T. Do, M. Hao, T. Leesatapornwongsa, T. Patanaanake, and H. S. Gunawi. Limplock: Understanding the impact of limpware on scale-out cloud systems. In 2013 ACM Symposium on Cloud Computing, SOCC'13, New York, NY, USA, 2013. ACM.
15. P. Reynolds, C. Killian, J. L. Wiener, J. C. Mogul, M. A. Shah, and A. Vahdat. Pip: detecting the unexpected in distributed systems. In Proceedings of the 3rd conference on Networked Systems Design and Implementation - Volume 3, NSDI'06, Berkeley, CA, USA, 2006. USENIX Association.
16. M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer. Path-based failure and evolution management. In Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation – Vol. 1, NSDI'04, Berkeley, CA, USA, 2004. USENIX Association.

17. C. Bennett and A. Tseitlin. Chaos monkey released into the wild. <http://techblog.netflix.com/2012/07/chaos-monkeyreleased-into-wild.html>. Retrieved in September 2013.
18. H. S. Gunawi, C. Rubio-González, A. C. ArpaciDusseau, R. H. Arpaci-Dusseau, and B. Liblit. Eio: error handling is occasionally correct. In Proceedings of the 6th USENIX Conference on File and Storage Technologies, FAST'08, pages 207–222, Berkeley, CA, USA, 2008. USENIX Association.
19. P. D. Marinescu, G. Candea. Efficient testing of recovery code using fault injection. *ACM Trans. Comp. Syst.*, 29(4):11:1 – 11:38, Dec. 2011.
20. J. B. Leners, H. Wu, W.-L. Hung, M. K. Aguilera, M. Walfish. Detecting failures in distributed systems with the falcon spy network. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11, pages 279 – 294, New York, NY, USA, 2011. ACM.
21. CloudStack. Apache CloudStack: Open source cloud computing. <http://cloudstack.apache.org/>. Retrieved in September 2013.
22. Eucalyptus. The Eucalyptus cloud. <http://www.eucalyptus.com/eucalyptuscloud/iaas>. Retrieved in September 2013.
23. H. S. Gunawi, T. Do, J. M. Hellerstein, I. Stoica, D. Borthakur, and J. Robbins. Failure as a service (Faas): A cloud service for large-scale, online failure drills. In Technical Report UCB/EECS-201187.
24. P. Barham, A. Donnelly, R. Isaacs, and R. Mortier. Using magpie for request extraction and workload modelling. In Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6, OSDI'04, pages 18–18, Berkeley, CA, USA, 2004. USENIX Association.
25. B. C. Tak, C. Tang, C. Zhang, S. Govindan, B. Ugaonkar, R.N.Chang. vpath: precise discovery of request processing paths from black-box observations of thread and network activities. In Proceedings of the 2009 conference on USENIX Annual technical conference, USENIX'09, pages 19– 19, Berkeley, CA, USA, 2009. USENIX Association.
26. P. Reynolds, J. L. Wiener, J. C. Mogul, M. K. Aguilera, A. Vahdat. WAP5: black-box performance debugging for wide-area systems. In Proceedings of the 15th international conference on WorldWideWeb'06, pp. 347 –356, New York, NY, USA, 2006. ACM.
27. D. Geels, G. Altekar, S. Shenker and I. Stoica. Replay debugging for distributed applications. In Proceedings of the annual conference on USENIX '06 Annual Technical Conference, ATEC '06, Berkeley, CA, USA, 2006. USENIX Association.
28. S.A. Baset, C. Tang, B.C. Tak, and L. Wang. Dissecting open source cloud evolution: An open stack case study. In USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'13. USENIX Association, 2013.
29. X. Liu, Z. Guo, X. Wang, F. Chen, X. Lian, J. Tang, M. Wu, M. F. Kaashoek, and Z. Zhang. D3S: debugging deployed distributed systems. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI'08, pages 423–437, Berkeley, CA, USA, 2008. USENIX Association.
30. X. Liu, W. Lin, A. Pan, and Z. Zhang. Wids checker: combating bugs in distributed systems. In Proceedings of the 4th USENIX conference on Networked systems design and implementation, NSDI'07, pp. 19 –19, Berkeley, CA, USA, 2007. USENIX Association.
31. U. Erlingsson, M. Peinado, S. Peter, M. Budiu, and G. Mainar-Ruiz. Fay: Extensible distributed tracing from kernels to clusters. *ACM Trans. Comp. Syst.*, 30(4):13:1–13:35, Nov. 2012.
32. J. Carreira, R. Rodrigues, G. Candea, and R. Majumdar. Scalable testing of file system checkers. In Proceedings of the 7th ACM European conference on Computer Systems, EuroSys '12, pages 239– 252, New York, NY, USA, 2012. ACM.

УДК 004

Maksymovych O.V., Melnyk K.V., Melnyk V.M., Gaday O.S.
Lutsk national technical University

AUDIO PLAYER WITH A LIBRARY OF DELPHI

Максимович О.В., Мельник К.В., Мельник В.М., Гадай О.С. Аудіоплеєр з фонотекою засобами мови Delphi. У статті представлена розробка аудіоплеєра для персональних комп'ютерів, що дозволяє: відтворювати аудіо файли в зручному для користувача вигляді; каталогізувати всі аудіо файли і забезпечити доступ до них в одному місці, потоково відтворювати аудіо записи з соціальної мережі VK.

Ключові слова: Аудіоплеєра, аудіо файли, мова Delphi.

Климюк М., Мельник К.В., Мельник В.М., Гадай О.С. Аудиоплеер с фонотекой средствами языка Delphi. В статье представлена разработка аудиоплеера для персональных компьютеров, что позволяет: воспроизводить аудио файлы в удобном для пользователя виде; каталогизировать все аудио файлы и обеспечить доступ к ним в одном месте, потоково воспроизводить аудио записи из социальной сети VK.

Ключевые слова: аудиоплеера, аудио файлы, язык Delphi.

Maksymovych O.V., Melnyk K.V., Melnyk V.M., Gaday O.S. Audio player with a library of Delphi. The article presents the development of the audio player for PC that allows you to play audio files in user-friendly form; all audio files have to be gathered in catalog and access one place, to play audio thread from VK social network.

Keywords: Audio player, audio files, Delphi

The scientific problem description. Regarding to availability of portable devices presence or absence for audio recordings reproducing, the main one to play audio recordings is the computer. Listen to audio recordings is an integral lively part of the most people. An important aspect for them contains the number of users increasing to the global Internet, so there is also a real need in the software that interacts with a global network. The number of this type software is large enough, and the time, when the user had to be able to play single audio recordings on the hard drive, is really gone off. To realize that, the users today have to get some minimum list of options, good and simple interface for using etc. A large list of audio players raises before users the problem how to choose the best option.

Particularly, the famous audio player Winamp offers the fast scaling of the program window size, the track (song) automatic recognition, the tags filling, album artwork finding, the library import from iTunes, the plugins extensions support, that add various functionality: the reading of lossless formats, the work with the disk appearances, etc. One of the disadvantages in user program using takes the tangled interface [1].

For example, an audio player named AIMP, is a modified Winamp player alternative, has an advanced built-in equalizer (18 sound versus frequencies opposite to 10 one in Winamp) with the set of different audio effects, the possibility of simultaneous work with several playlists (now playing one and second is editing), the built-in tag editor and the ability to download skins.

AIMP natively supports lossless appearances in availability of the markup file CUE. It has the possibility to use some different ways of outputting the signal for better sound like DirectSound, ASIO and WASAPI. AIMP also holds the listening to radio stations from Internet. The drawback is the small navigation controls in this player, the media library opens in a separate window, and the tags are not loading automatically.

One of the multifunctional audio player is aTunes, which has its own multiplatform ability, customizable interface, automatic album covers download, but the engine in Java is quite resource-volume and the list of the radio stations has to be added manually. Audio player foobar2000 gives to user himself the ability to customize the interface. It owns a small program size, easy interface, great set of plugins that extend the player functionality. As the shortcoming can be included uncomfortable graphical interface in the basic Assembly and no automated plug-in settings. A media player named Songbird is built with the Mozilla engine Firefox is cross-platformer and it can be used as a full-fledged browser on a separate tab (password Manager, protector from pop-up Windows). The music information import about composition is performing from iTunes and automatically watches the tracks changes in the specified folder and sync the music library.

The aim of the article. In this work we present the audio player development for PC that allows you to play audio files in user-friendly form; to catalog all the audio files and provide access to all of them in one place; to play threadlike audio from VK social network.

Result and discussions. The audio player which is called Art Player consists of two parts: the audio player and music library. All two parts located in the same window, with the ability to switch over between them. A program algorithm is depicted in the block diagram shown in fig.1.

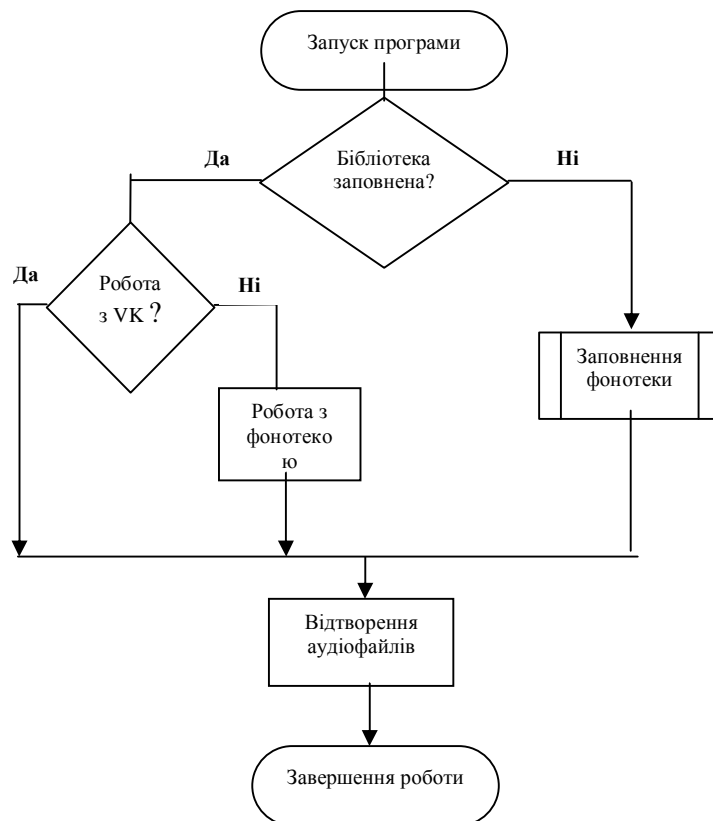


Fig.1 The Algorithm of the program. Block diagram

To display the window to the last user it need to carry out some actions such as: to initialize the device to play audio with plugins to work with different formats; create all the necessary lists for using during the program work; to create the class examples to provide the tags computing of the audio data; completeness checking for the database. All the functions and procedures described below are built on the Delphi language.

Having tested for the database content and to reveal its empty, the program gives the choice for the user to start filling the database while connected to the Internet: VK or to finish the work.

To fill in the database it should have data that retrieved by the file search on user's hard drive, and meet our parameters (*.mp3 format, *.aac, *.m4a file size larger than 1 MB). Search of the audio files is carried out using the JvSearchFiles1, and when each file that meets our requirements is found, the procedure is called:

```
TMainView. JvSearchFiles1FindFile(Sender: TObject; const AName: string);
```

After all the files are found, the database is filling in agreement with three categories: song, albums, and artists that are filling three tables. While filling the database, it is used another two procedures: the procedure for obtaining tags from audio files GetTagFromTrack(source: string; cover:boolean) and the procedure to obtain the image GetArtistImage(const Artistname: string).

The first procedure is responsible for receiving and processing the audio file tags and it is calling 2 parameters: source – the file from which the data is retrieved, and cover – the switcher to work with the cover or not.

The album artwork processing is to bring all covers to a single extension, size, format, for the working ease with them on next, files playback and the library operating. The covers are stored on disk in subfolder "Data/Cache" of the program folder. The saving of the all covers in one place increases the time of filling the database, but can significantly speed up the program on the next.

It is also three categories created for the sample: songs, albums, artists. The file selecting is performing using SQL queries with timely sorting data to growth the first letter of the keyword. The data is placed in a visible list, named DBList, with which the user works, and also in few invisible lists, which are responsible for the covers and files.

The color scheme adaptation is based on the album composition cover of the song that is playing. For such adaptation are responsible next procedures: BlurArtwork, GetPixelColor, ApplyColors.

For playing files statistics, albums and artists, the ChangePlaycount procedure is created and the same for all categories. The part of the code for tracking playback of the compositions is shown below:

```
ADOQuery1.close;
ADOQuery1.SQL.Clear;
```



```
ADOQuery1.SQL.Add('SELECT Playcount FROM Tracks WHERE Tracks=' +  
  quotedstr(title1) + ' AND Albums=' + quotedstr(album1) + ');  
ADOQuery1.Active := true;  
plcnt := ADOQuery1.Fields[0].asInteger;  
plcnt := plcnt + 1;  
ADOQuery1.SQL.Clear;  
ADOQuery1.SQL.Add('UPDATE Tracks SET Playcount=' + IntToStr(plcnt) +  
  ' WHERE Tracks=' + quotedstr(title1) + 'AND Albums=' +  
  quotedstr(album1) + ');  
ADOQuery1.ExecSQL;
```

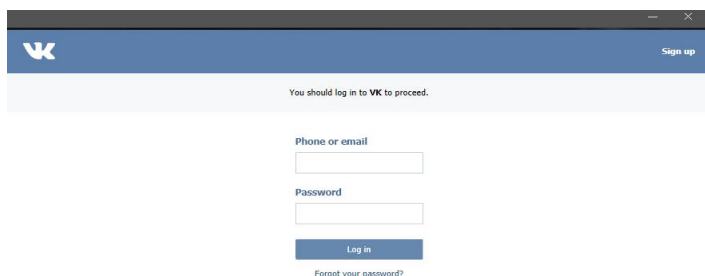


Fig. 2 The login Window VK

The procedure PlayerArtPlayFile is responsible for audio playback. The procedure receives the argument that is excepting the path to the file that has to be played. The program includes a free play option that allows you to diversify the play of the selected composition.

Internet services, which the developed product interacts with, there are more than one year. Therefore, they have already developed server-side

methods via convenient API. To be able to use the methods VK API it need to create create an application on the server, putting in the title, the list of the user data to which it will be accessed.

After creating the app its ID will be accepted, that has to be specified during an authorization procedure. The authorization also should be done through the browser, in this case through an instance of the system browser, embedded in the program. After the successful authorization is done in the ribbon, navigate line gets the token key, which grants access to the user data. Similarly to VK on the website Last.fm, there is also necessary to create your account and the app through which it will communicate with the API. After registration the api_key will be accessible, which will be used for interaction with the API.

Program testing

After the first time starting the program you are prompted to fill in the library, or to leave the program. The rate of the filling for the library depends on the disk size loaded with other programs at before time from the number of files that hold into the program filter in agreement with its size and extension.

After clicking on the corresponding icon the VK login window has the appearance shown in fig.2. The working case of this window together with VK is shown on fig.3.

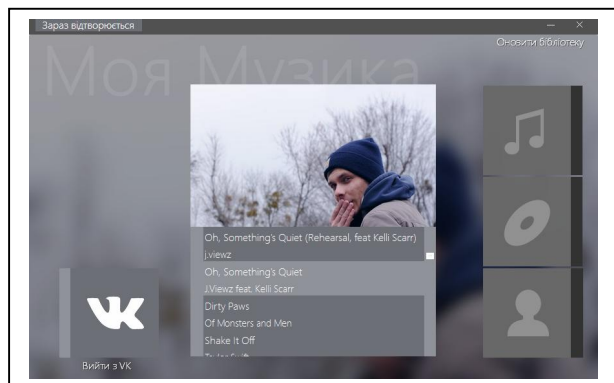


Fig. 3. The playlist when you work with VK

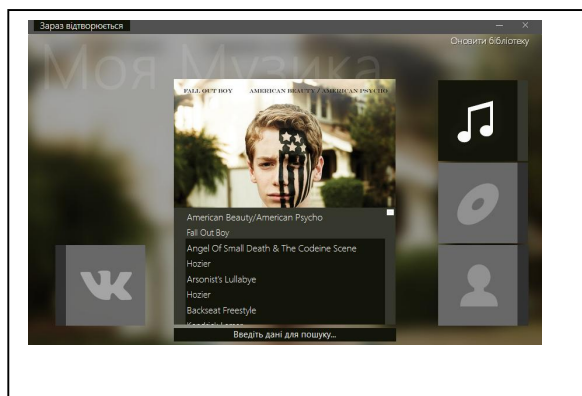


Fig.4. Song

The first icon displays a list of all songs, the second icon displays a list of all the albums in the music library, and the third list displays all the performers available in the music library (Fig.4).

Conclusions. The audio player constructed by Delphi means with Internet services support, which has such available features as audio playback MP3 format, AAC, simple and handy interface with customizing of the color scheme for application depended from the album composition cover and the song that's playing and the metadata supporting (tags) for files. It also holds the streaming audio play from VK social network with the ability to view compositions texts and the support of the music files ordering using the music library.

1. Лучшие аудио плееры для ПК | Обзор проигрывателей музыки на компьютере. ."- [Цит. 2015, 16 листопада].-Доступний з: < <http://mediapure.ru/multimedia/luchshie-audio-pleery-dlya-pk-obzor-proigryvatelej-muzyki-na-kompyutere>>.
2. "Статья - музыкальные audio форматы.".- [Цит. 2015, 10 листопада].-Доступний з: < <http://www.djmix.su/stat/index3.htm>>.
3. "Embarcadero RAD Studio XE".- [Цит. 2015, 10 листопада].-Доступний з: <<http://www.interface.ru/home.asp?artId=24428>>.
4. "BASS Audio Library".- [Цит. 2015, 10 листопада].-Доступний з: <<http://www.un4seen.com/bass.html>>.
5. "TMS Component Pack".- [Цит. 2015, 12 листопада].-Доступний з: <<http://www.tmssoftware.com/site/tmspack.asp>>.
6. "JCL: JEDI Code Library".- [Цит. 2015, 5 листопада].-Доступний з: <http://wiki.delphi-jedi.org/index.php?title=JEDI_Code_Library>.
7. "ImageEN Features".- [Цит. 2015, 5 листопада].-Доступний з: <<http://www.imageen.com/info/index.html>>.
8. Жулин И. "Бесплатные аудиоплееры.".-2010.- [Цит. 2015, 3 листопада].-Доступний з: <<http://www.bestfree.ru/review/soft/audioplayer.php>>.
9. Попов В.В. Программирование в Delphi. Оптимальный подход, Век, 2005.-150 с.
10. Фленов М.В. Программирование в Delphi глазами хакера + CD, BHV-Санкт-Петербург, 2004.-330 с.

УДК 004.54 (045)

Бортник К.Я.

Луцький національний технічний університет

СТРУКТУРА СИСТЕМИ БАГАТОШАРОВОГО ПЕРЦЕПТРОНУ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Бортник К.Я. Структура системи багатошарового перцептрону для мобільних пристроїв. Розглянуто структуру системи на чіпі (SoC) багатошарового перцептрону для мобільних пристроїв. Система придатна для пристроїв, які мають потребу у реалізації функції розпізнавання образів і які мають обмежені обчислювальні можливості. Проведено проектування системи, а також її верифікація на спеціалізованій макетній платформі. Для розроблення програмного забезпечення використано мову моделювання та синтезу програмного забезпечення VHDL. Доведено, що перенесення нейронних обчислень з центральних процесорів загального призначення на апаратні платформи на базі програмованих логічних інтегральних схем (FPGA) значно прискорює роботу алгоритмів розпізнавання.

Ключові слова: програмована логічна інтегральна схема (FPGA), система на чіпі (SoC), багатошаровий перцептрон (MLP), штучна нейронна мережа (ANN).

Бортник К.Я. Структура системы многослойного персептрона для мобильных устройств. Рассмотрена структура системы на чипе (SoC) многослойного персептрона для мобильных устройств. Система пригодна для устройств, которые нуждаются в реализации функции распознавания образов и имеющих ограниченные вычислительные возможности. Проведено проектирование системы, а также ее верификация на специализированной макетной платформе. Для разработки программного обеспечения использован язык моделирования и синтеза программного обеспечения VHDL. Доказано, что перенос нейронных вычислений от центральных процессоров общего назначения на аппаратные платформы на базе программируемых логических интегральных схем (FPGA) значительно ускоряет работу алгоритмов распознавания.

Ключевые слова: программируемая логическая интегральная схема (FPGA), система на чипе (SoC), многослойный персептрон (MLP), искусственная нейронная сеть (ANN).

Bortnyk Ka. Structure of the system based on Multi-Layer Perceptron for Smart Devices. The structure of the system on chip (SoC) multilayer perceptron for mobile devices was observed. The system is suitable for devices that need to implement the function of pattern recognition and which have limited computing power. A system design and its verification made on a specialized prototyping platform. For software development language used simulation and synthesis software suite VHDL. It is considered that the transfer of neural computation from the CPU of general purposes to hardware platform based on field-programmable gate array (FPGA) significantly speeds up recognition algorithms.

Keywords: field-programmable gate array (FPGA), system on chip (SoC), Multilayer Perceptron (MLP), artificial neural network (ANN).

Актуальність проблеми. Попит на «розумні» пристрої споживчої електроніки зростає. Це мотивується широким використанням недорогих вбудованих електронних пристроїв різноманітного призначення. Окрім того, бажано, щоб електронні пристрої мали здатність відчувати і розуміти їх оточення та адаптувати їхні сервіси відповідно до контексту.

Штучні нейронні мережі (ANN) можуть бути обрані для цієї мети у першу чергу, у зв'язку з їх широким спектром застосовності.

Багатошаровий перцептрон (MLP) є однією з найбільш часто використовуваних ANN через його здатність моделювати нелінійні системи і встановлювати межі нелінійних рішень в задачах класифікації, таких як оптичне розпізнавання символів (OCR), інтелектуальний аналіз даних і обробки/розпізнавання зображень.

Однак, оскільки MLP вимагає надзвичайно високої пропускну здатності, його обчислювальна складність вкрай небажана для виконання операцій в режимі реального часу, особливо для вбудованих пристроїв, які мають обмеження у своїх можливостях обробки даних. Привабливе вирішення цього полягає в розробці спеціалізованих апаратних засобів для прискорення MLP.

Апаратна реалізація MLP була нагальною темою протягом багатьох років, в основному у питаннях точності, необхідного простору і швидкості обробки. Різні реалізації обладнання для MLP були успішними, наприклад, метод проектування спеціалізованих інтегральних схем. Тим не менш, повна апаратна реалізація не є ефективною з точки зору вартості пристрою та складності його реалізації.

Останнім часом активно досліджується перенастроювана обчислювальна парадигма, в рамках якої розроблено ряд програмованих логічних інтегральних схем (FPGA). Але хоча кілька

апаратних реалізацій з використанням FPGA вже були запропоновані, апаратна реалізація в MLP, як і раніше, залишається складним завданням для вбудованих додатків.

Оскільки, різні методи преобробки і постобробки можуть бути суміщені з MLP в реальних додатках, система повинна налаштовуватись відповідно до додатків. Крім того, існує важлива вимога до проектування апаратного забезпечення таким чином, щоб внесення змін в структуру мережі не приводило до необхідності апаратної модернізації.

Ці проблеми можуть бути подолані за допомогою комбінованого програмного/апаратного методу проектування. Цей метод здійснюється шляхом аналізу різних частин алгоритму та реалізації його найбільш вимогливих ланок на швидких апаратних пристроях. Служити цьому може так звана система на чіпі (SoC), що складається з мікропроцесора та апаратних прискорювачів на базі FPGA, що може значно прискорити роботу, зберігаючи значною мірою гнучкість суто програмних моделей.

У цій статті описано архітектуру MLP-SoC смарт-додатків для вбудованих пристроїв. У SoC можна вносити зміни в структуру мережі і додатків без модифікації апаратних засобів.

Завдання з реалізації MLP-SoC. Нашою метою є розробка MLP-SoC, яка може бути використана для вбудованих додатків. У ході реалізації MLP-SoC на макетній платформі представлення даних, точність і апаратні компоненти відіграють важливу роль у виборі проектних рішень.

Макетна платформа. Для розробки та перевірки SoC найбільш доцільно використати основу на FPGA макетну платформу XILINX X2CV8000, яка стала популярною саме через можливість швидкого макетування і перевірки вбудованих додатків.

Макетування всього проекту цільової SoC в FPGA дає точне і швидке уявлення про майбутній пристрій.

Звичайно, деякі основні компоненти, включаючи центральний процесор, системи шин і внутрішніх комутованих блоків, вибираються для проектування реальної платформи на виробництві. Однак, для розробки і тестування алгоритмів додатку достатньо вбудованих у макетну платформу. Наприклад, мікропроцесор LEON 2 вбудований в FPGA, шина AHB/APB AMBA, призначена для зв'язку між внутрішніми компонентами, також вже вбудована в FPGA.

На додаток до мікросхеми FPGA, макетна платформа пропонує пам'ять на основі SDRAM чіпів розміром 128 МБ і флеш-пам'ять розміром 8 Мб. Блок пам'яті SDRAM використовується як оперативний запам'ятовувачий пристрій, у той час, як на базі флеш-пам'яті організовано ПЗП для зберігання програмного забезпечення. На рисунку 1 зображено макетну платформу, що має досить компактний розмір 112 на 129мм.



Рис. 1. Макетна платформа XILINX X2CV8000

MLP для обробки/розпізнавання зображень. MLP для обробки/розпізнавання зображення складається з вузлів обробки даних, розташованих шарами.

Як правило, він вимагає трьох або більше шарів вузлів обробки: вхідний шар, один або декілька внутрішніх шарів та вихідний прошарок. Кожен вузол обробки в одному конкретному шарі повністю або частково пов'язано з кожним вузлом шару, що розташований вище і нижче за

нього. Зважені з'єднання визначають поведінку в мережі і налаштовуються під час тренування мережі за допомогою спеціального алгоритму зворотного поширення.

Під час розпізнавання, вхідний вектор подається на вхідний шар. Для наступних шарів, вхід в кожен вузол є сумою скалярних добутків елементів вхідного вектора з відповідними їм ваговими коефіцієнтами:

$$sum_i = \sum_j w_{ij} \bullet out_j,$$

де w_{ij} - вага підключення вузла до вузла J , а out_j - вихід з вузла J .

Вихід вузла I , це $out_j = F(sum_i)$, який потім розсилається всім вузлам в наступному шарі. Ця операція повторюється для всіх шарів мережі доти, поки вихідний шар не буде досягнуто, де і обчислюється вихідний вектор. F – це функція активації кожного вузла. У якості функції активації найчастіше використовується сигмовидна функція або функція гіперболічного тангенса. У таблиці 1 показані апаратні обмеження для цільової SoC.

Таблиця 1.

Апаратні константи цільової SoC

Назва параметру	Значення
Максимальне число вхідних вузлів	1000
Максимальне число внутрішніх вузлів та шарів	128/2
Точність вагового коефіцієнта	12 біт (знаковий)
Точність виходу функції активації	9 біт (знаковий)
Діапазон вхідних даних	0 ... 255
Діапазон вихідних даних	-255 ... 255

Використання типу даних з плаваючою розрядністю (ваги, входи, виходи) в нейронній мережі може бути непрактичним для вбудованих апаратних засобів, тому варто використати тип даних з фіксованою розрядністю для значень ваг, входів і виходів.

Незнакові 8 бітні використовуються для представлення вхідних значень, у той час, як знакові 9 бітні використовуються для вихідних значень функції активації, бо вони можуть бути від'ємними. Коефіцієнти ваг зберігаються в таблиці ваг за допомогою знакових 12 бітних з фіксованою розрядністю.

Безпосередня апаратна реалізація певної функції активації не відповідає завданню, оскільки пристрій повинен мати можливість зміни конфігурації. Тому слід використати таблицю відповідності зберігання вихідних значень для визначення функцій активації. При використанні цього методу, кілька різних функцій активації можуть бути реалізовані апаратно.

Структура MLP-SoC. Рис. 2. показує загальну структуру МЛП-SoC. Вона включає в себе LEON 2 (основний процесор), MLP співпроцесор (нейронних обчислень), контролер пам'яті SDRAM, інтерфейс камери і шину АНВ/АРВ АМБА. Всі ці компоненти об'єднані в FPGA макетної платформи.

LEON 2 являє собою 32-розрядний RISC-процесор сумісний з архітектурою SPARC V8. Він легко програмується і, таким чином, дуже підходить для SoC.

Окрім того, програмне забезпечення, написане мовою C, може бути безпосередньо виконане на ядрі LEON 2 з використанням засобу крос-трансляції.

Програмування ядра LEON 2 (рис. 3.) може бути здійснено з використанням відкритого транслятора мови VHDL в код FPGA макетної платформи. Контролер інтерфейсу камери і схема вводу/виводу I2C здатні обробляти декілька датчиків зображення з використанням їх фіксованої логіки.

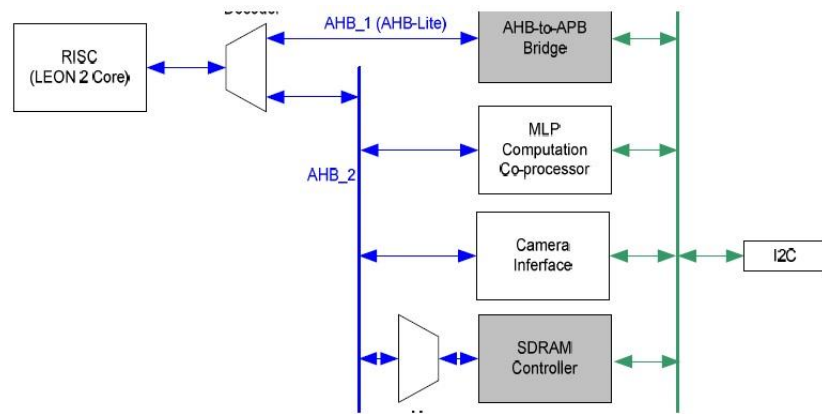


Рис. 2. Архітектурний огляд MLP-SoC

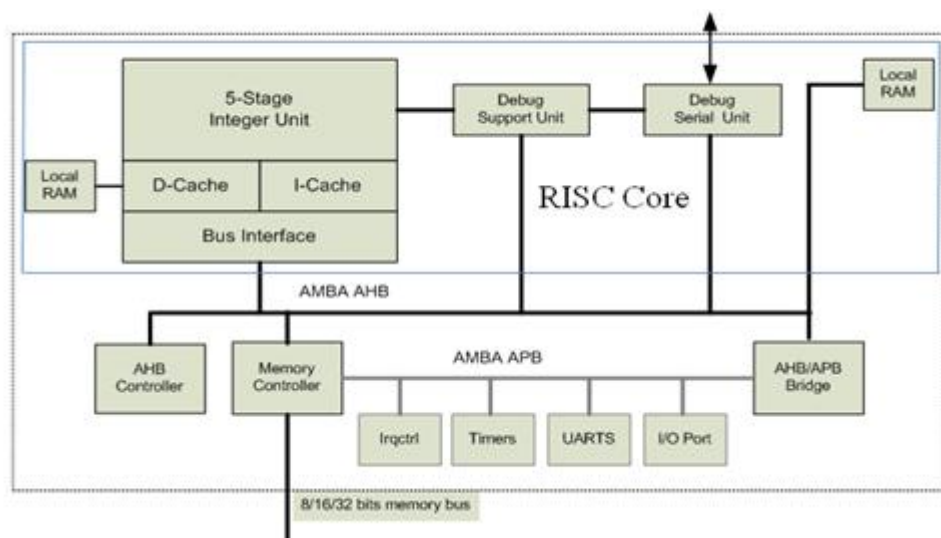


Рис. 3. Блок-схема процесора LEON 2

MLP співпроцесор На рис. 4 показана архітектура вбудованого MLP співпроцесора, основним призначенням якого є обчислення нейронів. Як видно на малюнку, співпроцесор складається з двох основних частин - блок хост-інтерфейсу для доступу до пам'яті та інтерфейс шини і MLP блок для нейронних обчислень.

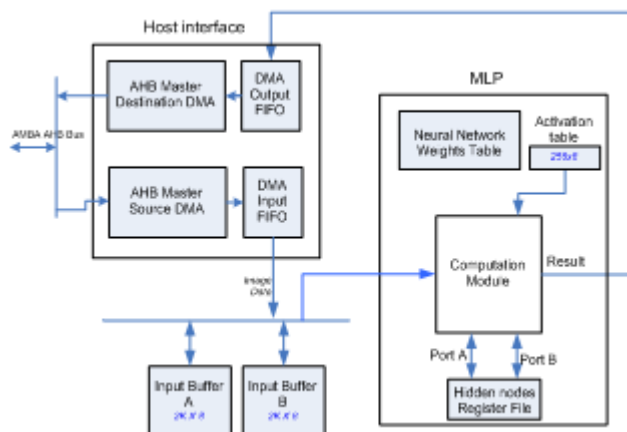


Рис. 4. Огляд архітектури MLP співпроцесора

Блок хост-інтерфейсу несе відповідальність за обслуговування шини між MLP співпроцесором та іншими контролерами.

Він складається з двох блоків прямого доступу до пам'яті (DMA), вхідного DMA і вихідного DMA. Вхідний DMA приймає блок даних із зовнішньої пам'яті і зберігає його у вхідних буферах (2K x 8 бітів). Два буфери завжди готові для прийому даних, а інші зберігають дані для подальших обчислень. Після прийому даних блок хост-інтерфейсу посилає сигнал до блоку MLP, щоб почати обчислення задачі для поточного вхідного сигналу. Коли обчислення завершується, вихідний DMA блок передає згенеровані блоки даних з МЛП блоку в зовнішню пам'ять. Використання DMA потоку даних є корисним, коли розмір і кількість вхідних/вихідних даних досить значні.

Для того, щоб врахувати обмеження, описані в таблиці 1, блок MLP складається зі сховищ даних і обчислювального модуля. Є три різних постійних блоків пам'яті: таблиця функцій активації, файл реєстру внутрішніх вузлів і таблиця ваг.

Функція активації, наприклад, сигмовидна або гіпертангенціальна, можуть бути реалізовані в таблиці функцій активації без зміни апаратних засобів.

Таблиця ваг містить 128K * 19 біт, 12 біт використовуються для збереження значення ваги і 7 біт використовуються для збереження індексу внутрішнього вузла.

Файл реєстру внутрішніх вузлів містить 128 * 24 біт для зберігання перехідних результатів внутрішніх вузлів.

Рис. 5. показує схему обчислювального модуля. Модуль обчислення отримує вхідні сигнали з вхідного буфера і обчислює значення активації всіх вузлів шарів послідовно до обчислення значень вихідних вузлів. Потім він відправляє вихідні значення для блоку хост-інтерфейсу для збереження їх в пам'яті SRAM. Рис. 5 також демонструє точність вбудованої логіки.

Програмна реалізація МЛП-SoC може бути повністю синтезована на основі VHDL моделі і передані в FPGA (XILINX X2CV8000) макетної платформи.

Ця архітектура МЛП-SoC забезпечує швидку обробку нейронних зв'язків і трансферних функцій, і добре підходить для нейронних моделей MLP-типу. Тактова частота FPGA 30 МГц. Цей тактовий сигнал подається в усі компоненти макетної платформи.

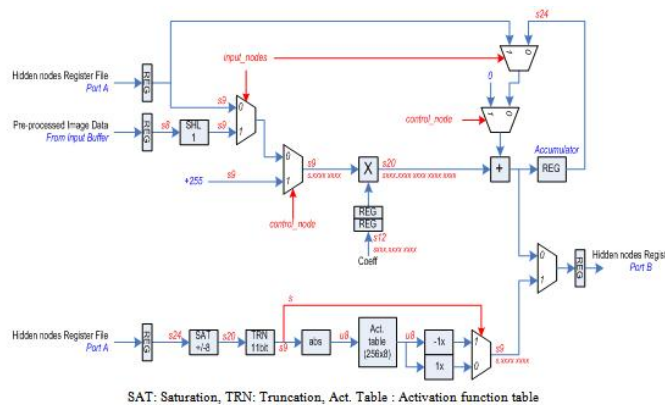


Рис. 5. Схема обчислювального модуля MLP співпроцесора

Приклад додатку: Система розпізнавання символів (OCR). OCR процес, за допомогою якого комп'ютер переводить оцифровані зображення символів у текст. Ця система є основою для багатьох різних типів вбудованих додатків, таких як портативні перекладачі, електронні словники і щоденники. Алгоритм цільової системи розпізнавання складається з трьох основних етапів, як показано на рис. 6.

По-перше, отримання зображення зі сканера MKM MT9V112, підключеного до інтерфейсу камери.

По-друге, попередньої обробки, яка виконується для того, щоб сегментувати зображення на окремі символи, використовуючи метод гістограм. Виділені символи перетворюються в бінарні матриці зображень (0 або 255). Тоді проводиться нормування похилих і підгонка розміру до отримання вхідного зображення MLP розміром 30x24 (в пікселях).

Оскільки структура нейронної мережі (число вузлів, функції активації) може змінюватися для конкретного додатку з метою підвищення продуктивності, а SoC для MLP повинна прийняти ці зміни без зміни конфігурації апаратної платформи, то, щоб показати здатність MLP_SoC до

переналаштування, слід побудувати два MLP на тій же архітектурі. Таблиця 2 показує конфігурації реалізованих MLP.

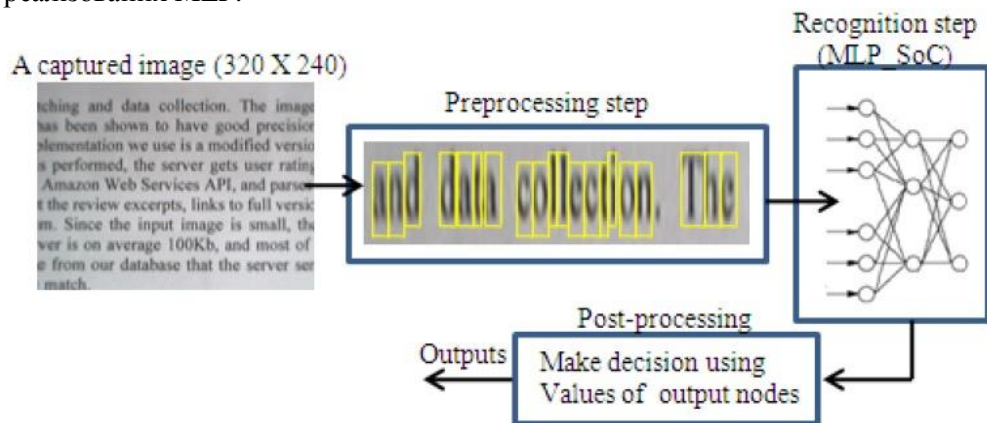


Рис. 6. Потік обробки реалізованої системи OCR

Оскільки структура нейронної мережі, число вузлів, функції активації, може змінюватися для конкретного додатка з метою підвищення продуктивності, А SoC для MLP повинні прийняти ці зміни без зміни апаратної платформи. То щоб показати здатність MLP_SoC до переналаштування, слід побудувати два MLP на тій же архітектурі. Таблиця 2 показує конфігурації реалізованих MLP. Кількість внутрішніх вузлів кожного MLP вибирається випадковим чином (3 ~ 5% від числа вхідних вузлів) та фіксується під час тренування і тестового розпізнавання.

Таблиця 2

Конфігурації MLP мереж

Назва параметру	Версія 1	Версія 2
Вузлів вхідних/внутрішніх/вихідних	720/24/26	720/32/26
Тип функції активації	Гіперболічно-тангенціальна	Сигмоїдна
Тип комутації вузлів	Повний	Частковий
Тип алгоритму навчання	Зворотнє поширення	Зворотнє поширення
Відсоток розпізнавання	94%	98%

Інше важливе питання для оцінки MLP-SoC є швидкість розпізнавання. Слід перевірити необхідний час роботи кожного модуля системи розпізнавання на MLP. У якості тестового зразка був взятий документ розміром 320X240 (пікселів), що містить 260 символів.

У таблиці 3 наведені час всіх модулів, затрачений для виконання цього завдання.

Таким чином, система розпізнавання здатна обробляти майже 43 символи в секунду. MLP обчислення займає всього 3.9 сек, в той час як програмна реалізація цільової MLP займає аж 869 секунд під час виконання на процесорі LEON 2.

Таблиця 3

Швидкість кожного модуля обробки для системи оптичного розпізнавання символів

Стадія алгоритму	Модуль	Час
Сегментування символів	Програма на LEON 2	690 мсек
Підготовка і нормалізація	Програма на LEON 2	1240 мсек
Нейронна мережа	Апаратні засоби	3980 мсек
Рішення і запис результату	Програма на LEON 2	120 мсек
Всього		6030 мсек

Цей результат в основному досягнутий за рахунок обчислення нейронної мережі MLP співпроцесором, що прискорило процес у 223 разів порівняно до програмної реалізації.

Комерційні системи програмного забезпечення OCR, реалізовані для серверів або настільних комп'ютерів, які зазвичай мають більш високі апаратні можливості, такі як потужні процесори, не займають багато часу для розпізнавання. Однак, мобільні пристрої часто мають обмеження у своїх апаратних можливостях через вимоги до обмеження споживання електроенергії.

Таким чином, апаратне прискорення є кращим рішенням для реалізації функцій розпізнавання образів у пристроях з обмеженими обчислювальними можливостями.

Висновки. У цій статті була запропонована архітектура MLP-SoC придатна для малогабаритних інтелектуальних пристроїв. Архітектура була випробувана і перевірена за допомогою макетної платформи FPGA. Платформа дає можливість без зміни існуючого обладнання побудувати різні варіанти прикладних систем шляхом реконфігурації SoC. Отже, MLP-SoC доцільно використати для мобільних пристроїв, які потребують функції розпізнавання образів.

1. Иванов А.И. Подсознание искусственного интеллекта: программирование автоматов нейросетевой биометрии языком их обучения. Электронная книга издательства ОАО "ПНИЭИ", 2012. —125 с.
2. E. M. Ortigosa, A. Canas, E. Ros, P. M. Ortigosa, S. Mota and J. Diaz, "Hardware description of multi-layer perceptrons with different abstraction levels," *Microprocessors and Microsystems*, vol. 30, pp. 435 – 444, 2006.
3. S. Vitabile, V. Conti, F. Gennaro and F. Sorbello, "Efficient MLP Digital Implementation on FPGA," *Proceedings of the 8th Euromicro conference on DSD*, 2005.
4. Новіков О., Кашенко С. Розпізнавання сервісів tcp/ip за допомогою нейронних мереж . *Періодичний науково-технічний збірник "КПІ – 1*, 2000 р., С 222-227.
5. Бонгард М. М. Проблема узнавания. — М. : Наука, 1967. — 320 с.
6. Керниган Б., Ритчи Д. Язык программирования Си. - СПб.: Невский Диалект, 2001. – 352 с.
7. LEON2 processor user's manual, Gaisler Research, <http://www.gaisler.com>.

УДК 519.618

Гринюк С.В., Міскевич О.І.

Луцький національний технічний університет

АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА ОБЛІКУ ТА РУХУ ТОВАРІВ НА СКЛАДІ ЗАСОБАМИ RAD STUDIO 2010

Гринюк С.В., Міскевич О.І. Автоматизована інформаційна система обліку та руху товарів на складі. В статті розглядається створення інформаційної системи засобами Rad Studio 2010, яка призначена для автоматизації роботи на складі. Дана програма забезпечує контроль обліку та руху товарів на складі підприємства.

Ключові слова: інформаційна система, автоматизація, програма, база даних.

Гринюк С.В., Мискевич О.И. Автоматизированная информационная система учета и движения товаров на складе. В статье рассматривается создание информационной системы средствами Rad Studio 2010, которая предназначена для автоматизации работы на складе. Данная программа обеспечивает контроль учета и движения товаров на складе предприятия.

Ключевые слова: информационная система, автоматизация, программа, база данных.

Grunjuk S., Miskevych O. Automated information system of accounting and movement of goods at the moment. This article describes how to create an information system means Rad Studio 2010, which is designed to automate the warehouse. This program provides accounting and control of the movement of goods in the warehouse of the company.

Keywords: information system, automation, application database.

Вступ. Термін «інформаційна система» належить до класу програмних продуктів, що автоматизують ведення бізнесу. Система називається інформаційною, якщо вона підтримує інформаційне забезпечення бізнесу.

Інформаційна система – це сукупність засобів збору, зберігання, передачі, оброблення інформації в певній Про для досягнення поставленої мети у процесі управління.

Автоматизована ІС – сукупність інформації, різних методів і моделей, апаратних, програмних, організаційних, технологічних засобів і відповідних фахівців. Отже, інформаційна система - це організаційно впорядкована сукупність фахівців, інформаційних ресурсів та інформаційних технологій, зокрема з використанням засобів обчислювальної техніки і зв'язку, що реалізують такі інформаційні процеси, як отримання вхідних даних; обробка цих даних і/або зміна власного внутрішнього стану (внутрішніх зв'язків/відносин), видача результату або зміна свого зовнішнього стану (зовнішніх зв'язків/відносин). За допомогою ІС надається можливість встановлення зв'язку між усіма елементами бізнес-процесів підприємства, що покращує можливості планування, контролю й регулювання процесів. Інформаційна система, при формуванні якої використано принцип зворотного зв'язку на всіх рівнях управління і сучасні ІКТ, забезпечує зв'язок між елементами системи управління й елементами бізнес-процесів, тобто між усіма етапами прийняття рішень, а також надає можливість накопичення даних, аналізу і моделювання.

У більшості випадків для створення власної інформаційної системи неможливо обійтися без використання баз даних. Важливою особливістю СУБД є забезпечення виконання запитів до бази даних. Ще однією важливою особливістю більшості сучасних СУБД є забезпечення режиму мультидоступу [1,2].

Постановка проблеми. Торгівля є однією з найбільших галузей економіки будь-якої країни як за обсягом діяльності, так і за чисельністю зайнятого в ній персоналу, а підприємства цієї галузі є найбільш масовими. Діяльність торгових підприємств пов'язана із задоволенням потреб кожної людини, схильна до впливу безлічі факторів і охоплює широкий спектр питань організаційно-технологічного, економічного і фінансового характеру, що вимагають повсякденного рішення. Динамічна конкурентне середовище, нові умови ведення бізнесу висувають підвищені вимоги до організації управління на торговельному підприємстві.

Виходячи з сучасних вимог, що пред'являються до якості роботи сучасних торговельних підприємств, не можна не відзначити, що ефективна робота його цілком залежить від рівня оснащення компанії інформаційними засобами на базі комп'ютерних систем автоматизованого складського обліку.

Комп'ютерний облік має свої особливості і радикально відрізняється від звичайного. Комп'ютер полегшує облік, скорочуючи час, потрібний на оформлення документів та узагальнення накопичених даних для аналізу ходу торговельної діяльності, необхідного для

управління нею. Таким чином, при застосуванні комп'ютера "кількість переходить в якість": збільшення швидкості розрахунків робить можливим якісне поліпшення самої схеми побудови торгівлі [3].

Актуальність створення автоматизованої інформаційної системи для обліку товарів на складі полягає в тому, що від достовірності, об'єктивності та повноти відображення обліку товарів й звітності господарських фактів, пов'язаних із рухом товарів залежить правильність визначення, а також правдивість і неупередженість усіх суттєвих для користувачів показників, які характеризують результати господарської діяльності та реальний фінансовий стан підприємства.

Метою роботи є створення проекту автоматизованої інформаційної системи для обліку та руху товарів на підприємстві.

До основних етапів створення даної системи належать:

- а) дослідження поставленої задачі та предметної області;
- б) дослідження існуючих інформаційних систем для автоматизації роботи підприємств;
- в) розробка концепції автоматизації діяльності підприємства;
- г) розробка проекту автоматизованої системи;
- д) розробка прототипу системи та його реалізація.

Інформаційну систему будь-якого підприємства можна представити різноманітними інформаційними потоками - документів, розпоряджень, запитів, які циркулюють усередині підприємства, виходять за межі підприємства або входять із зовнішнього середовища. На сучасних підприємствах практично вся робота з інформацією автоматизована, розроблено спеціальні програми, які дають змогу вести на комп'ютері бухгалтерський облік, документообіг, маркетингові дослідження, проводити прогнозування і стратегічне планування тощо. Але, крім автоматизації, актуальним залишається питання про ефективну побудову інформаційної системи та оптимізації інформаційних потоків.

Наявність відлагодженої автоматизованої інформаційної системи значно спрощує процес управління підприємством. ІС дає змогу своєчасно зібрати, відсортувати, опрацювати потрібну інформацію і прийняти правильне рішення. Інколи несвоєчасно прийняте рішення внаслідок нестачі або несвоєчасного надходження інформації може призвести до загибелі підприємства. Тому створенню і підтримці ефективного функціонування інформаційної системи підприємства слід приділяти велику увагу.

У сучасному інформаційному суспільстві обробка інформації є необхідною умовою організації виробництва. Для прийняття ефективних управлінських рішень виникає необхідність, враховуючи різноманітні фактори, обробляти значну кількість інформації. Стрімке зростання обсягу науково-технічної інформації, з одного боку, і розвиток обчислювальної техніки, з іншого боку, викликали необхідність створення нових інформаційних технологій. Електронні сховища інформації дозволили компактно зберігати, багатократно використовувати, сортувати, редагувати, відображати на екрані дисплея, а при необхідності отримувати копії документів на папері. Такі інформаційні системи забезпечували доступ до інформації великому числу користувачів незалежно від їх географічного місцезнаходження і дозволяли оперувати великими об'ємами даних.

База даних (БД) – це значна кількість однорідних даних з конкретної предметної галузі, які зберігаються в пам'яті комп'ютера.

Діяльність з пошуку прийнятих засобів для автоматизованої обробки інформації сприяла створенню на початку 60-х років спеціальних програмних комплексів, «Систем керування базами даних» (СКБД), які повинні надавати користувачеві способи введення, збереження даних, а також опису структури даних. Файли, в яких забезпечувалось збереження даних і опис їх структури і які знаходились під управлінням СКБД, були названі банками даних, а пізніше «Базами даних» (БД). СКБД повинна також надавати механізми пошуку даних за запитом користувачів, забезпечувати захист даних від некоректного оновлення та несанкціонованого доступу, підтримувати бази даних в актуальному стані [4].

Оскільки пропонується, що обробка інформації буде проводитись на робочих місцях за допомогою персонального комп'ютера, слід вибрати реляційну модель даних. Виберемо з реляційних СУБД - СУБД Access. Вона має зручний інтерфейс, а також найбільш поширена серед інших СУБД.

СУБД дозволяє підтримати посилальну цілісність, задавати обмеження на поля (в тому числі - на початкові значення, за замовчуванням).

У СУБД Access є наступні основні типи даних: текстовий, числовий, лічильник, логічний, грошовий, дата / час.

Основними об'єктами СУБД є таблиці, форми (вхідні об'єкти), запити, звіти (вихідні об'єкти). При використанні програмування прямо додатково застосовують такі об'єкти, як макроси і модулі.

СУБД Access дозволяє працювати в локальному та мережевому режимі.

Для взаємодії користувача з базою даних використовуються програмні комплекси – системи керування базами даних (СКБД).

Сучасні СКБД забезпечують:

- набір засобів для підтримки таблиць і співвідношень між зв'язаними таблицями;
- введення, модифікацію інформації, пошук і представлення інформації в текстовому або графічному вигляді;
- засоби програмування, за допомогою яких ви можете створювати власні додатки.

Отже, для створення бази даних і роботи з нею крім комп'ютера потрібно мати відповідний програмний продукт – СКБД. Робота з базами даних має такі етапи:

- створення структури бази даних;
- введення даних;
- редагування структури і даних;
- пошук інформації в базі даних;
- оформлення звітів.

Структурна схема інформаційної системи обліку та руху товарів на складі представлена на рис. 1.

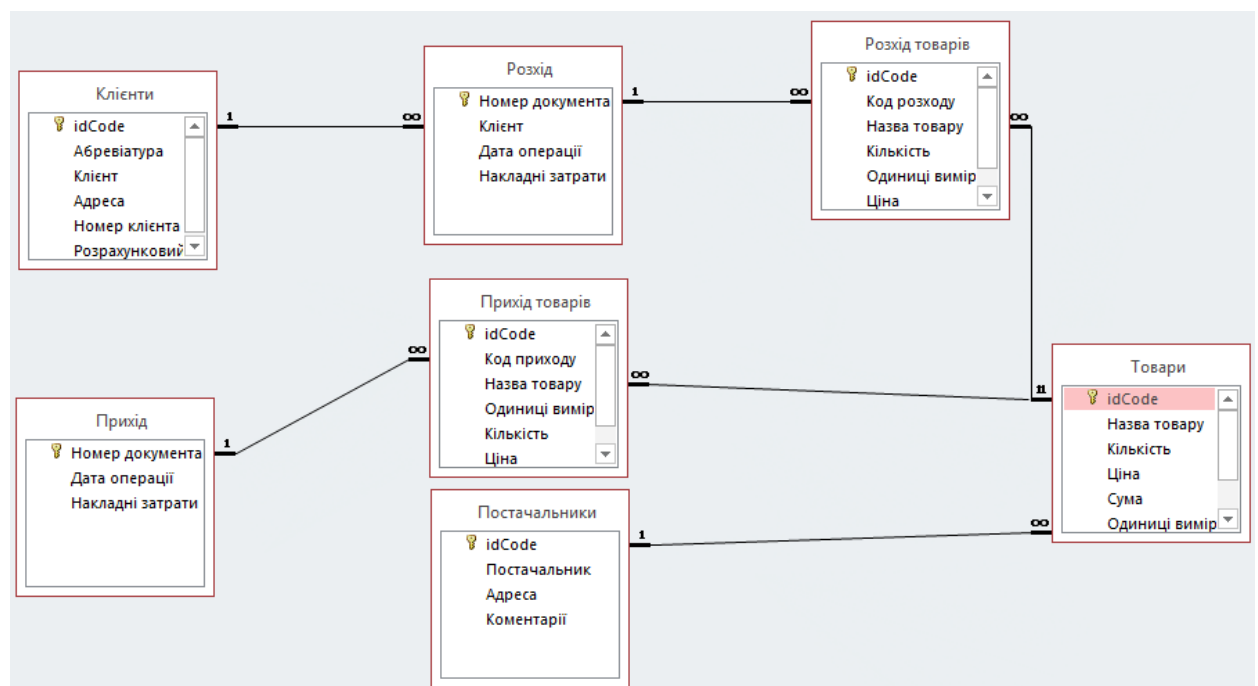


Рис. 1 – Структурна схема бази даних інформаційної системи

Сучасні програмно – інструментальні засоби розробки ПЗ характеризуються великою різноманітністю характеристик. Так, в даний час інструментальні засоби дозволяють:

- базуючись на стандартних компонентах створювати інтерфейс програми в залежності від стану системи передавати управління різним процесам;
- створювати бази даних і оболонки для баз даних;
- виконувати коректну обробку виняткових ситуацій, що дозволяє підвищити надійність ПЗ.

Сучасні засоби розробки характеризуються такими параметрами :

- підтримка об'єктно – орієнтованого стилю програмування ;

- б) можливість використання CASE – технологій для проектування розроблюваної системи, використання візуальних компонент для наочного проектування інтерфейсу;
- в) наявність візуальної технології розробки інтерфейсу;
- г) можливість використання алгоритмів реляційної алгебри для управління реляційними базами даних;
- д) надання коштів синхронізації і контролю версій складових частин проекту (ці кошти використовуються при розробці програмного забезпечення групами програмістів) ;
- е) створення інсталяційних пакетів для поширення розробленого програмного забезпечення.

При створенні прототипу програмного забезпечення головними критеріями вибору програмно- інструментальних засобів розробки є:

- а) швидкість розробки додатків;
- б) зручність використання;
- в) можливість швидкого внесення змін у програму ;

Забезпечити мінімальний час розробки можна тільки при виконанні цих умов. Виходячи з наведених вимог, виділимо наступні характеристики засобів розробки програмного забезпечення:

- а) вартість IDE;
- б) невисока потреба ресурсів;
- в) наочність розробки інтерфейсу;
- г) надані можливості роботи з базами даних;
- д) швидкість роботи розробленого програмного забезпечення;
- е) обробка виняткових ситуацій;
- ж) час створення розробленого програмного забезпечення;
- з) зручність експлуатації;
- и) засоби контролю версій складових частин проекту;
- к) наявність зручної довідкової системи.

Для вибору інструментального засобу скористаємося методом варіантних мереж. Цей метод призначений для вибору найкращого варіанта з декількох запропонованих і складається з наступних етапів:

- а) визначення критеріїв , за якими буде зроблене порівняння і
- б) ступенем їх важливості ;
- в) кожен варіант оцінюється за отриманим переліком критеріїв (виходить чисельне значення - оцінка) ;
- г) знаходження загальної кількості балів для кожного з варіантів (можна враховувати важливість критеріїв).

В результаті застосування методу варіантних мереж встановлено, що кращим інструментальним засобом з точки зору розробника в даному випадку є середовище RAD Studio 2010, а саме Delphi 2010 [4].

Головною формою даної програми є Form1 з ім'ям Sklad. Фактично вона була згенерована автоматично. Більшість Windows-програм мають головне меню, яке представляє собою ієрархічну структуру виконуючих функцій та команд. Для створення головного меню використовується елемент управління MainMenu, розміщений на панелі елементів. У головному меню будуть розміщуватись наступні пункти верхнього рівня: «Файл», «Створити», «Пошук», «Звіт» та «Про програму».

Головне вікно програми представлено на рис. 2.

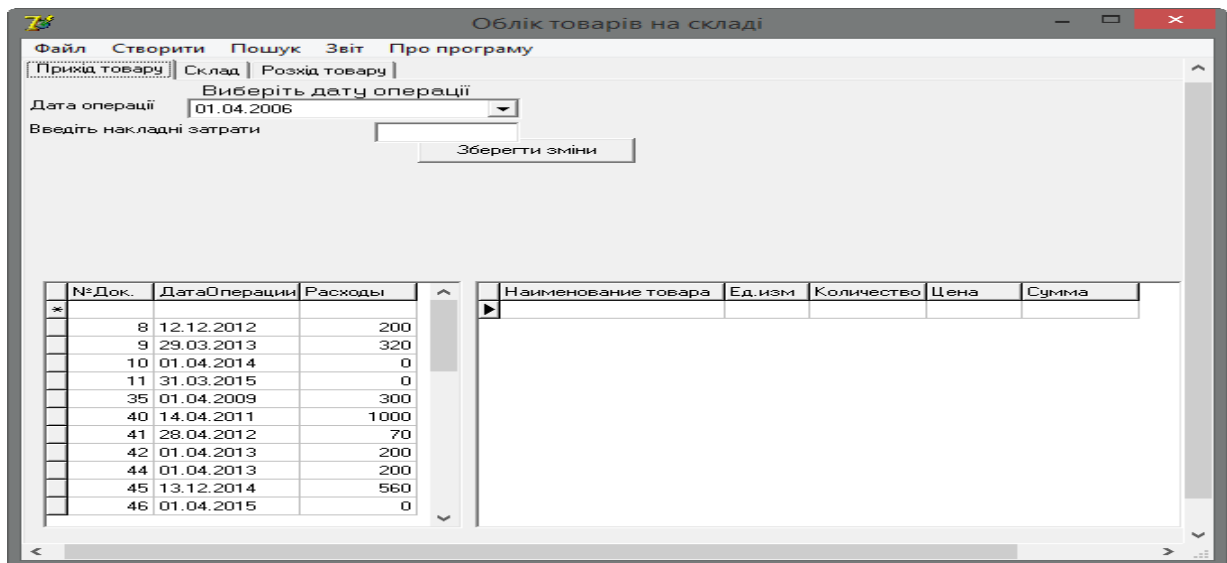


Рис. 2 – Головне вікно програми

Висновки. Створена автоматизована інформаційна система покращує ефективність роботи підприємства. Діалог з користувачем здійснюється за допомогою розробленого програмного інтерфейсу.

Виходячи з сучасних вимог, що пред'являються до якості роботи сучасних торговельних підприємств, не можна не відзначити, що ефективна робота його цілком залежить від рівня оснащення компанії інформаційними засобами на базі комп'ютерних систем автоматизованого складського обліку.

Розробивши концепцію автоматизованого програмного рішення було розроблено автоматизовану інформаційну систему обліку та руху товарів на підприємстві. Створена система автоматизації направлена на автоматизацію невеликих підприємств та дозволяє реалізовані всі функціональні вимоги.

1. Автоматизация работы предприятия [Електронний ресурс] / Режим доступу: <http://www.mlsit.com.ua/products/archimed> – 16.03.2012. – Назва зі сторінки Інтернету.
2. Автоматизована система керування [Електронний ресурс] / Режим доступу: http://uk.wikipedia.org/wiki/Автоматизована_система_керування – 21.04.2012. – Назва зі сторінки Інтернету.
3. Информационные технологии в экономике [Електронний ресурс] / Моисеенко Е.В., Лаврушина Е.Г. – Режим доступу: http://abc.vvsu.ru/Books/up_inform_tehnot_v_ekon/page0005.asp – 16.04.2012. – Назва зі сторінки Інтернету.
4. Олійник А.В., Шацька В.М. Інформаційні системи і технології у фінансових установах [Текст] / А.В.Олійник, В.М.Шацька – Навчальний посібник - Львів: "Новий Світ-2000", 2006 - 436 с ISBN 5-7218-0099-2.

УДК 004.434

Губаль Г. М.

Луцький національний технічний університет

АНАЛІЗ ДЕЯКИХ КОМАНД МОВИ СТВОРЕННЯ МАТЕМАТИЧНИХ ТЕКСТІВ L^AT_EX

Губаль Г. М. Аналіз деяких команд мови створення математичних текстів L^AT_EX. У статті досліджуються класи документів, які забезпечують розширення можливостей задання розмірів шрифтів, при цьому, як локально, так і глобально. Проаналізовано деякі команди, які використовуються у мові створення математичних текстів L^AT_EX.

Ключові слова: розширені розміри шрифтів, клас memoir, footnotesizes, L^AT_EX, пакет empheq.

Губаль Г. Н. Анализ некоторых команд языка создания математических текстов L^AT_EX. В статье исследуются классы документов, обеспечивающих расширение возможностей задания размеров шрифтов, при этом, как локально, так и глобально. Проанализировано некоторые команды, используемые в языке создания математических текстов L^AT_EX.

Ключевые слова: расширенные размеры шрифтов, класс memoir, footnotesizes, L^AT_EX, пакет empheq.

Hubal H. M. Analysis of some commands of the L^AT_EX language of mathematical texts . The document classes, providing the extension of opportunities to set font sizes while both locally and globally are investigated in the article. Some commands in the L^AT_EX language of the mathematical texts are analyzed.

Keywords: extended font sizes, the memoir class, footnotesizes, L^AT_EX, the empheq package.

Вступ. Робота з системою L^AT_EX, призначеною для створення математичних текстів [1, 2, 5-10, 12-15] подібна до програмування, тим більше, що ця система розрахована на використання текстового режиму, командного рядка і багатьох конфігураційних файлів [3, 4, 11].

У статті досліджуються класи документів, які забезпечують розширення можливостей задання розмірів шрифтів, при цьому, як локально, так і глобально. При цьому використовуються пакети moresize, anyfontsize, mathptmx, lmodern. Наводяться і аналізуються деякі команди, які використовуються у мові створення математичних текстів L^AT_EX.

Основна частина. У документах L^AT_EX можна змінювати розмір шрифту як локально (змінювати розмір шрифту окремих літер), так і глобально (змінювати розмір шрифту у всьому документі). При зміні розміру шрифту глобально змінюється розмір шрифту не тільки основного тексту, а й розмір шрифту заголовків, виносок, колонтитулів, колонцифр та ін. Тому іноді доцільно змінювати шрифт локально.

Щоб змінити глобально розмір шрифту в класах документів article, report і book (за замовчуванням у цих класах задано розмір шрифту 10pt), необхідно змінити опцію класу, яка відповідає за розмір шрифту. Наприклад, розмір шрифту 11pt у класі документа article задається такою командою:

```
\documentclass[11pt]{article}
```

Якщо необхідно використати розширені розміри шрифтів у основних класах, підключають пакети extsizes. Ці пакети генерують шрифти розмірів 8pt, 9pt, 14pt, 17pt і 20pt. Наприклад, шрифти розмірів 17pt і 14pt у класах extreport і extarticle генеруються такими командами:

```
\documentclass[17pt]{extreport}
```

```
\documentclass[14pt]{extarticle}
```

Класи документів ams мають більше розмірів шрифтів, ніж стандартні класи.

Щоб змінити розміри шрифту локально у стандартних класах документів (розміри слова, абзацу, виноска), необхідно використовувати макроси, які задаються одним оператором або оточенням:

```
\begin
```

```
...
```

```
\end
```

Наприклад,

```
{\large Це більший розмір шрифту.\par}

\begin{footnotesizes}
  Розмір цього шрифту має розмір шрифту виноски.
\end{footnotesizes}
```

Пакет `moresize` дає можливість використати макроси `\ssmall` і `\HUGE`. Макрос `\ssmall` генерує розмір шрифту в межах між `\scriptsize` і `\tiny`, а макрос `\HUGE` генерує розмір шрифту більший від найбільш допустимого за замовчуванням.

Стандартні розміри шрифтів у \LaTeX зведемо у таблицю 1.

Опція класу	10pt	11pt	12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Табл. 1. Стандартні розміри шрифтів у \LaTeX

Проблему збільшення розмірів стандартних шрифтів у \LaTeX за допомогою макроса `\HUGE` можна розв'язати, використавши типи масштабованих шрифтів з пакета `PSNFSS`. Наприклад, у класі документа `article` можна підключити пакети `mathptmx` і `moresize` і використати макроси `\HUGE`, `\ssmall`:

```
\documentclass{article}
\usepackage{mathptmx}
\usepackage[10pt]{moresize}
\begin{document}
  {\ssmall Розмір шрифту знаходиться в межах між \scriptsize і \tiny}
  {\HUGE Розмір шрифту більший від 25pt}
\end{document}
```

Аналогічним пакетом до пакета `moresize` є пакет `anyfontsize`. У цьому пакеті використовується команда `\fontsize` з двома обов'язковими аргументами: нового розміру шрифту і `baselineskip`:

```
\fontsize{size}{baselineskip}
```

Розглянемо клас документів `memoir`, який забезпечує розширені розміри шрифтів від 9pt до 60pt глобально. При цьому використовуються такі опції: 9, 10, 11, 12, 14, 17, 20, 25, 30, 36, 48, 60pt. Наприклад, розмір шрифту 25pt у класі `memoir` генерується такою командою:

```
\documentclass[25pt, extrafontsizes]{memoir}
```

Для задання розмірів шрифтів більших від 25pt необхідно використовувати пакет `Latin Modern` в T1 кодуванні за допомогою таких команд:

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
```

Щоб змінити розміри шрифтів локально у класі документів memoir використовують макроси як і в стандартних класах. На кінцях таблиці стандартних розмірів шрифтів у L^AT_EX використовуються такі макроси класу memoir: \minuscule і \HUGE.

Розглянемо деякі команди, які використовуються для представлення математичного тексту.

Розглянемо можливості зміни міжрядкових інтервалів. Щоб встановити одинарний, подвійний міжрядковий інтервал або міжрядковий інтервал 1,5 рядка, необхідно підключити пакет setspace і використати одну з команд \singlespacing, \doublespacing або \onehalfspacing відповідно. Наведемо приклад тексту з використанням команди \singlespacing:

The state of a symmetric system of finitely many identical particles is determined by a solution of the Cauchy problem for the BBGKY hierarchy of equations with initial data possessing the factorization property (the chaos property).

Наведений приклад тексту з використанням команди \onehalfspacing має вигляд:

The state of a symmetric system of finitely many identical particles is determined by a solution of the Cauchy problem for the BBGKY hierarchy of equations with initial data possessing the factorization property (the chaos property).

Розглянемо випадок, коли виникає необхідність помістити формулу в рамку. Наведемо приклад створення формули в рамці в оточенні align. При цьому представимо оператор \Aboxed:

```
\makeatletter
\newcommand\Aboxed[1]
{\@Aboxed#1\enddne}
{\settowidth
\@templ{$\displaystyle#1{}}$}
}
\setlength\@templ
{\@templ+\abxsep+\abxrule}
\global\@templ=\@templ\kern\@templ
&
\kern-\@templ\boxed{#1#2}
\makeatother
```

У документі звернемось до оператора \Aboxed за допомогою такого коду:

```
\begin{align}
\Aboxed{\&\frac{d}{dt}F(t)=\mathcal{A}F(t)}\\
&\&F(t)=U(t)F(0)
\end{align}
```

Наведений код генерує такий текст:

$$\boxed{\frac{d}{dt}F(t) = A F(t)} \quad (1)$$

$$F(t) = U(t)F(0)$$

У випадках, коли виникає необхідність створити ліву фігурну дужку в системі нумерованих рівнянь, яка задається оточенням subequations, необхідно підключити пакет empheq з використанням команди \empheqlbrace. При цьому використовується такий код:

```
\begin{subequations}
\begin{empheq}[left=\empheqlbrace]{align}
&\&\frac{d}{dt}F(t)=\mathcal{A}F(t)}\\
&\&\left. F(t) \right|_{t=0}=F(0)
\end{empheq}
\end{subequations}
```

`\end{subequations}`

Наведений код генерує такий текст:

$$\frac{d}{dt}F(t) = A F(t) \quad (1)$$

$$F(t)|_{t=0} = F(0) \quad (2)$$

Іноді буває зручно нумерувати абзаци для подальшого рецензування документа. У цьому випадку визначимо оператор `\Header` за допомогою такого коду:

```
\newcounter{vcount}
\def\Header#1
{\medskip\hbox{\bfseries #1}\setcounter{vcount}{1}
\everypar{\arabic{vcount}}
\stepcounter{vcount}}
```

Звернутись до цього оператора у тексті документа можна так:

```
\Header{Introduction}
```

The evolution of states of many-particle systems is described by the BBGKY hierarchy of equations.

A solution of the Cauchy problem for the BBGKY hierarchy of equations is represented as the expansion over particle groups whose evolution is governed by the cumulants.

States of many-particle systems can be described in terms of the one-particle distribution function that satisfies the kinetic equation.

Наведений код генерує такий текст з нумерованими абзацами:

Introduction

1. The evolution of states of many-particle systems is described by the BBGKY hierarchy of equations.

2. A solution of the Cauchy problem for the BBGKY hierarchy of equations is represented as the expansion over particle groups whose evolution is governed by the cumulants.

3. States of many-particle systems can be described in terms of the one-particle distribution function that satisfies the kinetic equation.

Висновки. У статті досліджено класи документів, які забезпечують розширення можливостей задання розмірів шрифтів, при цьому, як локально, так і глобально. При цьому використано пакети `moresize`, `anyfontsize`, `mathptmx`, `lmodern`. Проаналізовано команди, які використовуються у мові створення математичних текстів $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ для: зміни міжрядкових інтервалів, представлення формули в рамці, створення лівої фігурної дужки у системі нумерованих рівнянь, генерування нумерації абзацив.

1. Балдин Е. М. Компьютерная типография $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ / Е. М. Балдин. – СПб.: БХВ-Петербург, 2008.
2. Беляков Н. С. $\text{T}_{\text{E}}\text{X}$ для всех / Н. С. Беляков, В. Е. Палош, П. А. Садовский. – М.: Книжный дом «Либроком», 2009.
3. Губаль Г. М. Стратегії для створення математичної статті у видавничій системі $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 13.
4. Дубинич В. Н. Использование системы $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ для подготовки научных изданий / В. Н. Дубинич, М. В. Дубинич // Перспективы развития высшей школы: материалы V Международной науч.-метод. конф., Гродно: ГГАУ, 2012.
5. Жуков М. Ю. Оформление математических текстов при помощи пакета $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ООО «ЦВВР», 2003.

6. Жуков М. Ю. L^AT_EX2_ε: искусство набора и вёрстки текстов с формулами / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2009.
7. Каров П. Шрифтовые технологии. Описание и инструментарий / П. Каров. Перевод с английского Карпинского О. С. и Куликова И. И. под редакцией, с предисловием и дополнением Ефимова В. В. – М.: Мир, 2001.
8. Кнут Д. Е. Всё про T_EX / Д. Е. Кнут. Пер. с англ. М. В. Лисиной. – Протвино: АО RDT_EX, 1993.
9. Котельников И. А. L^AT_EX по-русски / И. А. Котельников, П. З. Чеботаев. – Новосибирск: Сибирский хронограф, 2004.
10. Львовский С. М. Набор и вёрстка в системе L^AT_EX / С. М. Львовский. – 3-е изд., испр. и доп. – М.: МЦНМО, 2003.
11. Ширяева Е. В. Введение в T_EX–программирование / Е. В. Ширяева, И. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2010.
12. Goossens M. The L^AT_EX companion / M. Goossens, F. Mittelbach, A. Samarin. – Addison-Wesley, 1994. Русский перевод: Гуссенс М. Путеводитель по пакету L^AT_EX и его расширению L^AT_EX2_ε / М. Гуссенс, Ф. Миттельбах, А. Самарин. Перевод с английского Маховой О. А., Третьякова Н. В., Тюменцева Ю. В. и Чистякова В. В. под редакцией Маховой И. А. – М.: Мир, 1999.
13. Копка Н. Guide to L^AT_EX / Н. Копка, P. Daly. – Addison-Wesley, 2004.
14. Lamport L. L^AT_EX. A document preparation system, user's guide and reference manual / L. Lamport. – Addison-Wesley, 1994.
15. Spivak M. The joy of T_EX. A gourmet guide to typesetting with the A_MS-T_EX macro package. – American mathematical society, Providence, RI, 1990. Русский перевод: Спивак М. Восхитительный T_EX: руководство по комфортному изготовлению научных публикаций в пакете A_MS-T_EX / М. Спивак. – М.: Мир, 1993.

УДК 004.021

Коцюба А.Ю., Лавренчук С.В., Остапчук Р.В.
Луцький національний технічний університет

МЕТОДИКА ПОБУДОВИ АЛГОРИТМІВ ГЕНЕРАЦІЇ ВАРІАНТІВ ГОЛОВОЛОМКИ СУДОКУ

Коцюба А.Ю., Лавренчук С.В., Остапчук Р.В. Методика побудови алгоритмів генерації варіантів головоломки Судоку. Однією з найцікавіших задач з комбінаторики є задача про знаходження всіх можливих варіантів гри Судоку та її різновидів. Кількість таких варіантів є числом досить високого порядку, тому найбільш важливою є не так проблема пошуку всіх цих варіантів та їх кількості, як проблема розробки загальної методики побудови алгоритмів генерації різних варіантів головоломки Судоку. Зазвичай, ці алгоритми генерують не всю множину варіантів, а лише деяку її підмножину. При цьому важливими були б знання про всі додаткові обмеження, що призводять до появи цієї підмножини. Розроблена методика є такою, що її можна розвивати для будь-яких різновидів даної гри. Крім цього, перед тим як її застосовувати, необхідно, описати всі вхідні обмеження.

Ключові слова: латинський квадрат Ейлера, головоломка Судоку, задача з комбінаторики, VBA, web-гра Судоку, JavaScript.

Коцюба А.Ю., Лавренчук С.В., Остапчук Р.В. Методика построения алгоритмов генерации вариантов головоломки Судоку. Одной из наиболее интересных задач по комбинаторике является задача о нахождении всех возможных вариантов игры Судоку и ее разновидностей. Количество таких вариантов является числом достаточно высокого порядка, поэтому наиболее важной является не столько проблема поиска всех этих вариантов и их количества, как проблема разработки общей методики построения алгоритмов генерации различных вариантов головоломки Судоку. В большинстве случаев эти алгоритмы генерируют не всё множество вариантов, а только некоторое его подмножество. При этом важными были бы знания обо всех дополнительных ограничениях, которые приводят к появлению этого подмножества. Разработанная методика такова, что ее можно развивать для любых разновидностей данной игры. Кроме этого, перед тем как ее применять, необходимо, описать все входящие ограничения.

Ключевые слова: латинский квадрат Эйлера, головоломка Судоку, задача по комбинаторике, VBA, web-игра Судоку, JavaScript.

Kotsyuba A.Yu, Lavrenchuk S.V., Ostapchuk R.V. The technique of building algorithms generate variations Sudoku puzzles. One of the most interesting problems in combinatorics is the problem of finding all possible Sudoku game and its varieties. The number of such choices is the number of very high order, and most important is not the problem of finding all these options and their number, as the problem of developing a common technique of building algorithms generating various options sudoku puzzles. Typically, these algorithms do not generate a whole set of options, but only some subset of it. It would be important knowledge of all additional restrictions that lead to the emergence of this subset. The method is such that it can be developed for any species of game. In addition, before its use, it is necessary to describe all incoming limitations.

Keywords: latin square Euler, Sudoku puzzle, the problem of combinatorics, VBA, web-game Sudoku, JavaScript.

Вступ. Однією із найбільш популярних головоломок ХХІ ст. є Судоку. Її історія починається з праць відомого швейцарського математика, механіка та фізика Леонарда Ейлера (1707 – 1783 р.р.). У своїх працях він досліджував різновиди “магічних квадратів” з кількістю комірок 9, 16, 25 та 36. Також він займався дослідженням “магічних квадратів”, в яких у рядках та стовпцях не повторюються символи (спочатку це були літери латинського алфавіту). Такі квадрати називалися ще латинськими. У сучасному вигляді головоломку судоку вперше опублікували в 1979 році в журналі Word Games magazine. Автором головоломки був Гарвард Гаріс. Він використав принцип латинського квадрата Ейлера і застосував його в матриці розмірністю 9×9 , додавши при цьому додаткові обмеження – цифри не повинні повторюватися у 9-ти внутрішніх квадратах розмірністю 3×3 . В квітні 1984 року ця головоломка була опублікована видавництвом збірників різних головоломок, японською компанією Nicoly Inc., під заголовком “Число використовується лише один раз”. Назву головоломці дав керівник видавництва – Каї Макі. Su перекладається як число, а доси – як єдине. 12 листопада 2004 року газета The Times вперше на своїх сторінках опублікувала головоломку Судоку. Ця публікація стала сенсацією, головоломка швидко поширилася по всій Британії, Австралії, Новій Зеландії та набула популярності у США.

З'явилося багато різновидів цієї головоломки (кількість найвідоміших сягає 30). Розглянемо коротко деякі з них:

1. Стрічки. У матриці порожніх комірок розмірністю 9×9 якимось (відмінним від білого) кольором, наприклад, сірим зафарбовуюються групи по 7 комірок (рис. 1).

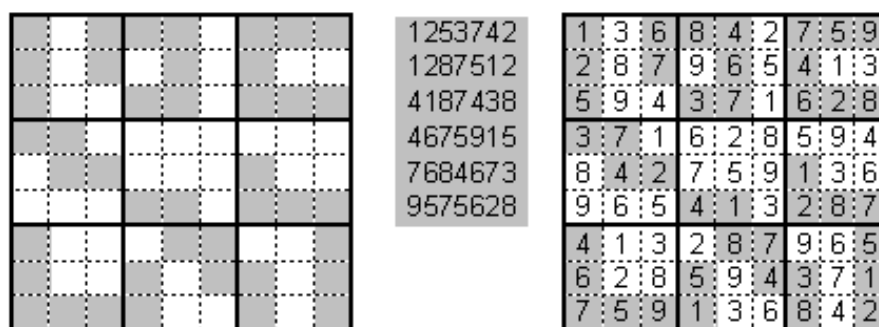


Рис. 1. Різновид головоломки Судоку – “Стрічки”

Дається набір цифр, яким необхідно заповнити зафарбовані комірки. Далі одержується судоку, яке необхідно розв’язати

2. Будівлі. Уявимо, що матриця з 81-ї комірки є скупченням такої ж кількості різноповерхових будинків, розташованих поруч один біля одного. При цьому спостерігач дивиться на ці будівлі зверху і нехай число, написане “на даху будинка”, означає кількість його поверхів. Поблизу комплексу бувель вештаються інші спостерігачі. Обходячи даний комплекс з різних сторін вони можуть бачити лише ті будинки, які не закриті вищим будинком. Відповідно і з’являються числа, які означають кількість видимих з деякої сторони будівель (рис. 2).

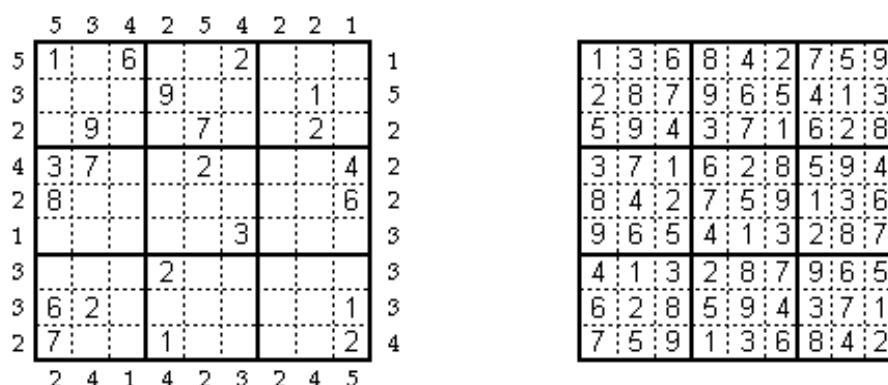


Рис. 2. Різновид головоломки Судоку – “Будівлі”

Всі інші правила заповнення є стандартними для Судоку.

3. Діагональна Судоку. Це різновид Судоку, який зустрічається найчастіше. Для його вирішення використовують стандартні правила для судоку. Відмінність від звичайного полягає лише в тому, що додається ще одне обмеження – цифри не повинні повторюватися по відмічених діагоналях (рис. 3).

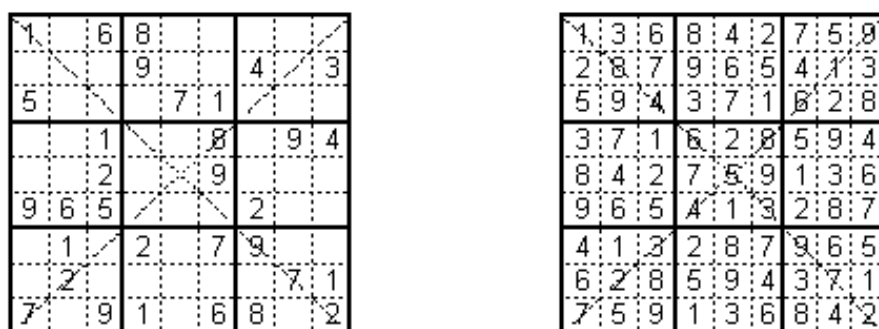


Рис. 3. Різновид головоломки Судоку з додатковими обмеженнями – “Діагональна Судоку”

Це були різновиди Судоку (не всі, а лише, на нашу думку, найвідоміші), в основі яких лежить правило, сформульоване Гарвардом Гарісом (надалі такі різновиди називатимемо класичними різновидами Судоку). Бувають ще й такі різновиди, у яких комірки розташовані взагалі не за

Гарісом (некласичні різновиди). Спільним у всіх цих головоломках залишається лише те, що закладено у їх назві – “Єдине число”.

Постановка наукової проблеми та її значення. З появою таких головоломок, виникла досить цікава, і на нашу думку, не досліджена до кінця проблема генерації різних варіантів головоломки Судоку. Нехай ми маємо справу з самим першим класичним різновидом цієї головоломки, запропонованим Гарвардом Гарісом (класичне судоку). На даний час не існує алгоритму, який би зміг згенерувати довільний варіант з множини всіх можливих варіантів цієї головоломки. Для побудови такого алгоритму необхідно було б спочатку розв’язати одну з нерозв’язаних, на нашу думку, задач з комбінаторики: “Скільки всього є варіантів класичної гри Судоку”. Вирішення цієї задачі дозволить генерувати не лише варіанти цієї гри, а й варіанти класичних різновидів головоломки Судоку таких як: “Стрічки”, “Будівлі” тощо. Крім того якщо розробити загальну методику, яка б дозволяла ще знаходити всі можливі варіанти для випадків класичних різновидів з додатковими обмеженнями – наприклад, діагональна Судоку тощо, то можна було б розпочати роботу над загальним алгоритмом, який дав би змогу розв’язувати відповідні комбінаторні задачі некласичних різновидів Судоку. І як результат – створити програмний комплекс, який би генерував різні варіанти різновидів гри Судоку. Це дало б можливість досліджувати алгоритми розв’язування різновидів Судоку, шукати мінімальну кількість підказок (ключів) не лише для класичної гри Судоку, а й для інших некласичних її різновидів.

Власне розробці такої методики і присвячена дана праця.

Аналіз останніх досліджень та публікацій. Більше дізнатися про латинський квадрат Ейлера можна з електронного ресурсу [1]. У джерелах [2, 3] наведено найвідоміші класичні та некласичні різновиди судоку. Варіант одного з найскладніших судоку наведено у праці [4]. Інформацію про мінімальну кількість підказок (ключів) для того, щоб судоку було задано однозначно можна знайти в електронних ресурсах [5, 6]. Для дослідження методик вирішення судоку можна використати джерела [3, 7–9]. Висновки про сучасні алгоритми генерації варіантів головоломки Судоку можна зробити з робіт [10–12]. Для вивчення технологій написання програм мовою VBA та скриптів мовою JavaScript можна використати праці [13] та [14] відповідно.

Мета роботи – розробка методики написання алгоритмів генерації різних варіантів головоломки Судоку. Слід зауважити, що мету можна розділити на початкову (власне яка і є основною вищесформульованою метою роботи) та кінцеву (єдина проблема, яка стоїть на заваді її практичної реалізації – це обчислювальна швидкодія сучасних персональних комп’ютерів). Кінцевою метою роботи є написання алгоритму генерації всіх можливих варіантів класичної гри Судоку.

Розробка методики. Спочатку теоретично розробимо методику, яка в разі наявності необхідної обчислювальної швидкодії сучасних ПК дозволить розв’язати задачу з комбінаторики: “Скільки всього є варіантів класичної гри Судоку”.

Очевидно, кількість всіх можливих варіантів класичного судоку не можна обчислювати за формулою

$$N = \prod_{i=1}^9 i!. \quad (1)$$

Досить часто в літературі чи в електронних ресурсах [5, 12] помилково вважають, що шукана кількість всіх можливих варіантів класичного судоку сягає порядку 10^{21} (неважко, побачити, що саме такого порядку є величина, що обчислюється за формулою (1)). Очевидно, що додаткові обмеження, які на квадрат Ейлера розмірністю 9×9 наклав Гарвард Гаріс (а саме, цифри не повинні повторюватися у 9-ти внутрішніх квадратах розмірністю 3×3) призведуть до суттєвого зниження порядку шуканої кількості. При цьому формула (1) взагалі ніяк не допоможе. Цікаве рішення цієї задачі наведено у праці [12]

$$2297902829591040 \approx 2.3 \cdot 10^{15}.$$

Наведений результат важко без доведення прийняти на віру, хоча він є досить реалістичним. Навіть, якщо припустити, що ця кількість і є розв'язком вищесформульованої задачі з комбінаторики, то необхідно навчитися генерувати всі можливі варіанти цієї головоломки.

Для того щоб, всі цифри у 9-ти внутрішніх квадратах (назвемо їх блоками (рис. 1–3)) не повторювалися, можна заповнювати Судоку за такою схемою: беремо спочатку першу цифру і розкладаємо її по блоках так, щоб не порушити базові обмеження латинського квадрата (цифри повинні бути різними у кожному рядку та стовпці), далі аналогічно по комірках, які залишилися розкладаємо другу цифру і так далі аж до останньої 9-ї цифри, для якої позиції в блоках вже будуть однозначно визначені. Для більшої наочності подальших викладок введемо позначення для цих цифр, які подані на рис. 4.

a_{11}	a_{21}	a_{31}	a_{12}	a_{22}	a_{32}	a_{13}	a_{23}	a_{33}
a_{41}	a_{51}	a_{61}	a_{42}	a_{52}	a_{62}	a_{43}	a_{53}	a_{63}
a_{71}	a_{81}	a_{91}	a_{72}	a_{82}	a_{92}	a_{73}	a_{83}	a_{93}
a_{14}	a_{24}	a_{34}	a_{15}	a_{25}	a_{35}	a_{16}	a_{26}	a_{36}
a_{44}	a_{54}	a_{64}	a_{45}	a_{55}	a_{65}	a_{46}	a_{56}	a_{66}
a_{74}	a_{84}	a_{94}	a_{75}	a_{85}	a_{95}	a_{76}	a_{86}	a_{96}
a_{17}	a_{27}	a_{37}	a_{18}	a_{28}	a_{38}	a_{19}	a_{29}	a_{39}
a_{47}	a_{57}	a_{67}	a_{48}	a_{58}	a_{68}	a_{49}	a_{59}	a_{69}
a_{77}	a_{87}	a_{97}	a_{78}	a_{88}	a_{98}	a_{79}	a_{89}	a_{99}

Рис. 4. Позначення для заповненого варіанту судоку

Згідно з наведеними на рис. 4 позначеннями для того, щоб в кожному блоці зустрічалася лише одна цифра, необхідно кожній цифрі поставити у відповідність деяке правило

$$(a_{k_1 1}, a_{k_2 2}, \dots, a_{k_9 9}), \quad (2)$$

у якому $k_1, k_2, \dots, k_9 = \overline{1, 9}$. Сформулюємо ще такі умови на числа k_1, k_2, \dots, k_9 , щоб виконувалися базові обмеження (надалі так називатимемо обмеження, які були характерні ще для латинського квадрата Ейлера). Якщо множину цифр від 1 до 9 розбити на 3 підмножини: $R_1 = \{1, 2, 3\}$, $R_2 = \{4, 5, 6\}$ та $R_3 = \{7, 8, 9\}$; то першу умову (цифри повинні бути різними у кожному рядку) можна сформулювати наступним чином: для будь-яких кортежів (k_i, k_j) таких, що $(i, j \in R_1) \vee (i, j \in R_2) \vee (i, j \in R_3)$ повинна виконуватися умова $(k_i, k_j \notin R_1) \wedge (k_i, k_j \notin R_2) \wedge (k_i, k_j \notin R_3)$. Для формулювання відповідної другої умови множину цифр від 1 до 9 необхідно розбити на 3 інші підмножини: $C_1 = \{1, 4, 7\}$, $C_2 = \{2, 5, 8\}$ та $C_3 = \{3, 6, 9\}$. Далі для будь-яких кортежів (k_i, k_j) друга умова формулюється аналогічно. В середовищі MS Excel за допомогою VBA неважко написати макрос, який би видав всі можливі правила (2), які у спрощеній формі легше буде записати у вигляді

$$(k_1, k_2, \dots, k_9). \quad (2')$$

Покажемо два фрагменти коду відповідного макросу (рис. 5).

```

1 Sub Nabors_spivpad()
2 '
3 ' Nabors_spivpad Макрос
4 ' Макрос записан 16.10.2015 (SPA)
5 '
6 ' Сочетание клавиш: Ctrl+d
7 '
8 Dim lich As Integer
9 Dim i1 As Integer
10 Dim i2 As Integer
11 Dim i3 As Integer
12 Dim i4 As Integer
13 Dim i5 As Integer
14 Dim i6 As Integer
15 Dim i7 As Integer
16 Dim i8 As Integer
17 Dim i9 As Integer
18 lich = -32768
19 For i1 = 1 To 9
20   For i2 = 1 To 9
21     For i3 = 1 To 9
22       For i4 = 1 To 9
23         For i5 = 1 To 9
24           For i6 = 1 To 9
25             For i7 = 1 To 9
26               For i8 = 1 To 9
27                 For i9 = 1 To 9
28                   If ((i2 = 2) And (i3 = 3)) Then
29                     ElseIf ((i2 = 3) And (i3 = 2)) Then
30                     ElseIf ((i2 = 1) And (i3 = 1)) Then
31                     ElseIf ((i2 = 1) And (i3 = 3)) Then
32                     ElseIf ((i2 = 3) And (i3 = 1)) Then
33                     ElseIf ((i2 = 2) And (i3 = 2)) Then
34                     ElseIf ((i2 = 1) And (i3 = 2)) Then
505                   ElseIf ((i3 = 6) And (i6 = 9)) Then
506                   ElseIf ((i3 = 9) And (i6 = 6)) Then
507                   ElseIf ((i3 = 3) And (i6 = 3)) Then
508                   ElseIf ((i3 = 3) And (i6 = 9)) Then
509                   ElseIf ((i3 = 9) And (i6 = 3)) Then
510                   ElseIf ((i3 = 6) And (i6 = 6)) Then
511                   ElseIf ((i3 = 3) And (i6 = 6)) Then
512                   ElseIf ((i3 = 6) And (i6 = 3)) Then
513                   ElseIf ((i3 = 9) And (i6 = 9)) Then
514                   Else
515                     lich = lich + 1
516                     Cells(lich + 32768, 1) = i1
517                     Cells(lich + 32768, 2) = i2
518                     Cells(lich + 32768, 3) = i3
519                     Cells(lich + 32768, 4) = i4
520                     Cells(lich + 32768, 5) = i5
521                     Cells(lich + 32768, 6) = i6
522                     Cells(lich + 32768, 7) = i7
523                     Cells(lich + 32768, 8) = i8
524                     Cells(lich + 32768, 9) = i9
525                   End If
526                 Next i9
527               Next i8
528             Next i7
529           Next i6
530         Next i5
531       Next i4
532     Next i3
533   Next i2
534   Next i1
535 End Sub

```

Рис. 5. Код макросу (початок і кінець), який видає всі можливі правила у вигляді (2')

Тепер залишається показати також у скороченому вигляді відповідний результат (рис. 6).

	A	B	C	D	E	F	G	H	I	J
1	1	4	7	2	5	8	3	6	9	
2	2	1	4	7	2	5	8	3	9	6
3	3	1	4	7	2	5	8	6	3	9
4	4	1	4	7	2	5	8	6	9	3
5	5	1	4	7	2	5	8	9	3	6
6	6	1	4	7	2	5	8	9	6	3
7	7	1	4	7	2	5	9	3	6	8
8	8	1	4	7	2	5	9	3	9	5
9	9	1	4	7	2	5	9	6	3	8
10	10	1	4	7	2	5	9	6	9	2
11	11	1	4	7	2	5	9	9	3	5
12	12	1	4	7	2	5	9	9	6	2
13	13	1	4	7	2	6	8	3	5	9
14	14	1	4	7	2	6	8	3	8	6
15	15	1	4	7	2	6	8	6	2	9
16	16	1	4	7	2	6	8	6	8	3
17	17	1	4	7	2	6	8	9	2	6
18	18	1	4	7	2	6	8	9	5	3
19	19	1	4	7	2	6	9	3	5	8
20	20	1	4	7	2	6	9	3	8	5
21	21	1	4	7	2	6	9	6	2	8

	A	B	C	D	E	F	G	H	I	J
46636	46636	9	6	3	8	4	1	4	8	2
46637	46637	9	6	3	8	4	1	7	2	5
46638	46638	9	6	3	8	4	1	7	5	2
46639	46639	9	6	3	8	4	2	1	5	7
46640	46640	9	6	3	8	4	2	1	8	4
46641	46641	9	6	3	8	4	2	4	2	7
46642	46642	9	6	3	8	4	2	4	8	1
46643	46643	9	6	3	8	4	2	7	2	4
46644	46644	9	6	3	8	4	2	7	5	1
46645	46645	9	6	3	8	5	1	1	4	8
46646	46646	9	6	3	8	5	1	1	7	5
46647	46647	9	6	3	8	5	1	4	1	8
46648	46648	9	6	3	8	5	1	4	7	2
46649	46649	9	6	3	8	5	1	7	1	5
46650	46650	9	6	3	8	5	1	7	4	2
46651	46651	9	6	3	8	5	2	1	4	7
46652	46652	9	6	3	8	5	2	1	7	4
46653	46653	9	6	3	8	5	2	4	1	7
46654	46654	9	6	3	8	5	2	4	7	1
46655	46655	9	6	3	8	5	2	7	1	4
46656	46656	9	6	3	8	5	2	7	4	1

Рис. 6. Результат роботи вищенаведеного макросу

З цього рисунку очевидно, що всього є 46656 правил виду (2'), за допомогою яких можна скласти всі варіанти класичного sudoku. При складанні варіантів sudoku слід зазначити, що не всі правила можна поєднувати. Тут мається на увазі, що якщо першій цифрі поставити у відповідність одне з 46656 правил, то другій цифрі можна ставити у відповідність будь-яке правило з множини цих правил, яке "немає нічого спільного з попереднім правилом". А тепер пояснимо, що означає вислів взятий у лапки.

Означення 1. Два правила виду (2') будемо називати *можливими сусідами* або такими, що *не мають нічого спільного*, якщо на однакових позиціях не має однакових цифр. Такі правила надалі коротко називатимемо *сусідними*.

Очевидно, що якщо скласти матрицю розмірністю 9×9 таку, щоб номер рядка був би, наприклад, цифрою, а номер стовпця – позицією, то в кожну комірку такої матриці можна було б записати множини всіх можливих правил, у яких на відповідній позиції зустрічається відповідна

цифра. І якщо потужності цих множин будуть однакові, то можна припустити, що програма дала правильний результат. Неважко методом від супротивного показати, що цей критерій є необхідною умовою повноти шуканої множини правил. І для вищеписаних 46656 правил виду (2'), було показано, що такі потужності в кожній комірці матриці розмірності 9×9 є однаковими і становлять $46656/9 = 5184$.

Для того, щоб знайти кількість всіх можливих варіантів класичного sudoku, нам необхідно знайти множину всіх множин (наборів) з 9 правил, які попарно є можливими сусідами. Якщо потужність такої множини позначимо, через N_{nab} , то, очевидно, розв'язок розглянутої задачі з комбінаторики буде знаходитися за формулою

$$N_{sudocu} = N_{nab} \cdot 9!, \quad (3)$$

тобто для генерації всіх можливих варіантів ігор, в програму необхідно внести:

- 46656 правил (2'), частина з яких нами наведена на рис. 6;
- масив, елементами, якого є різні відсортовані в порядку, наприклад, зростання набори (масиви) з 9-ти правил, кожні два з яких є можливими сусідами (цей масив повинен містити N_{nab} наборів).

Якщо припустити, що наведена у праці [12] кількість $N_{sudocu} = 2297902829591040$, то N_{nab} не буде натуральним числом, тобто можна вважати, що вищесформульована задача з комбінаторики не є розв'язаною.

Далі залишається лише за допомогою ГПВЧ згенерувати 2 числа:

- перше число (з множини $\{0, 1, \dots, N_{nab} - 1\}$) буде задавати набір;
- друге число (з множини $\{0, 1, \dots, 9! - 1\}$) буде задавати порядок, тобто номер правила для цифри 1, номер правила для цифри 2 і т.д.

Отже, розв'язання даної комбінаторної задачі зводиться лише до пошуку всіх можливих різних та відсортованих наборів із сусідніх правил та підрахунку кількості цих наборів.

Для вирішення цієї проблеми пропонуємо наступний алгоритм:

1 етап. Відсортуємо та пронумеруємо правила (як на рис. 6). Розіб'ємо їх (надалі набір будемо ототожнювати з номером, який йому відповідає) на 9 підмножин, таких, що в них жодні два правила не можуть бути сусідніми. Зробимо це, наприклад, по порядку

$$\begin{aligned} &\{1, 2, \dots, 5184\}, \{5185, 5186, \dots, 10368\}, \{10369, 10370, \dots, 15552\}, \\ &\{15553, 15554, \dots, 20736\}, \{20737, 20738, \dots, 25920\}, \{25921, 25922, \dots, 31104\}, \\ &\{31105, 31106, \dots, 36288\}, \{36289, 36290, \dots, 41472\}, \{41473, 41474, \dots, 46656\}. \end{aligned} \quad (4)$$

2 етап. Для кожного правила внесемо масив всіх можливих, наприклад, несусідніх по відношенню до нього правил. Можна брати і масив сусідніх правил, просто при цьому дещо зміняться наступні етапи алгоритму. Доречі, для даної задачі, виписати всіх сусідів, чи не сусідів, досить трудомісткий процес навіть для потужних ПК. Якщо з множини всіх несусідніх правил виключити ті правила, які входять в одну і ту ж підмножину (4), то одержимо, що для кожного фіксованого правила буде 23500 несусідніх правил. А тепер уявіть матрицю розмірності 46656×23500 чисел від 1 до 46656. Хоча визначення елементів даного масиву може зайняти якийсь час, це не найбільша проблема, яка стоїть на заваді пошуку відсортованих різних наборів.

3 етап. Далі залишається перебирати правила з першої підмножини (4). При кожному такому виборі перебирати правила з другої підмножини (4) і т.д. Для кожних з 9-ти вибраних правил попарно перевіряти, чи може серед них знайтися хоча б 2 несусідніх правила (для перевірки використовуємо вищесформований масив несусідніх правил). Якщо несусідніх правил немає, то такий набір записуємо, в протилежному випадку рухаємося далі.

Розбиття на підмножини (4) має свої вагомі переваги. Одержані в результаті перебору набори автоматично стають відсортованими і як результат – не може з'явитися ще один набір рівний (з точністю до перемішування) якомусь набору, який був уже знайдений.

Виникає логічне запитання: яка проблема не дозволить програмно реалізувати вищеописаний алгоритм? А проблема полягає у тому, що для повного перебору необхідно проробити $5184^9 \cdot 23500$ операцій. Залишається лише справа за малим: вибрати середовище та обчислювальні технології, які б дозволили за реальний проміжок часу проробити таку кількість операцій.

Апробація розробленої методики. У зв'язку з вищеописаними проблемами, нами було вирішено накласти додаткові обмеження: вимоги для виконання базових обмежень перемішати і додати обмеження про те, що k_1, k_2, \dots, k_9 є різними, тобто додати ще три умови:

- для будь-яких кортежів (k_i, k_j) таких, що $(i, j \in R_1) \vee (i, j \in R_2) \vee (i, j \in R_3)$ повинна виконуватися умова $(k_i, k_j \notin C_1) \wedge (k_i, k_j \notin C_2) \wedge (k_i, k_j \notin C_3)$;
- для будь-яких кортежів (k_i, k_j) таких, що $(i, j \in C_1) \vee (i, j \in C_2) \vee (i, j \in C_3)$ повинна виконуватися умова $(k_i, k_j \notin R_1) \wedge (k_i, k_j \notin R_2) \wedge (k_i, k_j \notin R_3)$;
- $k_1 \neq k_2 \neq \dots \neq k_9$.

В результаті таких обмежень одержимо 144 правила. На першому етапі алгоритму множини правил розіб'ємо аналогічно до сукупності (4) на такі підмножини:

$$\begin{aligned} & \{1, 2, \dots, 16\}, \{17, 18, \dots, 32\}, \{33, 34, \dots, 48\}, \\ & \{49, 50, \dots, 64\}, \{65, 66, \dots, 80\}, \{81, 82, \dots, 96\}, \\ & \{97, 98, \dots, 112\}, \{113, 114, \dots, 128\}, \{129, 130, \dots, 144\}. \end{aligned} \quad (4')$$

Застосувавши вищеописаний алгоритм до множини цих правил та до сукупності підмножин (4'), одержимо 664 набори. За формулою (3) неважко порахувати, що таким чином, можна побудувати алгоритм, який генеруватиме $664 \cdot 9! = 240952320$ варіантів гри класичного sudoku. Що і було зроблено засобами web-програмування. Покажемо фрагмент коду мовою JavaScript, у якому введено 144 правила та 664 набори з 9-ти сусідніх правил (рис. 7).

```
var pravilos=new Array(144);
pravilos[0]=[1,5,9,2,6,7,3,4,8]; pravilos[1]=[1,5,9,3,4,8,2,6,7]; pravilos[2]=[1,5,9,6,7,2,8,3,4]; pravilos[3]=[1,5,9,8,3,4,6,7,2]; pravilos[4]=[1,6,8,2,4,9,3,5,7]; pravilos[5]
pravilos[15]=[2,4,9,1,6,8,3,5,7]; pravilos[16]=[2,4,9,3,5,7,1,6,8]; pravilos[17]=[2,4,9,6,8,1,7,3,5]; pravilos[18]=[2,4,9,7,3,5,6,8,1]; pravilos[19]=[2,4,9,7,3,5,6,8,1]; pravilos[20]=[2,6,7,1,5,9,3,4,8]; pravilos[21]=[2,6,7,1,5,9,3,4,8]; pravilos[22]=[2,6,7,1,5,9,3,4,8];
pravilos[32]=[3,4,8,1,5,9,2,6,7]; pravilos[33]=[3,4,8,2,6,7,1,5,9]; pravilos[34]=[3,4,8,5,9,1,7,2,6]; pravilos[35]=[3,4,8,7,2,6,5,9,1]; pravilos[36]=[3,5,7,1,6,8,2,4,9]; pravilos[37]=[3,5,7,1,6,8,2,4,9];
pravilos[40]=[4,2,9,3,7,5,8,6,1]; pravilos[41]=[4,2,9,5,3,7,6,1,8]; pravilos[42]=[4,2,9,6,1,8,5,3,7]; pravilos[43]=[4,2,9,6,1,8,5,3,7]; pravilos[44]=[4,2,9,6,1,8,5,3,7]; pravilos[45]=[4,2,9,6,1,8,5,3,7];
pravilos[46]=[4,2,9,6,1,8,5,3,7]; pravilos[47]=[4,2,9,6,1,8,5,3,7]; pravilos[48]=[4,2,9,6,1,8,5,3,7]; pravilos[49]=[4,2,9,6,1,8,5,3,7]; pravilos[50]=[4,2,9,6,1,8,5,3,7]; pravilos[51]=[4,2,9,6,1,8,5,3,7];
pravilos[52]=[4,2,9,6,1,8,5,3,7]; pravilos[53]=[4,2,9,6,1,8,5,3,7]; pravilos[54]=[4,2,9,6,1,8,5,3,7]; pravilos[55]=[4,2,9,6,1,8,5,3,7]; pravilos[56]=[4,2,9,6,1,8,5,3,7]; pravilos[57]=[4,2,9,6,1,8,5,3,7];
pravilos[58]=[4,2,9,6,1,8,5,3,7]; pravilos[59]=[4,2,9,6,1,8,5,3,7]; pravilos[60]=[4,2,9,6,1,8,5,3,7]; pravilos[61]=[4,2,9,6,1,8,5,3,7]; pravilos[62]=[4,2,9,6,1,8,5,3,7]; pravilos[63]=[4,2,9,6,1,8,5,3,7];
pravilos[64]=[4,2,9,6,1,8,5,3,7]; pravilos[65]=[4,2,9,6,1,8,5,3,7]; pravilos[66]=[4,2,9,6,1,8,5,3,7]; pravilos[67]=[4,2,9,6,1,8,5,3,7]; pravilos[68]=[4,2,9,6,1,8,5,3,7]; pravilos[69]=[4,2,9,6,1,8,5,3,7];
pravilos[70]=[4,2,9,6,1,8,5,3,7]; pravilos[71]=[4,2,9,6,1,8,5,3,7]; pravilos[72]=[4,2,9,6,1,8,5,3,7]; pravilos[73]=[4,2,9,6,1,8,5,3,7]; pravilos[74]=[4,2,9,6,1,8,5,3,7]; pravilos[75]=[4,2,9,6,1,8,5,3,7];
pravilos[76]=[4,2,9,6,1,8,5,3,7]; pravilos[77]=[4,2,9,6,1,8,5,3,7]; pravilos[78]=[4,2,9,6,1,8,5,3,7]; pravilos[79]=[4,2,9,6,1,8,5,3,7]; pravilos[80]=[4,2,9,6,1,8,5,3,7]; pravilos[81]=[4,2,9,6,1,8,5,3,7];
pravilos[82]=[4,2,9,6,1,8,5,3,7]; pravilos[83]=[4,2,9,6,1,8,5,3,7]; pravilos[84]=[4,2,9,6,1,8,5,3,7]; pravilos[85]=[4,2,9,6,1,8,5,3,7]; pravilos[86]=[4,2,9,6,1,8,5,3,7]; pravilos[87]=[4,2,9,6,1,8,5,3,7];
pravilos[88]=[4,2,9,6,1,8,5,3,7]; pravilos[89]=[4,2,9,6,1,8,5,3,7]; pravilos[90]=[4,2,9,6,1,8,5,3,7]; pravilos[91]=[4,2,9,6,1,8,5,3,7]; pravilos[92]=[4,2,9,6,1,8,5,3,7]; pravilos[93]=[4,2,9,6,1,8,5,3,7];
pravilos[94]=[4,2,9,6,1,8,5,3,7]; pravilos[95]=[4,2,9,6,1,8,5,3,7]; pravilos[96]=[4,2,9,6,1,8,5,3,7]; pravilos[97]=[4,2,9,6,1,8,5,3,7]; pravilos[98]=[4,2,9,6,1,8,5,3,7]; pravilos[99]=[4,2,9,6,1,8,5,3,7];
pravilos[100]=[4,2,9,6,1,8,5,3,7]; pravilos[101]=[4,2,9,6,1,8,5,3,7]; pravilos[102]=[4,2,9,6,1,8,5,3,7]; pravilos[103]=[4,2,9,6,1,8,5,3,7]; pravilos[104]=[4,2,9,6,1,8,5,3,7]; pravilos[105]=[4,2,9,6,1,8,5,3,7];
pravilos[106]=[4,2,9,6,1,8,5,3,7]; pravilos[107]=[4,2,9,6,1,8,5,3,7]; pravilos[108]=[4,2,9,6,1,8,5,3,7]; pravilos[109]=[4,2,9,6,1,8,5,3,7]; pravilos[110]=[4,2,9,6,1,8,5,3,7]; pravilos[111]=[4,2,9,6,1,8,5,3,7];
pravilos[112]=[4,2,9,6,1,8,5,3,7]; pravilos[113]=[4,2,9,6,1,8,5,3,7]; pravilos[114]=[4,2,9,6,1,8,5,3,7]; pravilos[115]=[4,2,9,6,1,8,5,3,7]; pravilos[116]=[4,2,9,6,1,8,5,3,7]; pravilos[117]=[4,2,9,6,1,8,5,3,7];
pravilos[118]=[4,2,9,6,1,8,5,3,7]; pravilos[119]=[4,2,9,6,1,8,5,3,7]; pravilos[120]=[4,2,9,6,1,8,5,3,7]; pravilos[121]=[4,2,9,6,1,8,5,3,7]; pravilos[122]=[4,2,9,6,1,8,5,3,7]; pravilos[123]=[4,2,9,6,1,8,5,3,7];
pravilos[124]=[4,2,9,6,1,8,5,3,7]; pravilos[125]=[4,2,9,6,1,8,5,3,7]; pravilos[126]=[4,2,9,6,1,8,5,3,7]; pravilos[127]=[4,2,9,6,1,8,5,3,7]; pravilos[128]=[4,2,9,6,1,8,5,3,7]; pravilos[129]=[4,2,9,6,1,8,5,3,7];
pravilos[130]=[4,2,9,6,1,8,5,3,7]; pravilos[131]=[4,2,9,6,1,8,5,3,7]; pravilos[132]=[4,2,9,6,1,8,5,3,7]; pravilos[133]=[4,2,9,6,1,8,5,3,7]; pravilos[134]=[4,2,9,6,1,8,5,3,7]; pravilos[135]=[4,2,9,6,1,8,5,3,7];
pravilos[136]=[4,2,9,6,1,8,5,3,7]; pravilos[137]=[4,2,9,6,1,8,5,3,7]; pravilos[138]=[4,2,9,6,1,8,5,3,7]; pravilos[139]=[4,2,9,6,1,8,5,3,7]; pravilos[140]=[4,2,9,6,1,8,5,3,7]; pravilos[141]=[4,2,9,6,1,8,5,3,7];
pravilos[142]=[4,2,9,6,1,8,5,3,7]; pravilos[143]=[4,2,9,6,1,8,5,3,7]; pravilos[144]=[4,2,9,6,1,8,5,3,7];
var nabors=new Array(664);
nabors[0]=[1,22,33,58,79,90,99,120,131]; nabors[1]=[1,22,33,58,79,90,100,119,132]; nabors[2]=[1,22,33,58,79,90,103,116,135]; nabors[3]=[1,22,33,58,79,90,104,115,136]; nabors[4]=[1,22,33,58,79,90,105,114,137];
nabors[5]=[1,22,33,58,79,90,106,113,138]; nabors[6]=[1,22,33,58,79,90,107,112,139]; nabors[7]=[1,22,33,58,79,90,108,111,140]; nabors[8]=[1,22,33,58,79,90,109,110,141]; nabors[9]=[1,22,33,58,79,90,110,109,142];
nabors[10]=[1,22,33,58,79,90,111,108,143]; nabors[11]=[1,22,33,58,79,90,112,107,144]; nabors[12]=[1,22,33,58,79,90,113,106,145]; nabors[13]=[1,22,33,58,79,90,114,105,146]; nabors[14]=[1,22,33,58,79,90,115,104,147];
nabors[15]=[1,22,33,58,79,90,116,103,148]; nabors[16]=[1,22,33,58,79,90,117,102,149]; nabors[17]=[1,22,33,58,79,90,118,101,150]; nabors[18]=[1,22,33,58,79,90,119,100,151]; nabors[19]=[1,22,33,58,79,90,120,99,152];
nabors[20]=[1,22,33,58,79,90,121,98,153]; nabors[21]=[1,22,33,58,79,90,122,97,154]; nabors[22]=[1,22,33,58,79,90,123,96,155]; nabors[23]=[1,22,33,58,79,90,124,95,156]; nabors[24]=[1,22,33,58,79,90,125,94,157];
nabors[25]=[1,22,33,58,79,90,126,93,158]; nabors[26]=[1,22,33,58,79,90,127,92,159]; nabors[27]=[1,22,33,58,79,90,128,91,160]; nabors[28]=[1,22,33,58,79,90,129,90,161]; nabors[29]=[1,22,33,58,79,90,130,89,162];
nabors[30]=[1,22,33,58,79,90,131,88,163]; nabors[31]=[1,22,33,58,79,90,132,87,164]; nabors[32]=[1,22,33,58,79,90,133,86,165]; nabors[33]=[1,22,33,58,79,90,134,85,166]; nabors[34]=[1,22,33,58,79,90,135,84,167];
nabors[35]=[1,22,33,58,79,90,136,83,168]; nabors[36]=[1,22,33,58,79,90,137,82,169]; nabors[37]=[1,22,33,58,79,90,138,81,170]; nabors[38]=[1,22,33,58,79,90,139,80,171]; nabors[39]=[1,22,33,58,79,90,140,79,172];
nabors[40]=[1,22,33,58,79,90,141,78,173]; nabors[41]=[1,22,33,58,79,90,142,77,174]; nabors[42]=[1,22,33,58,79,90,143,76,175]; nabors[43]=[1,22,33,58,79,90,144,75,176]; nabors[44]=[1,22,33,58,79,90,145,74,177];
nabors[45]=[1,22,33,58,79,90,146,73,178]; nabors[46]=[1,22,33,58,79,90,147,72,179]; nabors[47]=[1,22,33,58,79,90,148,71,180]; nabors[48]=[1,22,33,58,79,90,149,70,181]; nabors[49]=[1,22,33,58,79,90,150,69,182];
nabors[50]=[1,22,33,58,79,90,151,68,183]; nabors[51]=[1,22,33,58,79,90,152,67,184]; nabors[52]=[1,22,33,58,79,90,153,66,185]; nabors[53]=[1,22,33,58,79,90,154,65,186]; nabors[54]=[1,22,33,58,79,90,155,64,187];
nabors[55]=[1,22,33,58,79,90,156,63,188]; nabors[56]=[1,22,33,58,79,90,157,62,189]; nabors[57]=[1,22,33,58,79,90,158,61,190]; nabors[58]=[1,22,33,58,79,90,159,60,191]; nabors[59]=[1,22,33,58,79,90,160,59,192];
nabors[60]=[1,22,33,58,79,90,161,58,193]; nabors[61]=[1,22,33,58,79,90,162,57,194]; nabors[62]=[1,22,33,58,79,90,163,56,195]; nabors[63]=[1,22,33,58,79,90,164,55,196]; nabors[64]=[1,22,33,58,79,90,165,54,197];
nabors[65]=[1,22,33,58,79,90,166,53,198]; nabors[66]=[1,22,33,58,79,90,167,52,199]; nabors[67]=[1,22,33,58,79,90,168,51,200]; nabors[68]=[1,22,33,58,79,90,169,50,201]; nabors[69]=[1,22,33,58,79,90,170,49,202];
nabors[70]=[1,22,33,58,79,90,171,48,203]; nabors[71]=[1,22,33,58,79,90,172,47,204]; nabors[72]=[1,22,33,58,79,90,173,46,205]; nabors[73]=[1,22,33,58,79,90,174,45,206]; nabors[74]=[1,22,33,58,79,90,175,44,207];
nabors[75]=[1,22,33,58,79,90,176,43,208]; nabors[76]=[1,22,33,58,79,90,177,42,209]; nabors[77]=[1,22,33,58,79,90,178,41,210]; nabors[78]=[1,22,33,58,79,90,179,40,211]; nabors[79]=[1,22,33,58,79,90,180,39,212];
nabors[80]=[1,22,33,58,79,90,181,38,213]; nabors[81]=[1,22,33,58,79,90,182,37,214]; nabors[82]=[1,22,33,58,79,90,183,36,215]; nabors[83]=[1,22,33,58,79,90,184,35,216]; nabors[84]=[1,22,33,58,79,90,185,34,217];
nabors[85]=[1,22,33,58,79,90,186,33,218]; nabors[86]=[1,22,33,58,79,90,187,32,219]; nabors[87]=[1,22,33,58,79,90,188,31,220]; nabors[88]=[1,22,33,58,79,90,189,30,221]; nabors[89]=[1,22,33,58,79,90,190,29,222];
nabors[90]=[1,22,33,58,79,90,191,28,223]; nabors[91]=[1,22,33,58,79,90,192,27,224]; nabors[92]=[1,22,33,58,79,90,193,26,225]; nabors[93]=[1,22,33,58,79,90,194,25,226]; nabors[94]=[1,22,33,58,79,90,195,24,227];
nabors[95]=[1,22,33,58,79,90,196,23,228]; nabors[96]=[1,22,33,58,79,90,197,22,229]; nabors[97]=[1,22,33,58,79,90,198,21,230]; nabors[98]=[1,22,33,58,79,90,199,20,231]; nabors[99]=[1,22,33,58,79,90,200,19,232];
nabors[100]=[1,22,33,58,79,90,201,18,233]; nabors[101]=[1,22,33,58,79,90,202,17,234]; nabors[102]=[1,22,33,58,79,90,203,16,235]; nabors[103]=[1,22,33,58,79,90,204,15,236]; nabors[104]=[1,22,33,58,79,90,205,14,237];
nabors[105]=[1,22,33,58,79,90,206,13,238]; nabors[106]=[1,22,33,58,79,90,207,12,239]; nabors[107]=[1,22,33,58,79,90,208,11,240]; nabors[108]=[1,22,33,58,79,90,209,10,241]; nabors[109]=[1,22,33,58,79,90,210,9,242];
nabors[110]=[1,22,33,58,79,90,211,8,243]; nabors[111]=[1,22,33,58,79,90,212,7,244]; nabors[112]=[1,22,33,58,79,90,213,6,245]; nabors[113]=[1,22,33,58,79,90,214,5,246]; nabors[114]=[1,22,33,58,79,90,215,4,247];
nabors[115]=[1,22,33,58,79,90,216,3,248]; nabors[116]=[1,22,33,58,79,90,217,2,249]; nabors[117]=[1,22,33,58,79,90,218,1,250]; nabors[118]=[1,22,33,58,79,90,219,0,251]; nabors[119]=[1,22,33,58,79,90,220,0,252];
nabors[120]=[1,22,33,58,79,90,221,0,253]; nabors[121]=[1,22,33,58,79,90,222,0,254]; nabors[122]=[1,22,33,58,79,90,223,0,255]; nabors[123]=[1,22,33,58,79,90,224,0,256]; nabors[124]=[1,22,33,58,79,90,225,0,257];
nabors[125]=[1,22,33,58,79,90,226,0,258]; nabors[126]=[1,22,33,58,79,90,227,0,259]; nabors[127]=[1,22,33,58,79,90,228,0,260]; nabors[128]=[1,22,33,58,79,90,229,0,261]; nabors[129]=[1,22,33,58,79,90,230,0,262];
nabors[130]=[1,22,33,58,79,90,231,0,263]; nabors[131]=[1,22,33,58,79,90,232,0,264]; nabors[132]=[1,22,33,58,79,90,233,0,265]; nabors[133]=[1,22,33,58,79,90,234,0,266]; nabors[134]=[1,22,33,58,79,90,235,0,267];
nabors[135]=[1,22,33,58,79,90,236,0,268]; nabors[136]=[1,22,33,58,79,90,237,0,269]; nabors[137]=[1,22,33,58,79,90,238,0,270]; nabors[138]=[1,22,33,58,79,90,239,0,271]; nabors[139]=[1,22,33,58,79,90,240,0,272];
nabors[140]=[1,22,33,58,79,90,241,0,273]; nabors[141]=[1,22,33,58,79,90,242,0,274]; nabors[142]=[1,22,33,58,79,90,243,0,275]; nabors[143]=[1,22,33,58,79,90,244,0,276]; nabors[144]=[1,22,33,58,79,90,245,0,277];
nabors[145]=[1,22,33,58,79,90,246,0,278]; nabors[146]=[1,22,33,58,79,90,247,0,279]; nabors[147]=[1,22,33,58,79,90,248,0,280]; nabors[148]=[1,22,33,58,79,90,249,0,281]; nabors[149]=[1,22,33,58,79,90,250,0,282];
nabors[150]=[1,22,33,58,79,90,251,0,283]; nabors[151]=[1,22,33,58,79,90,252,0,284]; nabors[152]=[1,22,33,58,79,90,253,0,285]; nabors[153]=[1,22,33,58,79,90,254,0,286]; nabors[154]=[1,22,33,58,79,90,255,0,287];
nabors[155]=[1,22,33,58,79,90,256,0,288]; nabors[156]=[1,22,33,58,79,90,257,0,289]; nabors[157]=[1,22,33,58,79,90,258,0,290]; nabors[158]=[1,22,33,58,79,90,259,0,291]; nabors[159]=[1,22,33,58,79,90,260,0,292];
nabors[160]=[1,22,33,58,79,90,261,0,293]; nabors[161]=[1,22,33,58,79,90,262,0,294]; nabors[162]=[1,22,33,58,79,90,263,0,295]; nabors[163]=[1,22,33,58,79,90,264,0,296]; nabors[164]=[1,22,33,58,79,90,265,0,297];
nabors[165]=[1,22,33,58,79,90,266,0,298]; nabors[166]=[1,22,33,58,79,90,267,0,299]; nabors[167]=[1,22,33,58,79,90,268,0,300]; nabors[168]=[1,22,33,58,79,90,269,0,301]; nabors[169]=[1,22,33,58,79,90,270,0,302];
nabors[170]=[1,22,33,58,79,90,271,0,303]; nabors[171]=[1,22,33,58,79,90,272,0,304]; nabors[172]=[1,22,33,58,79,90,273,0,305]; nabors[173]=[1,22,33,58,79,90,274,0,306]; nabors[174]=[1,22,33,58,79,90,275,0,307];
nabors[175]=[1,22,33,58,79,90,276,0,308]; nabors[176]=[1,22,33,58,79,90,277,0,309]; nabors[177]=[1,22,33,58,79,90,278,0,310]; nabors[178]=[1,22,33,58,79,90,279,0,311]; nabors[179]=[1,22,33,58,79,90,280,0,312];
nabors[180]=[1,22,33,58,79,90,281,0,313]; nabors[181]=[1,22,33,58,79,90,282,0,314]; nabors[182]=[1,22,33,58,79,90,283,0,315]; nabors[183]=[1,22,33,58,79,90,284,0,316]; nabors[184]=[1,22,33,58,79,90,285,0,317];
nabors[185]=[1,22,33,58,79,90,286,0,318]; nabors[186]=[1,22,33,58,79,90,287,0,319]; nabors[187]=[1,22,33,58,79,90,288,0,320]; nabors[188]=[1,22,33,58,79,90,289,0,321]; nabors[189]=[1,22,33,58,79,90,290,0,322];
nabors[190]=[1,22,33,58,79,90,291,0,323]; nabors[191]=[1,22,33,58,79,90,292,0,324]; nabors[192]=[1,22,33,58,79,90,293,0,325]; nabors[193]=[1,22,33,58,79,90,294,0,326]; nabors[194]=[1,22,33,58,79,90,295,0,327];
nabors[195]=[1,22,33,58,79,90,296,0,328]; nabors[196]=[1,22,33,58,79,90,297,0,329]; nabors[197]=[1,22,33,58,79,90,298,0,330]; nabors[198]=[1,22,33,58,79,90,299,0,331]; nabors[199]=[1,22,33,58,79,90,300,0,332];
nabors[200]=[1,22,33,58,79,90,301,0,333]; nabors[201]=[1,22,33,58,79,90,302,0,334]; nabors[202]=[1,22,33,58,79,90,303,0,335]; nabors[203]=[1,22,33,58,79,90,304,0,336]; nabors[204]=[1,22,33,58,79,90,305,0,337];
nabors[205]=[1,22,33,58,79,90,306,0,338]; nabors[206]=[1,22,33,58,79,90,307,0,339]; nabors[207]=[1,22,33,58,79,90,308,0,340]; nabors[208]=[1,22,33,58,79,90,309,0,341]; nabors[209]=[1,22,33,58,79,90,310,0,342];
nabors[210]=[1,22,33,58,79,90,311,0,343]; nabors[211]=[1,22,33,58,79,90,312,0,344]; nabors[212]=[1,22,33,58,79,90,313,0,345]; nabors[213]=[1,22,33,58,79,90,314,0,346]; nabors[214]=[1,22,33,58,79,90,315,0,347];
nabors[215]=[1,22,33,58,79,90,316,0,348]; nabors[216]=[1,22,33,58,79,90,317,0,349]; nabors[217]=[1,22,33,58,79,90,318,0,350]; nabors[218]=[1,22,33,58,79,90,319,0,351]; nabors[219]=[1,22,33,58,79,90,320,0,352];
nabors[220]=[1,22,33,58,79,90,321,0,353]; nabors[221]=[1,22,33,58,79,90,322,0,354]; nabors[222]=[1,22,33,58,79,90,323,0,355]; nabors[223]=[1,22,33,58,79,90,324,0,356]; nabors[224]=[1,22,33,58,79,90,325,0,357];
nabors[225]=[1,22,33,58,79,90,326,0,358]; nabors[226]=[1,22,33,58,79,90,327,0,359]; nabors[227]=[1,22,33,58,79,90,328,0,360]; nabors[228]=[1,22,33,58,79,90,329,0,361]; nabors[229]=[1,22,33,58,79,90,330,0,362];
nabors[230]=[1,22,33,
```


- нехай з множини $\{0,1,...,9!-1\}$ наш ГПВЧ згенерував число 100330.

Тоді з першого згенерованого числа 213 одержимо (із врахуванням зміщення на 1 в індексах) наступні набір та відповідні правила (рис. 8).

```
nabors[213]=[6,17,38,58,79,90,100,119,132];
pravylos[5]= [1,6,8,3,5,7,2,4,9];
pravylos[16]= [2,4,9,1,6,8,3,5,7];
pravylos[37]= [3,5,7,2,4,9,1,6,8];
pravylos[57]= [4,8,3,5,9,1,6,7,2];
pravylos[78]= [5,9,1,6,7,2,4,8,3];
pravylos[89]= [6,7,2,4,8,3,5,9,1];
pravylos[99]= [7,2,6,9,1,5,8,3,4];
pravylos[118]=[8,3,4,7,2,6,9,1,5];
pravylos[131]=[9,1,5,8,3,4,7,2,6];
```

Рис. 8. Випадково згенерований набір та відповідні йому правила

Далі покажемо, що завдяки другому згенерованому числу 100330 кортеж (1,2,3,4,5,6,7,8,9) перетвориться в кортеж (5,8,1,4,2,6,9,7,3), тобто набір (6,17,38,58,79,90,100,119,132) перетвориться в набір (79,119,6,58,17,90,132,100,38). В результаті у відповідній web-реалізації розробленої методики одержимо випадок класичного sudoku (рис. 9).

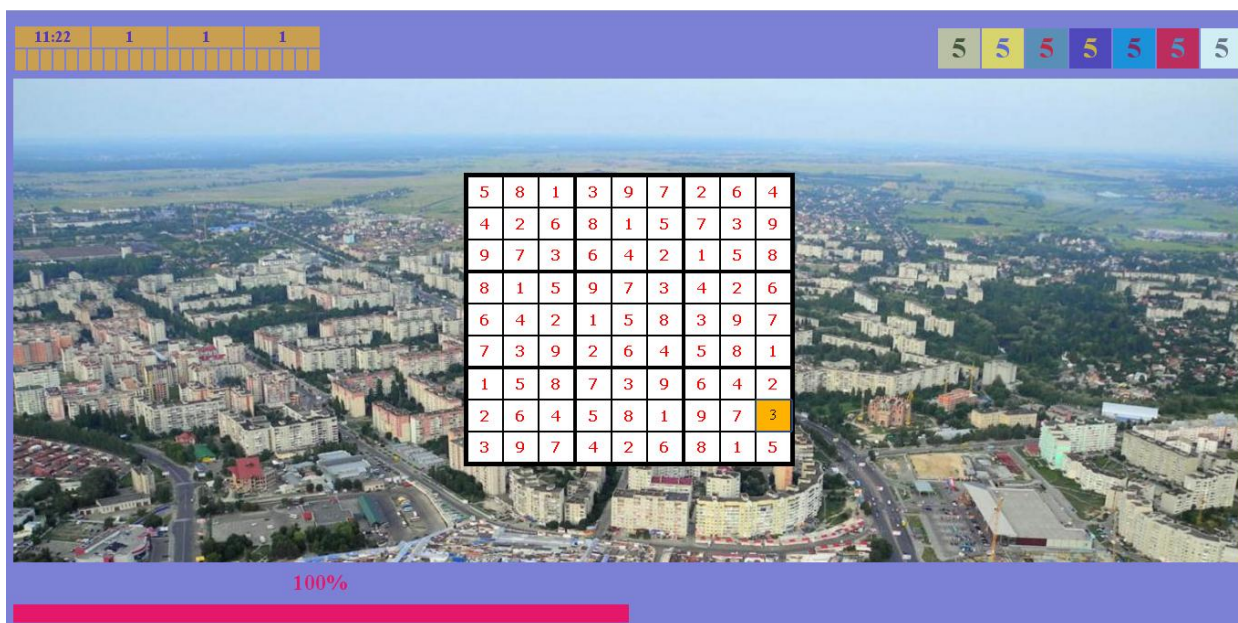


Рис. 9. Приклад web-реалізації розробленої методики генерації варіантів класичної гри Судоку

Далі покажемо, як для вищеописаних параметрів (213;100330) відбувається дві випадкові генерації з 25-ма ключами (рис. 10)

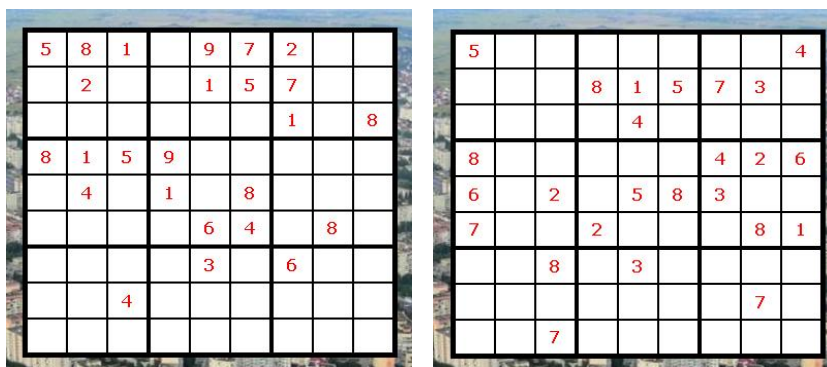


Рис. 10. Два приклади головоломки Судоку, зображеного на рис. 9, з 25-ма ключами

Висновки. Отже, початкову мету роботи реалізовано. Щодо реалізації кінцевої мети, то тут необхідно ще попрацювати в наступних напрямках:

- по-перше, можна накласти не такі сильні обмеження (наприклад, можна спробувати дослідити класичний різновид даної головоломки – діагональне судоку);
- по-друге, у будь-якому випадку необхідно спробувати одні із найшвидших для таких задач мови програмування C++ або Fortran (при цьому важливо навчитися розподіляти операції між ядрами комп'ютерів локальної мережі);
- по-третє, є некласичні різновиди Судоку, вирішення комбінаторної задачі для яких може потребувати значно меншої кількості операцій.

Хоча в даній роботі нами не було знайдено за формулою (3) число N_{sudocu} всіх можливих варіантів класичної гри Судоку, але все ж таки ефективну методику, яку можна розвинути для некласичних різновидів цієї головоломки було розроблено, тобто з основною (не кінцевою) метою роботи справилися.

1. Википедия: Свободная энциклопедия [Електронний ресурс]. – Режим доступу : https://ru.wikipedia.org/wiki/Латинский_квадрат.
2. Википедия: Свободная энциклопедия [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/Судоку>.
3. Судоку-клуб [Електронний ресурс]. – Режим доступу : <http://www.sudoku-club.ru/variations.html>.
4. Попробуйте решить самую трудную головоломку в мире [Електронний ресурс]. – Режим доступу : <http://tengrinews.kz/progress/poprobuyte-reshit-samuyu-trudnuyu-golovolomku-v-mire-216712/index.php>.
5. Lenta.ru: Наука и техника. – Математики решили задачу о подсказках в судоку [Електронний ресурс]. – Режим доступу : <http://lenta.ru/news/2012/01/09/sudoku/index.php>.
6. ВКонтакте: Социальная сеть [Електронний ресурс]. – Режим доступу : https://vk.com/topic-1196-3359_27503324.
7. Судоку и какуро. Решебник [Електронний ресурс]. – Режим доступу : <http://su-doku.ru/puzzle/-standart/index.php>.
8. Хабрахабр. Методы решения судоку [Електронний ресурс]. – Режим доступу : <http://habrahabr.ru/post/173795/index.php>.
9. Интернет-журнал "KM.RU". – Математики придумали формулу для решения судоку [Електронний ресурс]. – Режим доступу : <http://www.km.ru/science-tech/2012/10/19/issledovaniya-rossiiskikh-i-zarubezhnykh-uchenykh/695292-matematiki-pridumal>.
10. Хабрахабр. Алгоритм генерации судоку [Електронний ресурс]. – Режим доступу : <http://habrahabr.ru/post/192102/index.php>.
11. Интернет-журнал "IT race". – Генератор игрового поля в Судоку [Електронний ресурс]. – Режим доступу : http://it-race.blogspot.com/2012/02/blog-post_29.html.
12. Василенко С.Л. Числовая гармония Судоку / С.Л. Василенко // Научно-технический форум : SciTecLibrary.ru [Електронний ресурс]. – 2012. – Режим доступу : <http://www.sciteclibrary.ru>.
13. Каленюк П.І. Практикум з програмування на VBA / П.І. Каленюк, А.Ф. Обшта, Н.М. Гоблик та ін. – Львів : Видавництво Національного університету "Львівська політехніка", 2005. – 208 с.
14. Флэнаган Д. JavaScript. Подробное руководство / Д. Флэнаган. – СПб. : "Символ-Плюс", 2008. – 992 с.

УДК 004.023

Марченко О.І., к.т.н., доцент, Марченко О.О., аспірант, Орлова М.М., к.т.н., доцент
Національний технічний університет України «Київський політехнічний інститут»

СТРУКТУРА ТА КРИТЕРІЇ КЛАСИФІКАЦІЇ СПОСОБІВ РЕАЛІЗАЦІЇ ТА ПОКРАЩЕННЯ ПОШУКУ ПО ДЕРЕВУ МЕТОДОМ МОНТЕ-КАРЛО

Марченко О.І., Марченко О.О. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло. У статті, на основі отриманого з різних джерел великого обсягу інформації про пошук по дереву методом Монте-Карло (MCTS), особливості його поведінки при використанні в різних випадках та реалізації цього методу у багатьох застосунках, пропонується структура та критерії класифікації різних способів реалізації методу MCTS, його покращення та паралелізації. Розглянутий підхід до класифікації як для суто теоретичних способів покращення та налагодження методу MCTS для конкретних ситуацій, так і для способів пришвидшення роботи програм, що реалізують цей метод, за допомогою апаратних засобів. Передбачається, що запропоновані структура та критерії класифікації можуть бути використані для систематизації знань про метод MCTS, детальної класифікації способів реалізації цього методу та виявлення нових можливостей його покращення.

Ключові слова: задачі штучного інтелекту, дерева ігор, пошук в дереві, метод Монте-Карло, MCTS, способи покращення MCTS.

Марченко А.И., Марченко А.А. Структура и критерии классификации способов реализации и улучшения поиска по дереву методом Монте-Карло. У статье, на основе полученного из разных источников большого объема информации о поиске по дереву методом Монте-Карло (MCTS), особенностях его поведения при использовании в разных случаях и реализации этого метода во многих приложениях, предлагаются структура и критерии классификации разных способов реализации метода MCTS, его улучшения и параллелизации. Рассмотрен подход к классификации как сугубо теоретических способов улучшения и настройки метода MCTS для конкретных ситуаций, так и способов ускорения работы программ, что реализуют этот метод, при помощи аппаратных средств. Предполагается, что предложенные структура и критерии классификации могут быть использованы для систематизации знаний о методе MCTS, детальной классификации способов реализации этого метода и выявления новых возможностей его улучшения.

Ключевые слова: задачи искусственного интеллекта, деревья игр, поиск в дереве, метод Монте-Карло, MCTS, методы улучшения MCTS.

Marchenko O.I., Marchenko O.O. Structure and criteria for classification of techniques for implementation and improvement of Monte-Carlo tree search. Basing on large information taken from various sources about Monte-Carlo tree search (MCTS) method, specific features of its behavior for various cases of usage, and implementation of this method for many applications, the article proposes structure and criteria for classification of various techniques for MCTS method implementation, improvement, and parallelization. An approach to classification of pure theoretical techniques for MCTS improving and tuning in particular situations as well as techniques for hardware speedup of programs, which implement the method, is discussed. It is supposed that the structure and criteria for classification can be used for systematization of knowledge about MCTS method, detailed classification of implementation techniques of the method, and discovering of new possibilities for its improvement.

Keywords: artificial intelligence tasks, game trees, tree search, Monte-Carlo method, MCTS, MCTS improvement methods.

Постановка наукової проблеми. Однією з найбільш поширених форм подання інформації в задачах штучного інтелекту є форма у вигляді дерев послідовних рішень, а одним з методів штучного інтелекту, що був запропонований відносно недавно і призначений для виконання швидкого пошуку правильних рішень у дереві інформації, є метод пошуку в дереві з використанням методу Монте-Карло (Monte Carlo Tree Search – MCTS) [2, 3]. Задачі штучного інтелекту відзначаються надзвичайно великим обсягом даних і високими вимогами до швидкості їх обробки. Характерними і популярними задачами штучного інтелекту, на яких, як правило, виконується апробація нових методів, алгоритмів, структур даних є складні інтелектуальні ігри, наприклад, такі як ігри в шахи та Го.

MCTS є методом пошуку оптимальних рішень у заданій області за допомогою випадкових значень із заданого простору значень і побудови дерева пошуку за отриманими результатами. Цей метод вже встиг суттєво вплинути на розв'язання задач штучного інтелекту, а особливо складних ігрових задач (шахи, Го, тощо).

Аналіз досліджень. За час існування методу MCTS було створено достатньо багато його варіантів та модифікацій з метою прискорення процесу пошуку. Одні з цих модифікацій зроблені на алгоритмічному рівні MCTS, в той час як інші – на рівні розпаралелення процесу пошуку. Цьому методу присвячено вже багато досліджень і публікацій, на декілька з них автори посилаються в даній

роботі. Найбільш ґрунтовною узагальнюючою роботою з методу MCTS є огляд 2012 року, підготовлений десятима провідними вченими під керівництвом Камерона Броуне, що досліджують цей метод [3], але і в ній також відсутня строга ієрархічна класифікація способів реалізації MCTS.

Автори вважають, що систематизація інформації про різні підходи до прискорення пошуку методом MCTS та знаходження нових критеріїв для виконання такого прискорення дозволять більш чітко виявити ще не використані можливості пошуку цим методом.

Метою даної роботи є з'ясування структури та критеріїв для класифікації та систематизації інформації про способи реалізації методу пошуку в дереві методом Монте-Карло (MCTS) та особливостей його поведінки при використанні в різних випадках.

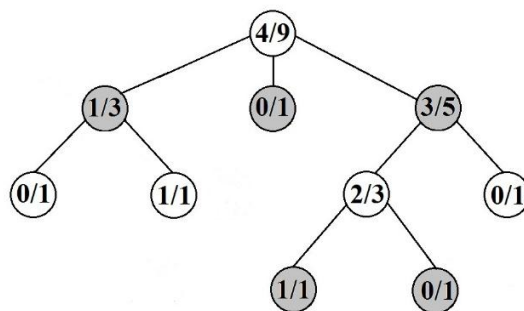
Пошук в дереві з використанням методу Монте-Карло (MCTS). Моделювання методом Монте-Карло [1] є стохастичним і за допомогою повторень такого моделювання можна отримати корисну інформацію. Особливо добре таке моделювання вирішує проблеми, у яких немає відомого детермінованого рішення, і тому його було вирішено застосувати також до рішення таких складних задач як моделювання ігор в шахи та Го.

Одне моделювання гри методом Монте-Карло виконує процес моделювання від деякої поточної позиції до кінця гри, вибираючи ігрові ходи випадковим чином. Як тільки моделювання досягає кінця гри, виконується обчислення балів для кожного гравця та встановлюється переможець. Процес моделювання починаючи від однієї й тієї самої позиції виконується багато разів, після чого відношення числа вигравів до числа програшів формує сумарну оцінку цієї позиції.

Незважаючи на те, що таке моделювання використовує тільки правила відповідної гри (шахи, Го, тощо) для вибору наступних ходів, за допомогою повторень воно в змозі надати корисну інформацію відносно того, який хід є найкращим в даній позиції [3, 4].

Для покращення ігрової потужності метод Монте-Карло був застосований до дерева пошуку однієї з ігор, і, в результаті, був отриманий метод пошуку в дереві з використанням методу Монте-Карло – Monte-Carlo Tree Search (MCTS). Вперше MCTS був застосований в комп'ютерній грі Го [3]. З того часу MCTS отримав широке використання, особливо в галузі загально-ігрових задач – General Game Playing (GGP) [5].

Нове під-дерево MCTS асоціюється з поточною позицією гри як коренева вершина, а всі вершини в цьому під-дереві будуть зберігати деяку кількість вигравів і програшів [3]. На рисунку 1 наведено приклад дерева MCTS, де в кожній вершині показано результат багатократного моделювання гри від цієї вершини. Всі термінальні (листові) вершини і нетермінальні вершини, до яких зберігається доцільність додавання нових листів-нащадків, формують межу (передній фронт) дерева



MCTS.

Рис.1. Приклад дерева MCTS

Загальний підхід методу MCTS. З метою більш глибокого аналізу особливостей методу MCTS розглянемо його загальний підхід у більш формалізованому вигляді.

Як було сказано, дерево пошуку будується шляхом додавання до нього кожен раз по одній вершині відповідно до результатів моделювання гри від заданої вершини до закінчення гри, а загальний підхід методу MCTS складається з чотирьох повторюваних кроків, показаних на рис.2 [3].

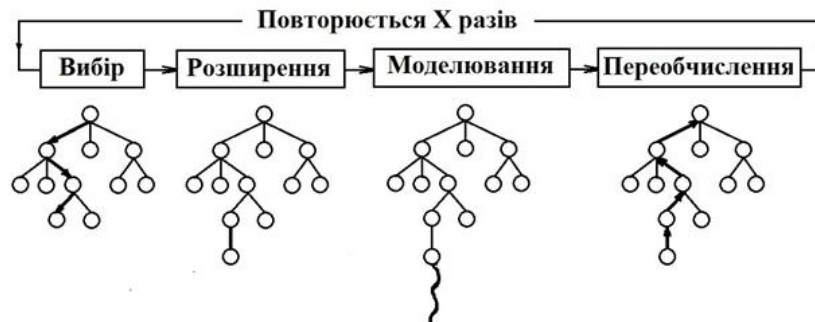


Рис.2. Загальний підхід методу MCTS

Крок 1. Вибір. Починаючи від поточного кореневого вузла R і використовуючи задану політику дерева (Tree Policy), вибираються оптимальні вершини-нащадки (пояснюється нижче) доки не буде досягнута довільна вершина-лист L.

Крок 2. Розширення. Якщо L є нетермінальною вершиною (тобто на ній гра ще не закінчується), то додається нова вершина V, тобто відбувається спроба виконати черговий хід згідно правил гри. Черговий хід із множини можливих в даному стані ходів береться згідно політики розширення (Expansion Policy), якою, як правило, є взяття випадкового ходу.

Крок 3. Моделювання. Починаючи від взятої вершини V виконується процес моделювання гри до досягнення результату Res згідно прийнятої політики моделювання (Simulation Policy). Найчастіше береться найпростіша політика моделювання, яка називається політикою за умовчанням (Default Policy) і яка полягає у взятті всіх наступних ходів випадковим чином.

Крок 4. Переобчислення. Враховуючи отриманий результат моделювання Res у зворотному порядку від вершини L до кореневої вершини R виконується корекція значень атрибутів у вершинах поточної послідовності ходів згідно політики зворотного переобчислення (Backpropagation Policy).

Як результат однократного виконання функції пошуку, повертається одна з нових вершин, отриманих під час процесу пошуку від кореня R, значення результату якої є найкращим.

Більш формально загальний підхід методу MCTS можна записати у вигляді функції пошуку MCTS_Search [3].

```
function MCTS_Search (S) {
    Create (R, S);
    while WithinComputationalBudget do {
        L = TreePolicy (R);
        V = ExpansionPolicy (L);
        Res = SimulationPolicy (V);
        BackpropagationPolicy (L, Res);
    }
    return BestChild (R);
}
```

MCTS_Search (S) – функція пошуку в дереві методом MCTS, яка приймає у якості параметра поточний стан гри (позицію) S і повертає найкращу з нових, отриманих під час моделювання, вершин дерева, тобто новий найкращий хід від даної позиції.

Create (R, S) – функція створення нової кореневої вершини R для заданого поточного стану гри S.

WithinComputationalBudget – булева функція, яка повертає значення True, якщо обчислювальний бюджет (заданий час чи задана кількість повторень) ще не вичерпався.

TreePolicy (R) – функція, яка реалізує дії першого кроку методу MCTS.

ExpansionPolicy (L) – функція, яка реалізує дії другого кроку методу MCTS.

SimulationPolicy (V) – функція, яка реалізує дії третього кроку методу MCTS.

BackpropagationPolicy (L, Res) – функція, яка реалізує дії четвертого кроку методу MCTS.

BestChild (R) – функція, яка визначає і повертає найкращу з нових, отриманих під час моделювання від кореня R, вершин дерева, тобто новий найкращий хід від даної позиції.

Побудова довільного дерева MCTS (як правило, дерева гри) виконується інкрементальним і асиметричним способом. На кожній ітерації методу MCTS використовується політика дерева для знаходження найкращої для продовження гри вершини у поточному дереві. При розробці політики дерева намагаються збалансувати певним чином *кількість досліджень (exploration)*, виконуючи вибір нової вершини у тих частинах дерева, які були ще недостатньо добре опрацьовані, та *кількість використання (exploitation)*, виконуючи вибір нової вершини у тих частинах дерева, які вже є на даний момент найбільш перспективними. Після цього виконується моделювання (випадкова або статистично розподілена послідовність ходів гри до її закінчення), починаючи від обраної вершини, а потім значення деяких вершин дерева корегуються згідно з отриманим результатом. Під час такого процесу додається одна вершина-нащадок, яка відповідає ходу, виконаному від обраної вершини, а корекція значень вершин відбувається у її вершинах-предках. Нагадаємо, що під час моделювання ходи виконуються згідно політики моделювання, яка у найпростішому випадку за умовчанням (Default Policy) є політикою взяття однаково випадкових ходів.

Структура та критерії класифікації способів реалізації методу MCTS. Згідно розглянутого вище загального підходу метода MCTS логічним буде класифікування способів реалізації цього метода за критерієм покращення кожного з етапів роботи MCTS. З іншої сторони, в попередні роки було з'ясовано, що в прикладенні до гри Го, за наявності достатньої обчислювальної потужності, MCTS працює набагато краще, ніж інші методи, в результаті чого, цей метод став парадигмою для комп'ютерної гри Го. Крім того, в [3] було зазначено, що з нескінченним часом метод MCTS буде давати сходження до оптимальної гри, і що зі збільшенням числа моделювань, ефективність пошуку методом MCTS буде покращуватись [6]. Для збільшення числа моделювань потрібно або збільшувати час виконання моделювання або збільшувати швидкість моделювань. Це може бути досягнуто за допомогою паралелізації пошуку та обчислень. Відповідно до цього, виконання паралелізації є ще одним критерієм класифікації методів та способів реалізації MCTS. В свою чергу, за критерієм орієнтації способу паралелізації на конкретні апаратні ресурси, чи відсутності такої орієнтації, існуючі способи паралелізації можуть бути поділені на апаратно-незалежні способи та апаратно-орієнтовані способи.

Згідно розглянутих критеріїв структура класифікації буде такою, як показано на рисунку 3.

MCTS є загальним методом пошуку в дереві з використанням методу Монте-Карло, якому відповідає ціла множина досить різних алгоритмів, які відрізняються використаними політиками дерева, розширення, моделювання та зворотного переобчислення.

В залежності від використаної політики дерева, тобто вибору вершини в дереві для подальшого моделювання, виділяють наступні групи алгоритмів, що відповідають загальному методу MCTS [3]:

- Plain UCT (Upper Confidence bounds for Trees): прямий спосіб реалізації згідно методу MCTS з політикою вибору дерева, у якій використовується метод верхніх довірчих границь UCB (Upper Confidence Bounds) у вигляді, як початково було запропоновано в [4];
- UCT: UCT-орієнтоване під-сімейство способів реалізації MCTS з політикою вибору дерева, у якій використовується довільний варіант реалізації методу верхніх довірчих границь UCB;
- Non-UCT: сімейство способів реалізації, що працюють згідно наведеного вище загального підходу до пошуку в дереві з використанням методу Монте-Карло з довільною політикою вибору дерева, але не використовують UCT.

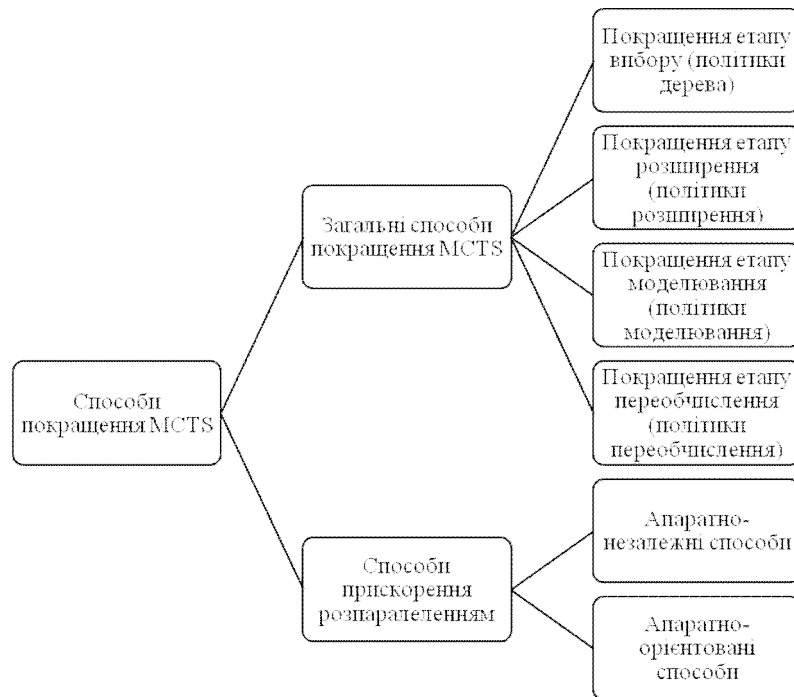


Рис. 3. Структура класифікації способів покращення MCTS

Аналогічно існують також різні варіанти реалізації MCTS, які орієнтовані на покращення інших етапів роботи MCTS. Це може бути предметом подальшої деталізації загальних способів покращення MCTS у запропонованій класифікації.

З часу створення метода MCTS пошуки кращих варіантів його реалізації відбувались, в основному, серед сімейства UCT-орієнтованих способів. Але зовсім нещодавно, у січні 2015 року, Трістаном Казенаве [7] був запропонований новий перспективний підхід, який він назвав SHOT (Sequential Halving applied to Trees), і який належить до Non-UCT сімейства способів MCTS. Зрозуміло, що дослідження покращення/пришвидшення цього способу реалізації MCTS ще тільки починаються.

Результуюча структура класифікації способів покращення етапу вибору (політики дерева) показана на рис.4.

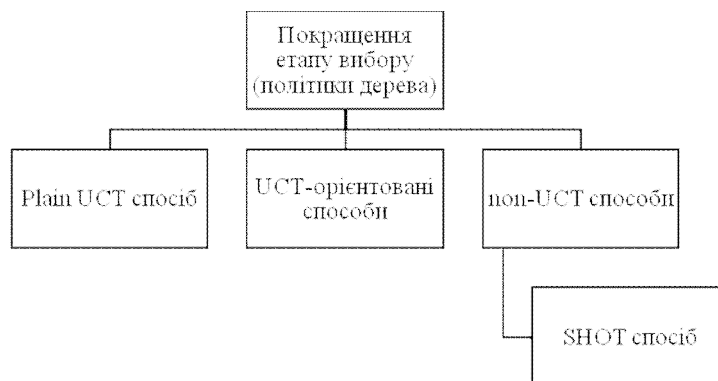


Рис. 4. Структура класифікації способів покращення етапу вибору (політики дерева)

На сьогоднішній день для розпаралелення пошуку в дереві методом MCTS на апаратно-незалежному рівні були запропоновані три загальних підходи, які були названі: листкова паралелізація, коренева паралелізація і деревна паралелізація [3].

Ці підходи показані на рисунках 5, 6, 7а, 7б.

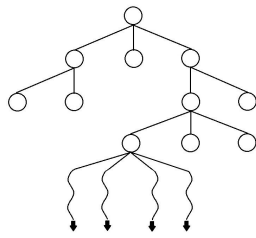


Рис.5. Листкова паралелізація

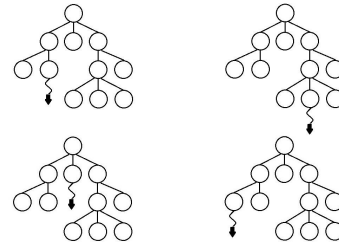


Рис.6. Коренева паралелізація

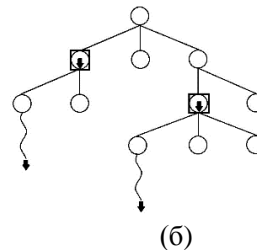
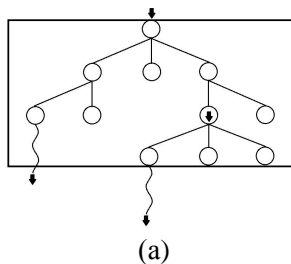


Рис.7. Деревна паралелізація з глобальним м'ютексом (а) та з локальними м'ютексами (б)

Розглянемо кожен з цих підходів паралелізації більш детально.

Листкова паралелізація як визначено в [8] – це виконання множини одночасних моделювань від вершини-листа, яка була додана після закінчення процесу пошуку актуальної вершини, використовуючи політику дерева. Ідея полягає у тому, щоб отримати кращу статистику на кожній вершині-листу за допомогою кращої початкової оцінки. В [9] таку схему назвали паралелізацією на листках (at-the-leaves parallelization). Пізніше вкоренився термін «листова паралелізація» (leaf parallelisation).

Кореневу паралелізацію [8] іноді називають багато-деревним MCTS, оскільки декілька дерев пошуку MCTS будуються одночасно (тобто паралелізуються на корені дерева). Як правило, для повідомлення про те, який хід був обраний алгоритмом у кожному з дерев, використовується інформація з першого рівня цих дерев. Однією з переваг такого підходу є те, що кожен паралельний потік може виконуватись протягом фіксованого часу і зупинений у будь-який момент.

Деревна паралелізація є процесом, який виконує декілька одночасних кроків моделювання на одному й тому ж дереві [8]. В цьому випадку виникає класична проблема паралельного програмування, що полягає у недопущенні одночасного доступу до дерева з різних потоків. Для цього потрібно забезпечити ексклюзивний доступ до певного під-дерева MCTS кожному потоку, а також зробити так, щоб всі потоки працювали у різних областях дерева, доки м'ютекс не буде звільнений. В одній схемі, запропонованій в [8], використовується глобальний м'ютекс на кореневій вершині (рис.3а). Такий підхід є резонним у тому випадку, якщо час виконання моделювання є набагато довшим, ніж час вибору вершини з дерева, оскільки один потік може проходити по дереву або робити переобчислення у дереві, в той час, як інші потоки виконують моделювання. В іншій схемі (рис.3б) використовуються локальні м'ютекси на кожній внутрішній вершині, яка блокується і звільняється кожен раз, коли якийсь потік приходить у цю вершину.

Як зазначалося вище, розробка, дослідження і паралелізація SHOT-орієнтованих способів є перспективними напрямками.

Відповідно до розглянутих вище критеріїв структура класифікації апаратно-незалежних способів реалізації MCTS має вигляд, показаний на рисунку 8.

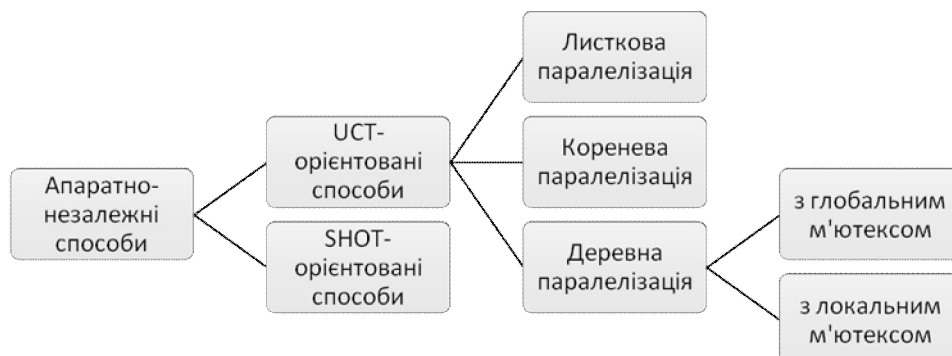


Рис.8. Структура класифікації апаратно-незалежних способів реалізації MCTS.

На апаратно-орієнтованому рівні покращення MCTS вже використовувався широкий спектр універсальних, спеціалізованих та змішаних (гібридних) апаратних платформ і мультикомп'ютерних систем. Тому початок цієї гілки класифікації може виглядати як показано на рисунку 9.

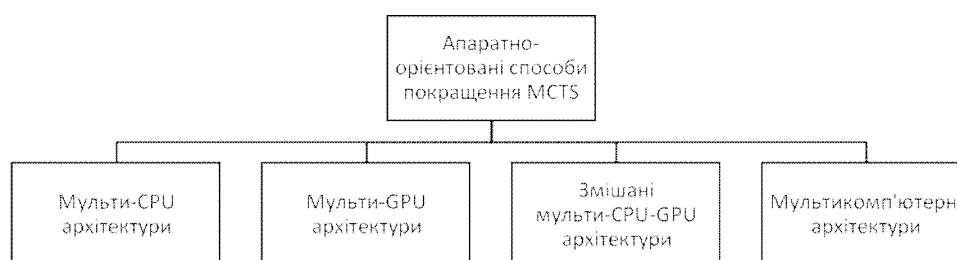


Рис. 9. Структура класифікації апаратно-орієнтованих способів реалізації MCTS.

Деталізація апаратно-орієнтованих способів паралелізації MCTS є напрямом подальших досліджень.

Висновки. Запропонована критерії та структура класифікації способів реалізації пошуку в дереві методом MCTS. Отримані результати дозволяють виконати систематизацію інформацію як про вже існуючі способи реалізації MCTS, так і виявити ще не дослідженні шляхи та варіанти реалізації цього методу. Подальшим напрямком дослідження може бути як подальша деталізація запропонованої структури класифікації, так власне класифікація існуючих способів реалізації методу MCTS згідно даних критеріїв та структури враховуючи особливості його поведінки при використанні в різних випадках.

1. Метод Монте-Карло [Електронний ресурс]. Режим доступу: http://uk.wikipedia.org/wiki/Метод_Монте-Карло
2. Monte Carlo tree search [Електронний ресурс]. Режим доступу: http://en.wikipedia.org/wiki/Monte-Carlo_tree_search.
3. Cameron Browne and others. A Survey of Monte Carlo Tree Search Methods // IEEE Trans. on Computational Intelligence and AI in Games. — vol. 4. — no. 1. — March 2012.
4. L. Kocsis and C. Szepesvarri, "Bandit based Monte-Carlo Planning," in Euro. Conf. Mach. Learn. Berlin, Germany: Springer, 2006, pp. 282–293.
5. J. Mehat and T. Cazenave, "Combining UCT and nested monte carlo search for single-player general game playing," IEEE Transactions on Computational Intelligence and AI in Games, vol. 2, no. 4, p. 271, 2010.
6. R. B. Segal, "On the scalability of parallel UCT," Lecture Notes in Computer Science, vol. 6515/2011, pp. 36–47, 2011.
7. T. Cazenave, "Sequential Halving Applied to Trees," IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, no. 1, pp. 102–105, 2015.
8. G. M. J.-B. Chaslot, M. H. M. Winands, and H. J. van den Herik. Parallel Monte-Carlo Tree Search // Proc. Comput. And Games, LNCS 5131, Beijing, China. — 2008, pp. 60–71.
9. T. Cazenave and N. Jouandea. On the Parallelization of UCT // Proc. Comput. Games Workshop, Amsterdam, Netherlands. — 2007, pp. 93–101.

УДК 61:004.651(075.8)

Андрушак І.Є.

Луцький національний технічний університет

МУЛЬТИВАРІАТИВНИЙ МЕТОД ЯКІСНОГО АНАЛІЗУ МОДЕЛІ НА ОСНОВІ КЛАСИФІКАЦІЙНИХ ПРАВИЛ.

Андрушак І.Є. Мультівариативний метод якісного аналізу моделі на основі класифікаційних правил. У роботі запропоновано підхід якісного аналізу системи протипухлинного імунітету на основі мультівариативного методу, що включає алгоритм послідовного покриття. При цьому на виході отримуємо множину класифікаційних правил для прогнозування форми патологічного процесу.

Ключові слова: мультівариативний метод, якісний аналіз, протипухлинний імунітет, диференціальні рівняння із запізненням, класифікаційні правила.

Андрушак И.Е. Мультивариативный метод анализа модели на основе классификационных правил. В работе предложен подход анализа системы противоопухолевого иммунитета на основе мультивариативного метода, включая алгоритм последовательного покрытия. При этом на выходе получаем множество классификационных правил для прогнозирования формы патологического процесса.

Ключевые слова: мультивариативный метод, качественный анализ, противоопухолевый иммунитет, дифференциальные уравнения с опозданием, классификационные правила.

Andrushchak I.Ye. Multivariate method of qualitative analysis models based on rules classification. In this work proposes an approach qualitative analysis of the antitumor immunity from multivariate method comprising sequential algorithm coverage. Thus the output we get the set of classification rules for predicting forms of pathological process.

Keywords: multivariate method, qualitative analysis, antitumor immunity, differential equations with delay, classification rules.

Завданням методу є встановлення механізмів багатопараметричних впливів у моделі протипухлинного імунітету.

Загальні ідеї методу були розроблені в роботі [Koch, 2013] для випадку ЗДР. У даній роботі його буде розвинено для ФДР. При цьому використовуємо підхід Монте-Карло, який полягає у випадковій генерації параметрів та побудові на їх основі моделі ФДР. Далі застосовують алгоритм індукції дерева рішень. Зауважимо, що метод роботи [Koch, 2013] застосовувався лише для дослідження впливу початкових умов на траєкторію системи ФДР. В даній роботі метод буде розвинено до швидкісних параметрів системи функціонально-диференціальних рівнянь також.

Отже, припускається існування моделі на основі ФДР при початкових значеннях та швидкісних параметрах із заданих інтервалів:

$$\frac{dL(t)}{dt} = \alpha_L L(t) \ln \frac{\theta_L}{L(t)} - \gamma_L F(t) L(t), \quad (1)$$

$$\frac{dC(t)}{dt} = \xi(m) \alpha L(t - \tau) F(t - \tau) - \mu_C (C - C_0), \quad (2)$$

$$\frac{dF(t)}{dt} = \rho C - (\mu_f + \eta \gamma_L L(t)) F(t), \quad (3)$$

$$\frac{dm(t)}{dt} = \sigma L(t) - \mu_m m(t), \quad (4)$$

з початковими умовами при $t \in [-\tau, 0]$:

$$V(t) = V_0, F(t) = F_0, C(t) = C_0, m(t) = 0.$$

Тут α_L – коефіцієнт розмноження пухлинних клітин; θ_L – життєва ємність популяції пухлинних клітин; γ_L – коефіцієнт, що визначає ймовірність нейтралізації (руйнування) пухлинної клітини антитілом; α – коефіцієнт, що зумовлює ймовірність зустрічі антиген-

антитіло; μ_c – коефіцієнт, обернений до часу життя плазмоклітин; ρ – швидкість виробництва антитіл однією плазмоклітиною; μ_f – коефіцієнт, обернено пропорційний до часу розпаду антитіл; η – число антитіл, що вимагається на нейтралізацію одного антигена; σ – коефіцієнт, що визначає швидкість загибелі клітин за рахунок пошкоджуючої дії антигена; μ_m – коефіцієнт, що враховує швидкість відновлення пошкодженого органу; τ – фаза запізнення (час, за який здійснюється формування каскаду плазмоклітин); $\xi(m)$ – неперервна незростаюча функція ($0 \leq \xi(m) \leq 1$), що характеризує порушення нормального функціонування імунної системи через значне пошкодження органа-мішені.

Параметри:

$$\begin{aligned} p \in P = \{(\alpha_L, \theta_L, \gamma_L, \alpha, \mu_c, \rho, \mu_f, \eta, \sigma, \mu_m) : \alpha_L^{\min} \leq \alpha_L \leq \alpha_L^{\max}, \\ \theta_L^{\min} \leq \theta_L \leq \theta_L^{\max}, \gamma_L^{\min} \leq \gamma_L \leq \gamma_L^{\max}, \alpha^{\min} \leq \alpha \leq \alpha^{\max}, \mu_c^{\min} \leq \mu_c \leq \mu_c^{\max}, \\ \rho^{\min} \leq \rho \leq \rho^{\max}, \mu_f^{\min} \leq \mu_f \leq \mu_f^{\max}, \eta^{\min} \leq \eta \leq \eta^{\max}, \sigma^{\min} \leq \sigma \leq \sigma^{\max}, \\ \mu_m^{\min} \leq \mu_m \leq \mu_m^{\max}\} \subset R^{10}, \end{aligned}$$

а початкові умови

$$\begin{aligned} (L_0, F_0, C_0, m_0) \in X_0 = \{(L_0, F_0, C_0, m_0) : L_0^{\min} \leq L_0 \leq L_0^{\max}, F_0^{\min} \leq F_0 \leq F_0^{\max}, \\ C_0^{\min} \leq C_0 \leq C_0^{\max}, m_0^{\min} \leq m_0 \leq m_0^{\max}\} \subset R^4. \end{aligned}$$

Далі випадковим чином генеруватимемо початкові значення та значення швидкісних параметрів, які б належали практично обґрунтованій області. Для кожного з наборів таких параметрів здійснюється інтегрування системи (1)-(4) з отриманням відповідних траєкторій. До отриманих результатів далі застосовується алгоритм індукції дерева рішень з метою знаходження певних шаблонів для прийняття рішень.

Отже, вцілому підхід включає такі п'ять кроків.

1. Означення класів траєкторій системи. Зазначимо, що в практичних застосуваннях переважно мають справу з набагато складнішими формами поведінки, щоб охарактеризувати їх поняттями «стійка-нестійка» і відповідно вдатися до аналізу власних значень або ж експонент Ляпунова динамічної системи. Тому визначення якісних форм процесу доцільно передати до компетенції експертів. В даному випадку використовуватимемо класи, пов'язані з формами патологічного процесу: субклінічна, гостра, хронічна, летальна. Для позначення класу траєкторії вводиться атрибут класу C , який приймає одне з 4-х дискретних значень $C \in \overline{1,4}$.

2. Генерація матриці випадкових початкових значень та швидкісних параметрів. Для того, щоб дослідити весь простір початкових значень та швидкісних параметрів щодо генерації класів траєкторій, визначених на першому кроці, генерується матриця випадкових початкових значень та швидкісних параметрів на основі ймовірнісних розподілів у визначених інтервалах. У даній роботі ми припускаємо, що початкові значення та швидкісні параметри розподілені рівномірно на інтервалах. Кожен стовпчик відповідає множині значень одного параметру -- або початкове значення, або швидкісний параметр. Кожен рядок є набором початкових значень та швидкісних параметрів для одного запуску моделі на основі ФДР:

$$M = \begin{pmatrix} V_0^1 & F_0^1 & C_0^1 & m_0^1 & \alpha_L^1 & \theta_L^1 & \gamma_L^1 & \alpha^1 & \mu_c^1 & \rho^1 & \mu_f^1 & \eta^1 & \sigma^1 & \mu_m^1 \\ V_0^2 & F_0^2 & C_0^2 & m_0^2 & \alpha_L^2 & \theta_L^2 & \gamma_L^2 & \alpha^2 & \mu_c^2 & \rho^2 & \mu_f^2 & \eta^2 & \sigma^2 & \mu_m^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ V_0^N & F_0^N & C_0^N & m_0^N & \alpha_L^N & \theta_L^N & \gamma_L^N & \alpha^N & \mu_c^N & \rho^N & \mu_f^N & \eta^N & \sigma^N & \mu_m^N \end{pmatrix} \in R^{N \times 14}$$

3. Запуск моделі і класифікація набору вхідних даних. Кожен набір початкових значень та швидкісних параметрів, згенерованих на другому кроці, використовуються в якості входу для моделі протипухлинного імунітету. Чисельне інтегрування рівнянь здійснюється за допомогою методу Адамса [Хайрер, Нерсет]. Вихідні траєкторії класифікуються на основі критеріїв, запропонованих на першому кроці. Виходячи з результатів класифікації наборам початкових значень і швидкісних параметрів приписуються відповідні атрибути класів.

4. Побудова матриці залежностей між початковими значеннями та між швидкісними параметрами. Метод припускає, що для форми траєкторій системи співвідношення між початковими значеннями та між швидкісними значеннями є набагато важливішими порівняно з їх абсолютними значеннями. Тому будується матриця, що включає інформацію у категоризованому кодованому вигляді про співвідношення між початковими значеннями та між швидкісними параметрами.

$C_i \in \overline{1,4}$ – значення атрибуту класу, пов'язані з відповідними формами траєкторій.

Отже, на даному кроці чисельні значення початкових значень та швидкісних параметрів трансформуються у категоріальні значення атрибутів наборів навчальних даних. Оскільки ймовірність рівності випадкових чисел дорівнює нулю, то матриця D виглядає свого роду «бінаризацією» співвідношень між початковими значеннями та між швидкісними параметрами. Тобто матриця D включатиме лише значення 0 та 2.

5. Застосування алгоритму послідовного покриття до співвідношень між початковими значеннями та між швидкісними параметрами. Матриця бінарних співвідношень D побудована на кроці 4, є набором навчальних даних для алгоритму послідовного покриття. Побудований набір класифікаційних правил міститиме перевірку співвідношень між початковими значеннями та швидкісними параметрами в антецедентах. В якості консеквентів правил будуть класи траєкторій моделі $C \in \overline{1,4}$.

Програмна реалізація мультиваріативного методу для моделі протипухлинного імунітету. Для реалізації методу розроблено пакет Java-класів rule.model. До складу пакету входять класи (рис.1):

beans-класи Attribute, Attribute_for_list для роботи з даними відповідних таблиць та Rule – для представлення правил. SQL-запити щодо отримання відповідних даних реалізовано в класах AttributeListPeer та TuplesPeer.

У класі Rule_set зберігається набір навчальних правил. До того ж даний клас безпосередньо реалізує алгоритм послідовного покриття. Клас містить члени: менеджер даних m_dataManager, хеш-таблиці наборів навчальних даних m_htTuples, усіх атрибутів з їх можливими значеннями m_htAtt_vals та безпосередньо множину правил m_htRule_set.

У конструкторі класу Rule_set здійснюється побудова хеш-таблиць m_htTuples та m_htAtt_vals, а також застосування алгоритму послідовного покриття – через виклик методу Sequential_covering(m_htTuples, m_htAtt_vals). Отримана множина правил виводиться в текстовий файл.

Клас Rule призначений для зберігання окремих правил. Його членами класу є дві хеш-таблиці: m_htAntecedent – для зберігання антецеденту правила та m_htConsequent – для консеквенту. За допомогою методу:

public void conjunctCondition(Attribute_for_list attribute, String sAttribute_value)

здійснюється кон'юнкція нової умови до правила. За допомогою методу:

public Rule copy()

створюється «глибока» копія правила. При цьому використовується протокол JOS (Java Object Serialization).

Підрахунок кількості позитивних та негативних навчальних наборів здійснюється у методах класу TuplesPeer.

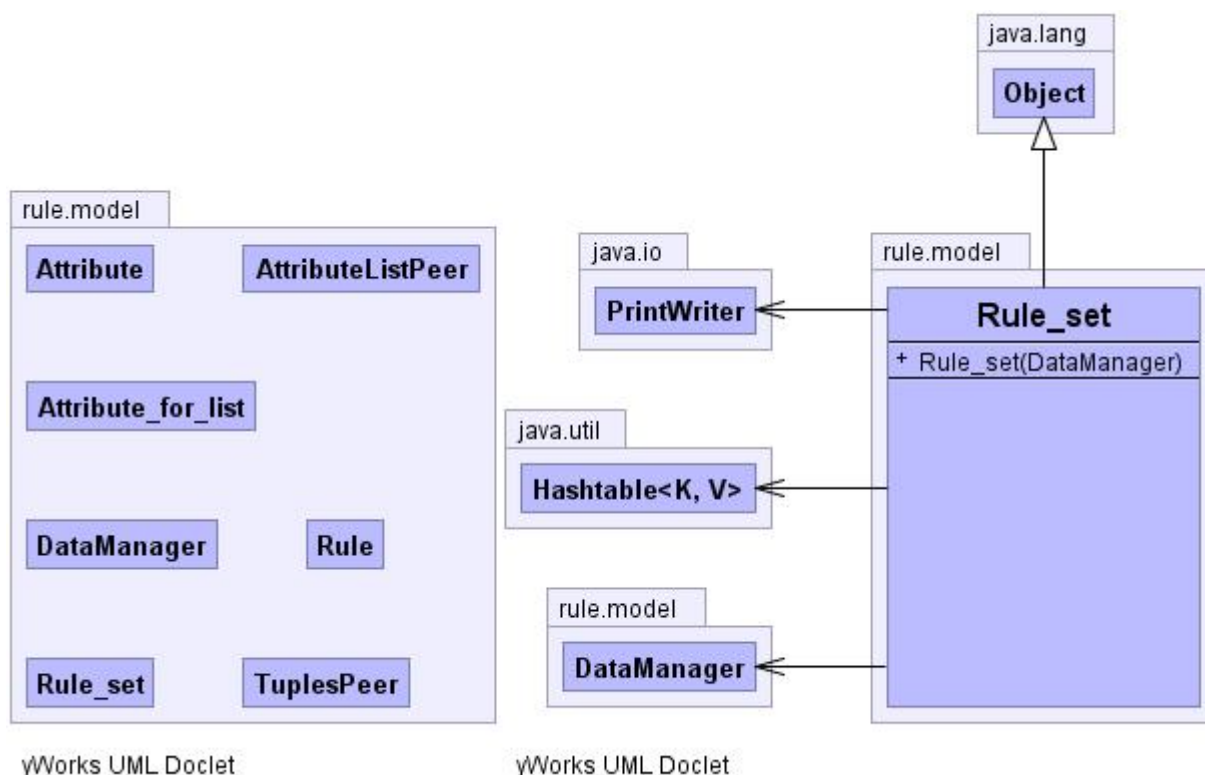


Рис. 1. Пакет rule.model

У класі `fde.MultiVariateMethod` (рис.2) здійснюється генерація випадкових значень параметрів (крок 2):

```
M_x0 = dm.getRandomInitialValues();
M_rateConstants = dm.getRandomRateConstants();
```

Далі запускається аплет інтегрування системи ФДР. При цьому експерт здійснює вибір форми отриманої траєкторії (крок 3). Після цього запускається крок генерації матриці взаємозв'язків параметрів (крок 4). Зауважимо, що послідовність кроків 2-4 може виконуватися як завгодно багато разів. У будь-який момент користувач може запустити алгоритм індукції дерева рішень (крок 5):

```
rule.model.Rule_set rule_set = new rule.model.Rule_set(rule_dataManager, sql0, rules_file_url);
```

База даних `fde`, що використовується в пакеті, реалізована в СУБД MySQL. Вона включає такі таблиці:

`attribute` – опис атрибутів для побудови класифікаційних правил, тобто взаємозв'язків між початковими значеннями та між швидкісними константами;

`categorized_data` – навчальні набори, що використовуються в класифікаційному алгоритмі (в даному випадку алгоритму послідовного покриття) і представляють собою матрицю D на четвертому кроці;

`init_values_values` – матриця згенерованих випадковим чином початкових значень:

$$\begin{pmatrix} V_0^1 & F_0^1 & C_0^1 & m_0^1 \\ V_0^2 & F_0^2 & C_0^2 & m_0^2 \\ \vdots & \vdots & \vdots & \vdots \\ V_0^N & F_0^N & C_0^N & m_0^N \end{pmatrix} \in R^{N \times 4}$$

`initial_values` – опис початкових значень (включаючи мінімальні та максимальні значення);

`parameter_kind` – вид параметру;

`rate_constants` – опис швидкісних констант (включаючи мінімальні та максимальні значення);

rate_constants_values – матриця згенерованих випадковим чином швидкісних констант:

$$\left(\begin{array}{cccccccccc} \alpha_L^1 & \theta_L^1 & \gamma_L^1 & \alpha^1 & \mu_C^1 & \rho^1 & \mu_f^1 & \eta^1 & \sigma^1 & \mu_m^1 \\ \alpha_L^2 & \theta_L^2 & \gamma_L^2 & \alpha^2 & \mu_C^2 & \rho^2 & \mu_f^2 & \eta^2 & \sigma^2 & \mu_m^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha_L^N & \theta_L^N & \gamma_L^N & \alpha^N & \mu_C^N & \rho^N & \mu_f^N & \eta^N & \sigma^N & \mu_m^N \end{array} \right) \in R^{N \times 10}$$

Особливості програмної реалізації моделі протипухлинного імунітету. Пакет rule.model може бути використаний для широкого класу систем функціонально-диференціальних рівнянь. Для цього модель на основі функціонально-диференціальних рівнянь повинна бути реалізована у вигляді відповідного пакету Java-класів.

Даний пакет було розроблено в роботі [Гвоздецька, 2012]. Для інтеграції з пакетом decision_tree.fde в класі TumorImmunitySystemGraph, що здійснює графічну візуалізацію моделі, поряд з існуючим було додано новий конструктор, що використовує посилання за значенням на інстанцію класу MultiVariateMethod. У цьому конструкторі додатково створюється об'єкт класу JComboBox, що дозволяє вибирати форму траєкторії та запускати виконання 4-го кроку методу:

```
String[] classStrings = {"subclinical", "chronic", "acute", "lethal"};
JComboBox m_jcbClassName = new JComboBox(classStrings);
m_jcbClassName.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JComboBox jcbClass = (JComboBox) e.getSource();
        m_sClassName = (String) jcbClass.getSelectedItem();
        mvm.m_sClassName = m_sClassName;
        ((AdvancedFrame) getParent()).dispose();
        mvm.run4thStep();
    }
});
```

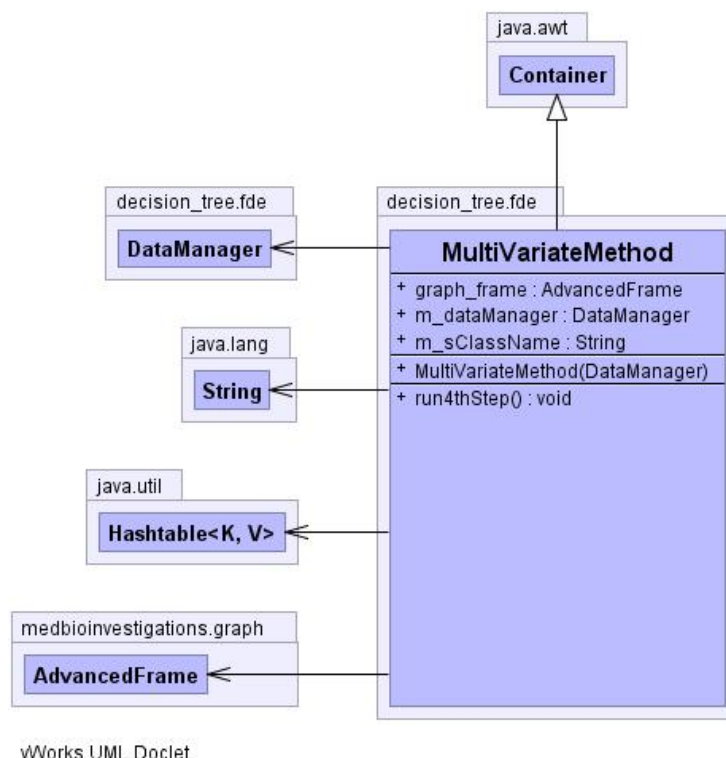


Рис.2. UML-діаграма класу MultiVariateMethod

Також слід внести відповідні зміни у таблиці бази даних fde в такому порядку:

- описати усі початкові значення та швидкісні константи в таблицях initial_values та rate_constants відповідно;

- описати залежності між початковими значеннями та між швидкісними константами, які досліджуватимемо, у таблиці attribute;

- у таблиці categorised_data створити поля у відповідності з даними таблиці attribute.

Існує чимало реалізацій алгоритму послідовного покриття побудови класифікаційних правил, зокрема у складі програм Mathcad, StatSoft Statistica і ін. Суттєвою перевагою даної реалізації є те, що алгоритм розвинено до Інтернет-проекту, де до формування навчальних наборів залучаються експерти з Інтернет-доступом, що значно розширює та балансує множину навчальних наборів.

1. Ахо А., Хопкрофт Дж., Ульман Дж., Построение и анализ вычислительных алгоритмов. – М.: Мир. – 1979. – 536 с.
2. Гвоздецька І.С. Математичні моделі пухлинного росту на основі динаміки Гомперца / Дис. на здобуття наукового ступеня кандидата техн. наук. – Тернопіль: 2012. – 130 с.
3. Марценюк В.П. Математичні моделі в системі підтримки прийняття рішень страхового забезпечення лікування онкологічних захворювань: підхід на основі динаміки Гомперца / В.П. Марценюк, І.Є. Андрушак, І.С. Гвоздецька, Н.Я. Климук // Доповіді Національної академії наук України. –2012. – №10. – С. 34 – 39.
4. Марценюк В.П. Підхід на основі актуарних математичних моделей до задач страхової медицини / В. П. Марценюк, І.Є. Андрушак, Н.Я. Климук // Медична інформатика та інженерія. Науково-практичний журнал. – 2010. – №4. – С. 85 – 87.
5. Хайрер Э., Нерсерт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. – М.: Мир. –1990. – 512 с.
6. Clemens Kuhn, Christoph Wierling, Alexander Kuhn, Edda Klipp, Georgia Panopoulou, Hans Lehrach and Albert J Poustka, Monte Carlo analysis of ODE Model of the Sea Urchin Endomesoderm Network / BMC Systems Biology, 2009, 3:83 doi: 1186/1752-0509-3-83.
7. Essex B, Healy M: Evaluation of a rule base for decision making in general practice. British Journal of General Practice, 1994, 44. – P. 211 – 213.
8. Laupacis A, Secar N, Stiell IG: Clinical prediction rules: A review and suggested modifications of methodological standards. JAMA 1997; 277. – P. 88 – 494. Lea A. Segel, Mathematical Models in Molecular and Cellular Biology. CUP Archive, 1980 – 757 pp.
9. Stiell IG, Wells GA: Methodologic Standards for the Development of Clinical Decision Rules in Emergency Medicine. Annals of Emergency Medicine 1999; 33:4. – P. 437 – 447.
10. Yvonne Koch, Thomas Wolf, Peter K. Sorger, Roland Eils, Benedikt Brars, Decision-Tree Based Model Analysis for Efficient Identification of Parameter Relations Leading to Different Signaling States / PLOS ONE | www.plosone.org, December 2013, Volume 8, Issue 12, e82593.

УДК.539.3

Грінченко Л. Г.

Луцький національний технічний університет

ПОПЕРЕЧНИЙ ЗГИН ТОНКОЇ ІЗОТРОПНОЇ ПРЯМОКУТНОЇ ПЛИТИ, ЖОРСТКО ЗАКРІПЛЕНОЇ НА КОНТУРІ

Грінченко Л.Г. Поперечний згин тонкої ізотропної прямокутної плити, жорстко закріпленої на контурі. В роботі розглянутий згин тонкої ізотропної прямокутної плити, з певними граничними умовами. Через базові функції і функції форми визначені: функція прогину, кути поворотів нормалей. Проведено числові розрахунки розподілу, прогину та згинних моментів в головних перерізах плити.

Ключові слова: плита, напруження, згин.

Гринченко Л.Г. Поперечный изгиб тонкой изотропной прямоугольной плиты, жестко закрепленной на контуре. В работе рассмотрен изгиб тонкой изотропной прямоугольной плиты, с определенными граничными условиями. Через базовые функции и функции формы определены: функция прогиба, углы поворота нормалей. Проведены числовые расчеты для распределения прогиба и сгинальных моментов в главных разрезах плиты.

Ключевые слова: плита, напряжение, изгиб.

Grinchenko L.G. Transverse bend of thin isotope rectangular plate, rigidly fixed in the contour. In the work is studied the bend of thin isotope rectangular plate, with definite limiting conditions. Through the basic functions and the functions of the form are determined function of deflexion and angles of normal bending. Numerical calculations of distribution, bending and curving moments in main cutting of the plate.

Keywords: flag, tension, bend.

Розглянемо тонку прямокутну ізотропну плиту з розмірами в плані $2a_j$ ($j = 1, 2$) і товщиною h жорстко закріплену на контурі і завантажену на верхній поверхні сталим навантаженням $q(x_1, x_2) = q_0$. Згин такої пластини описується диференціальним рівнянням четвертого порядку [1-4]:

$$\frac{\partial^4 w}{\partial x_1^4} + 2 \frac{\partial^4 w}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 w}{\partial x_2^4} = \frac{q_0}{D}; \quad (1)$$

Тут $w(x_1, x_2)$ – прогин плити; $D = \frac{Eh^3}{12(1-\nu^2)}$ – її циліндрична жорсткість; де E і ν відповідно модуль Юнга і коефіцієнт Пуассона матеріалу плити.

Моменти M_{11} , M_{22} в плиті визначаються за формулами:

$$M_{11} = -D \left(\frac{\partial^2 w}{\partial x_1^2} + \nu \frac{\partial^2 w}{\partial x_2^2} \right), \quad M_{22} = -D \left(\frac{\partial^2 w}{\partial x_2^2} + \nu \frac{\partial^2 w}{\partial x_1^2} \right) \quad (2)$$

через функцію прогину $w(x_1, x_2)$. Для розрахунку напружено-деформованого стану плити необхідно знати функцію прогину $w(x_1, x_2)$. Знаходимо її з розв'язку диференціального рівняння (1), задовільнивши при цьому відповідні граничні умови. Загальний розв'язок неоднорідного диференціального рівняння (1) шукаємо у вигляді суми

$$w = w_0 + w_* \quad (3)$$

$$\text{загального розв'язку } w_0 \text{ однорідного диференціального рівняння: } \frac{\partial^4 w}{\partial x_1^4} + 2 \frac{\partial^4 w}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 w}{\partial x_2^4} = 0 \quad (4)$$

і часткового розв'язку w_* неоднорідного диференціального рівняння (1). Цей розв'язок вибираємо у вигляді:

$$w_* = \frac{q_0 (x_1^2 - a_1^2) \cdot (x_2^2 - a_2^2)}{8D}. \quad (5)$$

Загальний розв'язок однорідного рівняння (4) шукаємо у вигляді [5]:

$$w_0 = \sum_{k=1}^{\infty} \left[f_k^{[1]}(x_1) \cos \delta_k^{[2]} x_2 + f_k^{[2]}(x_2) \cos \delta_k^{[1]} x_1 \right] \quad (6)$$

Підставивши вираз (6) до рівняння (4) після розділення змінних приходимо до системи двох незв'язних диференціальних рівнянь $f_k^{[j]^{(iv)}}(x_j) - 2\delta_k^{[3-j]^{(2)}} f_k^{[j]}(x_j) + \delta_k^{[3-j]^4} f_k^{[j]}(x_j) = 0, j=1, 2$ (7)

на невідомі функції $f_k^{[j]}(x_j)$. Часткові розв'язки даної системи вибираємо у вигляді

$$f_k^{[j]}(x_j) = R_k^{[j]*} \exp(\lambda_k^{[j]} x_j), \quad (8)$$

де $R_k^{[j]*}$, $\lambda_k^{[j]}$ невідомі параметри. Підставляючи ці розв'язки до рівнянь (7) отримуємо характеристичні рівняння на параметри $\lambda_k^{[j]}$: $\lambda_k^{[j]^4} - 2\delta_k^{[3-j]^2} \cdot \lambda_k^{[j]^2} + \delta_k^{[3-j]^4} = 0$. (9)

$$\text{Корені } \lambda_{(1,2)k}^{[j]} = \delta_k^{[3-j]}; \quad \lambda_{(3,4)k}^{[j]} = -\delta_k^{[3-j]} \quad (10)$$

цих рівнянь є кратні, тому загальний розв'язок однорідного рівняння (4) набуває вигляду:

$$w_0 = \sum_{k=1}^{\infty} \left\{ \left[R_{1(k)}^{[1]*} \exp(\delta_k^{[2]} x_1) + R_{2(k)}^{[1]*} x_1 \exp(\delta_k^{[2]} x_1) \right] \cos(\delta_k^{[2]} x_2) + \left[R_{1(k)}^{[2]*} \exp(\delta_k^{[1]} x_2) + R_{2(k)}^{[2]*} x_2 \exp(\delta_k^{[1]} x_2) \right] \cos(\delta_k^{[1]} x_1) \right\} \quad (11)$$

Оскільки розглядувана задача є симетричною то прогин плити повинен бути парною функцією змінних змінних x_1, x_2 . Представимо його у вигляді:

$$w_0 = \sum_{k=1}^{\infty} \left\{ \left[R_{1(k)}^{[1]*} ch(\delta_k^{[2]} x_1) + R_{2(k)}^{[1]*} x_1 sh(\delta_k^{[2]} x_1) \right] \cos(\delta_k^{[2]} x_2) + \right. \\ \left. + \left[R_{1(k)}^{[2]*} ch(\delta_k^{[1]} x_2) + R_{2(k)}^{[2]*} x_2 sh(\delta_k^{[1]} x_2) \right] \cos(\delta_k^{[1]} x_1) \right\} \quad (12)$$

При збільшенні кількості наближень k експоненціальні функції суттєво зростають. Тому для кращої можливості проведення числових розрахунків розв'язок (12), нормуємо представивши його у вигляді [6]:

$$w_0 = \sum_{k=1}^{\infty} \left\{ \left[\tilde{R}_{1(k)}^{[1]} \Phi_{1(k)}^{[1]}(x_1) + \tilde{R}_{2(k)}^{[1]} \Phi_{2(k)}^{[1]}(x_1) \right] C_k^{[2]}(x_2) + \right. \\ \left. + \left[\tilde{R}_{1(k)}^{[2]} \Phi_{1(k)}^{[2]}(x_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(x_2) \right] \cdot C_k^{[1]}(x_1) \right\} \quad (13)$$

де

$$\Phi_{1(k)}^{[j]}(x_j) = \frac{ch(\delta_k^{[3-j]} x_j)}{\exp(\delta_k^{[3-j]} a_j)}; \quad \Phi_{2(k)}^{[j]} = \frac{x_j}{a_j} \cdot \frac{sh(\delta_k^{[3-j]} x_j)}{\exp(\delta_k^{[3-j]} a_j)} \quad (14)$$

$$\tilde{R}_{1(k)}^{[j]} = R_{1(k)}^{[j]*} \exp(\delta_k^{[3-j]} a_j); \quad \tilde{R}_{2(k)}^{[j]} = R_{2(k)}^{[j]*} \cdot a_j \exp(\delta_k^{[3-j]} a_j); \\ C_k^{[j]}(x_j) = \cos(\delta_k^{[3-j]} x_j); \quad \delta_k^{[j]} = \frac{(2k-1)\pi}{2a_j}; \quad \delta_k = \frac{(2k-1)\pi}{2}. \quad (15)$$

Загальний розв'язок неоднорідного рівняння (1) отримуємо за формулою (3), враховуючи співвідношення (5), (13). Подамо його в безрозмірних величинах

$$w = \frac{q_0 a_1^4}{D} \sum_{k=1}^{\infty} \left\{ \left[\tilde{R}_{1(k)}^{[1]} \Phi_{1(k)}^{[1]}(x_1) + \tilde{R}_{2(k)}^{[1]} \Phi_{2(k)}^{[1]}(x_1) \right] C_k^{[2]}(x_2) + \right. \\ \left. + \left[\tilde{R}_{1(k)}^{[2]} \Phi_{1(k)}^{[2]}(x_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(x_2) \right] C_k^{[1]}(x_1) + \frac{(x_1^2/a_1^2 - 1)(x_2^2/a_2^2 - 1)}{8} \right\}. \quad (16)$$

Диференціюючи функцію прогину $w(x_1, x_2)$ по змінних x_1, x_2 отримуємо вирази на кути поворотів нормалей до серединної поверхні плити після деформації

$$\varphi_1 = \frac{\partial w}{\partial x_1} = \frac{q_0 a_1^4}{D} \sum_{k=1}^{\infty} \left\{ \left[\tilde{R}_{1(k)}^{[1]} \Phi_{1(k)}^{[1]'}(x_1) + \tilde{R}_{2(k)}^{[1]} \Phi_{2(k)}^{[1]'}(x_1) \right] C_k^{[2]}(x_2) + \right. \\ \left. + \left[\tilde{R}_{1(k)}^{[2]} \Phi_{1(k)}^{[2]}(x_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(x_2) \right] C_k^{[1]'}(x_1) + \frac{x_1(x_2^2/a_2^2 - 1)}{4} \right\}. \quad (17)$$

$$\varphi_2 = \frac{\partial w}{\partial x_2} = \frac{q_0 a_1^4}{D} \sum_{k=1}^{\infty} \left\{ \left[\tilde{R}_{1(k)}^{[1]} \Phi_{1(k)}^{[1]}(x_1) + \tilde{R}_{2(k)}^{[1]} \Phi_{2(k)}^{[1]}(x_1) \right] C_k^{[2]'}(x_2) + \right. \\ \left. + \left[\tilde{R}_{1(k)}^{[2]} \Phi_{1(k)}^{[2]'}(x_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]'}(x_2) \right] C_k^{[1]}(x_1) + \frac{x_2(x_1^2/a_1^2 - 1)a_2^2}{4a_1^2} \right\}. \quad (18)$$

$$\text{Невідомі сталі } x_j = \pm a_j \text{ маємо граничні умови: } w \Big|_{x_i = a_i} = 0; \quad \varphi_i = \frac{\partial w}{\partial x_i} \Big|_{x_i = a_i} = 0; \quad (19)$$

де $\varphi_i(x_i)$ кут повороту грані $x_i = a_i$.

Для зручності обчислень запишемо отримані співвідношення в безрозмірних змінних

$$\xi_j = x_j / a_j \quad (-1 \leq \xi_j \leq 1) \\ w = \frac{q_0 a_1^4}{D} \sum_{k=1}^K \left\{ \left[R_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]}(\xi_1) + R_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]}(\xi_1) \right] \cdot C_k^{[2]}(\xi_2) + \right. \\ \left. + \left[R_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(\xi_2) + R_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]}(\xi_2) \right] \cdot C_k^{[1]}(\xi_1) \right\} + \frac{q_0 \cdot a_2^2 (\xi_1^2 - 1)(\xi_2^2 - 1)}{8a_1^2}, \\ \frac{\partial w}{\partial x_1} = \left(\frac{q_0 a_1^4}{D} \right) \left[\frac{1}{a_1} \sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]'}(\xi_1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]'}(\xi_1) \right] \cdot C_k^{[2]}(\xi_2) + \right. \right. \\ \left. \left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(\xi_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]}(\xi_2) \right] \cdot C_k^{[1]'}(\xi_1) \right\} + \frac{a_2^2}{8a_1^2} (\xi_2^2 - 1) 2\xi_1 \cdot \frac{1}{a_1} \right]; \quad (20)$$

$$\frac{\partial w}{\partial x_2} = \left(\frac{q_0 a_1^4}{D} \right) \left[\frac{1}{a_2} \sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]}(\xi_1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]}(\xi_1) \right] \cdot C_k^{[2]'}(\xi_2) + \right. \right. \\ \left. \left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]'}(\xi_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]'}(\xi_2) \right] \cdot C_k^{[1]}(\xi_1) \right\} + \frac{a_2^2}{8a_1^2} (\xi_1^2 - 1) 2\xi_2 \cdot \frac{1}{a_2} \right];$$

$$\text{де: } R_{1(k)}^{[j]} = \frac{\tilde{R}_{1(k)}^{[j]}}{\left(\frac{q_0 a_1^4}{D} \right)}; \quad R_{2(k)}^{[j]} = \frac{\tilde{R}_{2(k)}^{[j]}}{\left(\frac{q_0 a_1^4}{D} \right)};$$

$$\Phi_{1(k)}^{[j]}(\xi_j) = \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] + \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right]; \quad (21)$$

$$\Phi_{2(k)}^{[j]}(\xi_j) = \frac{1}{2} \xi_j \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] - \frac{1}{2} \cdot \xi_j \cdot \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right];$$

$$\Phi_{1(k)}^{[j]'}(\xi_j) = \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_1) \right] \cdot \delta_k \frac{a_j}{a_{3-j}} - \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right] \delta_k \cdot \frac{a_j}{a_{3-j}};$$

$$\Phi_{2(k)}^{[j]'}(\xi_j) = \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] + \frac{1}{2} \xi_2 \cdot \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_{j2}) \right] \delta_k \frac{a_j}{a_{3-j}} - \quad (22)$$

$$- \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_2}{a_1} (1 + \xi_2) \right] + \frac{1}{2} \xi_2 \exp \left[-\delta_k \cdot \frac{a_2}{a_1} (1 + \xi_2) \right] \cdot \delta_k \cdot \frac{a_2}{a_1};$$

$$\text{Підставляючи вирази (17), (18), (20) до граничних умов } w|_{\xi_1=1} = 0; \quad \varphi_1 = \frac{\partial w}{\partial x_1} \Big|_{\xi_1=1} = 0;$$

$$w|_{\xi_2=1} = 0; \quad \varphi_2 = \frac{\partial w}{\partial x_2} \Big|_{\xi_2=1} = 0; \quad (23)$$

і враховуючи, що $C_k^{[1]}(1) = C_k^{[2]}(1) = 0$ отримуємо систему $2k + 2$ алгебраїчних рівнянь на невідомі коефіцієнти $\tilde{R}_{v(k)}^{[j]}$,

$$v=1,2 \quad \tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]}(1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]}(1) = 0;$$

$$\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(1) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]}(1) = 0;$$

$$\sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]'}(1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]'}(1) \right] \cdot C_k^{[2]}(\xi_2) + \right. \quad (24)$$

$$\left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(\xi_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]}(\xi_2) \right] \cdot C_k^{[1]'}(1) \right\} + \frac{a_2^2}{4a_1^2} (\xi_2^2 - 1) = 0;$$

$$\sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]}(\xi_1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]}(\xi_1) \right] \cdot C_k^{[2]'}(1) + \right.$$

$$\left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]'}(1) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]'}(1) \right] \cdot C_k^{[1]}(\xi_1) \right\} + \frac{a_2^2}{4a_1^2} (\xi_1^2 - 1) = 0;$$

З перших $2k$ рівнянь системи (24) знаходимо:

$$R_{2(k)}^{[j]} = - \frac{\Phi_{1(k)}^{[j]}(1)}{\Phi_{2(k)}^{[j]}(1)} R_{1(k)}^{[j]} \quad ; \quad (25)$$

Підставляючи ці зв'язки до інших двох рівнянь даної системи отримуємо систему двох рівнянь зі змінними коефіцієнтами на визначення невідомих коефіцієнтів $R_{1(k)}^{[1]}, R_{2(k)}^{[2]}$

$$\sum_{k=1}^K \left\{ R_{1(k)}^{[1]} \left[\Phi_{1(k)}^{[1]'}(1) - \frac{\Phi_{1(k)}^{[1]}(1)}{\Phi_{2(k)}^{[1]}(1)} \Phi_{2(k)}^{[1]'}(1) \right] \cdot C_k^{[2]}(\xi_2) + \right. \quad (26)$$

$$\left. + R_{2(k)}^{[2]} \left[\Phi_{1(k)}^{[2]}(\xi_2) - \frac{\Phi_{1(k)}^{[2]}(1)}{\Phi_{2(k)}^{[2]}(1)} \Phi_{2(k)}^{[2]}(\xi_2) \right] \cdot C_k^{[1]'}(1) \right\} + \frac{a_2^2}{4a_1^2} (\xi_2^2 - 1) = 0;$$

$$\sum_{k=1}^K \left\{ R_{1(k)}^{[1]} \left[\Phi_{1(k)}^{[1]}(\xi_1) - \frac{\Phi_{1(k)}^{[1]}(1)}{\Phi_{2(k)}^{[1]}(1)} \Phi_{2(k)}^{[1]}(\xi_1) \right] \cdot C_k^{[2]'}(1) + \right. \\ \left. + R_{1(k)}^{[2]} \left[\Phi_{1(k)}^{[2]'}(1) - \frac{\Phi_{1(k)}^{[1]}(1)}{\Phi_{2(k)}^{[1]}(1)} \cdot \Phi_{2(k)}^{[2]'}(1) \right] \cdot C_k^{[1]}(\xi_1) \right\} + \frac{a_2^2}{4a_1^2} (\xi_1^2 - 1) = 0; \quad (27)$$

Систему рівнянь (26), (27) розв'язуємо методом колокацій. Для цього проміжок $\xi_1 \in [0;1]$ та $\xi_2 \in [0;1]$ розбиваємо на ряд інтервалів довжиною $\frac{1}{K}$ і записуємо цю систему рівнянь для кожної точки:

$$\xi_1 = 0; \xi_1 = \frac{1}{K}; \xi_1 = \frac{2}{K}; \dots; \xi_1 = \frac{K-1}{K} \text{ та } \xi_2 = 0; \xi_2 = \frac{1}{K}; \xi_2 = \frac{2}{K}; \dots; \xi_2 = \frac{K-1}{K}.$$

Кількість точок колокацій (параметр K) вибираємо так, щоб досягти заданої точності задоволення системи рівнянь (26), (27). В рахунках покладено $\nu = 0,3$. Точність задоволення граничних умов перевіряли за максимальним відхиленням від нуля лівих частин рівнянь (26), (27) при відомих уже коефіцієнтах $R_{v(k)}^{[j]}$, табулюючи їх як функції від ξ_1, ξ_2 з кроком $H = 0.01$. В результаті табуляції встановлено, що для десяти точок колокацій ($K = 10$) максимальне відхилення від нуля за абсолютною величиною на всьому інтервалі $\xi_j \in [-1;1]$ у випадку, коли відношення сторін плити задовільняє умові: $\frac{a_1}{a_2} = 0.5; 0.6; 0.7; 0.8; 0.9; 1.0$ для рівняння (26) не перевищує $9.551644 \cdot 10^{-5}$, для рівняння (27) не перевищує $4.864484 \cdot 10^{-5}$. У випадку відношення $\frac{a_1}{a_2} = 1.11; 1.25; 1.43; 1.67; 2.00$ максимальне відхилення від нуля за абсолютною величиною для рівняння (26) не перевищує $4.856289 \cdot 10^{-5}$, для рівняння (27) не перевищує $9.549409 \cdot 10^{-5}$, тобто точність задоволення граничних умов є достатньо висока. Знаючи параметри $R_{v(k)}^{[j]}$ визначаємо прогин

$w(x_1, x_2)$ в кожній точці плити, а маючи функцію прогину за формулами (2) легко знаходимо моменти в плиті. Запишемо вирази для кривин волокон через які визначаються згинні моменти в

$$\frac{\partial^2 w}{\partial x_1^2} = \left(\frac{q_0 a_1^4}{D} \right) \left[\frac{1}{a_1^2} \sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]''}(\xi_1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]''}(\xi_1) \right] \cdot C_k^{[2]}(\xi_2) + \right. \right. \\ \left. \left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]}(\xi_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]}(\xi_2) \right] \cdot C_k^{[1]''}(\xi_1) \right\} + \frac{a_2^2}{8a_1^2} (\xi_2^2 - 1) 2 \cdot \frac{1}{a_1^2} \right]; \\ \frac{\partial^2 w}{\partial x_2^2} = \left(\frac{q_0 a_1^4}{D} \right) \left[\frac{1}{a_2^2} \sum_{k=1}^K \left\{ \left[\tilde{R}_{1(k)}^{[1]} \cdot \Phi_{1(k)}^{[1]}(\xi_1) + \tilde{R}_{2(k)}^{[1]} \cdot \Phi_{2(k)}^{[1]}(\xi_1) \right] \cdot C_k^{[2]''}(\xi_2) + \right. \right. \\ \left. \left. + \left[\tilde{R}_{1(k)}^{[2]} \cdot \Phi_{1(k)}^{[2]''}(\xi_2) + \tilde{R}_{2(k)}^{[2]} \cdot \Phi_{2(k)}^{[2]''}(\xi_2) \right] \cdot C_k^{[1]}(\xi_1) \right\} + \frac{a_2^2}{8a_1^2} (\xi_1^2 - 1) 2 \cdot \frac{1}{a_2^2} \right]; \quad (28)$$

Де

$$\Phi_{1(k)}^{[j]}(\xi_j) = \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] \cdot (\delta_k)^2 \cdot \left(\frac{a_j}{a_{3-j}} \right)^2 + \frac{1}{2} \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right] \cdot (\delta_k)^2 \cdot \left(\frac{a_j}{a_{3-j}} \right)^2; \\ \Phi_{2(k)}^{[j]}(\xi_j) = \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] \delta_k \cdot \frac{a_j}{a_{3-j}} + \frac{1}{2} \xi_j \cdot \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 - \xi_j) \right] (\delta_k)^2 \cdot \left(\frac{a_j}{a_{3-j}} \right)^2 + \\ + \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right] \cdot \delta_k \cdot \frac{a_j}{a_{3-j}} - \frac{1}{2} \xi_j \exp \left[-\delta_k \cdot \frac{a_j}{a_{3-j}} (1 + \xi_j) \right] \cdot (\delta_k)^2 \cdot \left(\frac{a_j}{a_{3-j}} \right)^2;$$

проведені числові розрахунки розподілу прогину та згинних моментів в головних перерізах ($\xi_j = 0$) плити. Результат обчислень графічно представлені на рис. 1, 3, 5, 7, 9.

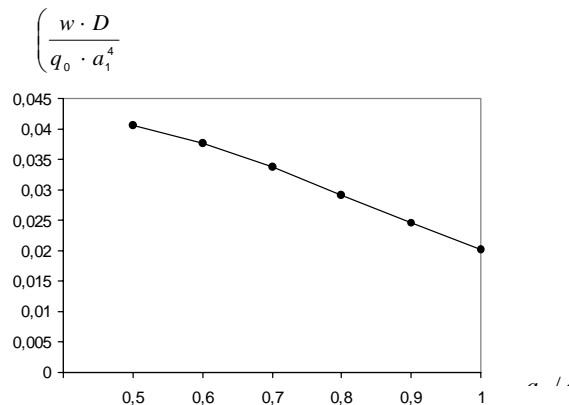


Рис. 1. Зміна максимального прогину плити в залежності від співвідношення її сторін.

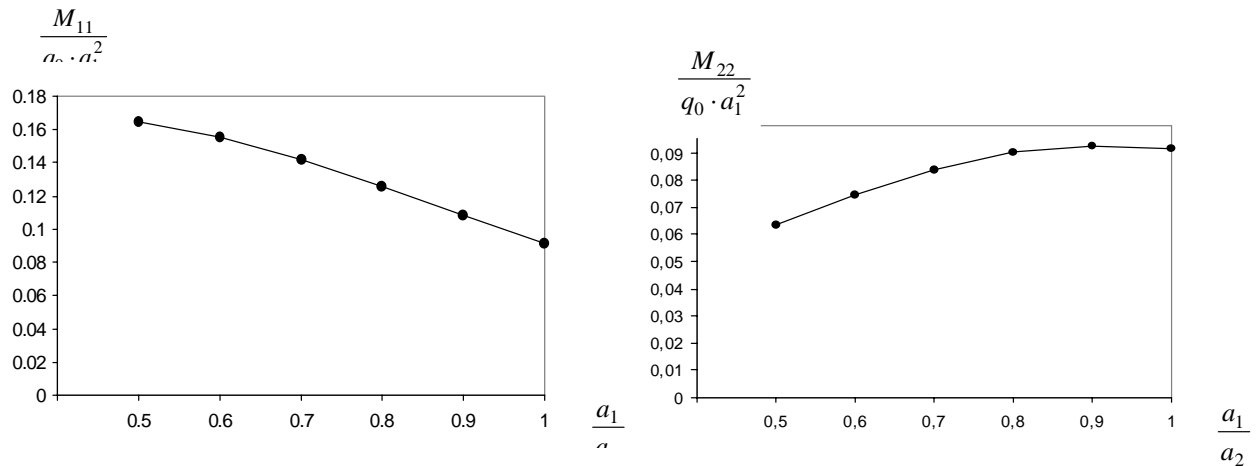


Рис. 2. Зміна максимального згинного моменту $M_{11} / q_0 a_1^2$ в центрі плити в залежності від співвідношення її сторін. Рис. 3. Зміна максимального згинного моменту $M_{22} / q_0 a_1^2$ в центрі плити в залежності від співвідношення її сторін.

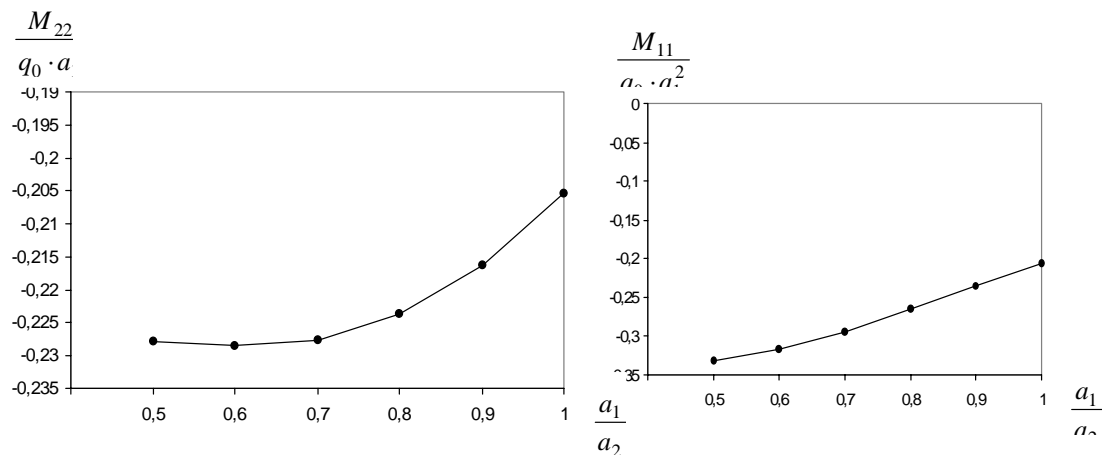


Рис. 4. Зміна максимального згинного моменту $M_{22} / q_0 a_1^2$ на краю плити ($x_2 = a_2; x_1 = 0$) в залежності від співвідношення її сторін. Рис. 5. Зміна максимального згинного моменту $M_{11} / q_0 a_1^2$ на краю плити ($x_1 = a_1; x_2 = 0$) в залежності від співвідношення її сторін. На рисунках 1-5 представлені графіки зміни відносних величин прогину w , та згинних моментів M_{11}, M_{22} в центрі плити та на її краях в залежності від співвідношення сторін a_1 / a_2 . Залежність максимального прогину плити (в її центрі) від співвідношення a_1 / a_2 ілюструє рисунок 1. Із збільшенням a_1 / a_2 прогин падає і досягає найменшого значення для квадратної плити. Графіки зміни моментів M_{11} (рис.2) та M_{22} (рис.3) в центрі плити мають протилежний характер: збільшення a_1 / a_2 приводить до зростання моменту M_{22} і падіння моменту M_{11} . Із збільшенням відношення a_1 / a_2 згинний момент M_{11} на краю плити ($x_1 = a_1, x_2 = a_2$) суттєво зростає (рис.4) в той же час згинний момент M_{22} (рис.5) майже не залежить від співвідношення сторін.

1. Тимошенко С.П., Войновски-Кригер. Пластины и оболочки.-М.: Наука, 1966.- 635с.
2. Бережницкий Л.Т., Делявский М.В., Панасюк В.В. Изгиб тонких пластин с дефектами типа трещин.-Киев: Наукова думка, 1979.- 400с.
3. Прусов И.А. Метод сопряжения в теории плит.-Минск: Изд-во Белорус. ун-та, 1975.- 256 с.
4. Kączkowski Z. Płyty. Obliczenia statyczne.- Warszawa: Arkady, 1980.-
5. Делявский М.В. Расчет напряженного состояния в толстой ортотропной плите под действием изгибающей нагрузки // Пробл. прочности .- № 11-12.- С. 117-123.
6. Delyavskyy M., Gołaś J., Podhorecka A.: O pewnym podejściu do rozwiązywania płyt wielowarstwowych, XLV Konf. Nauk. Krynica 99, s.63-70.

УДК 515.2

Ісмаїлова Н.П., к.т.н., доц.

Одеська державна академія будівництва та архітектури

ТВЕРДОТІЛЬНЕ МОДЕЛЮВАННЯ СПРЯЖЕНИХ ПОВЕРХОНЬ НА БАЗІ ПАРАМЕТРИЧНОГО КІНЕМАТИЧНОГО ГВИНТА

Ісмаїлова Н.П. Твердотільне моделювання спряжених поверхонь «Евольвентний гелікоїд – конус обертання» на базі параметричного кінематичного гвинта. Запропоновано комп'ютерне твердотільне моделювання спряжених поверхонь «Евольвентний гелікоїд – конус обертання» на базі параметричного кінематичного гвинта в системі автоматизованого проектування САПР Autodesk Inventor.

Ключові слова: спряжені поверхні, твердотільне моделювання, кінематичний гвинт.

Исмаилова Н.П. Твердотельное моделирование сопряженных поверхностей «Эвольвентный геликоид – конус вращения» на базе параметрического кинематического винта. Предложено компьютерное моделирование сопряженных поверхностей «Эвольвентный геликоид – конус вращения» на базе параметрического кинематического винта в системе автоматизированного проектирования САПР Autodesk Inventor.

Ключевые слова: сопряженные поверхности, твердотельное моделирование, кинематический винт.

Ismailova N.P. Solid design of the attended surfaces of «Evol'ventnyy a helicoid is a cone of rotation» on the base of self-reactance kinematics screw. Offered computer design of the attended surfaces of «Evol'ventnyy a helicoid is a cone of rotation» on the base of self-reactance kinematics screw in a computer-aided design CADD Autodesk Inventor.

Keywords: attended surfaces, design, kinematics screw.

Постановка наукової проблеми. Проектування спряжених криволінійних поверхонь деталей в машинобудуванні вельми трудомісткий процес. Комп'ютерне твердотільне геометричне моделювання деталей в машинобудуванні для обробки складних спряжених криволінійних поверхонь підвищує продуктивність розрахунково-конструкторських робіт і точність профілювання.

Аналіз досліджень. При профілюванні спряжених поверхонь деталей в машинобудуванні застосовувались трохідографи та електронно-моделюючі установки [1], які мали низьку точність та продуктивність побудови сімейства трохіодів.

Комп'ютерне твердотільне моделювання спряжених криволінійних поверхонь деталей в машинобудуванні складної форми вирішує проблему підвищення точності профілювання та продуктивності праці конструктора. Метою даного дослідження є розробка комп'ютерного твердотільного геометричного моделювання спряжених криволінійних поверхонь деталей в машинобудуванні.

Основна частина. Розглянутий кінематичний метод і алгоритм формування спряжених поверхонь стосовно моделювання деталей машинобудування на верстатах з числовим програмним управлінням. Кінематичний гвинт добре себе зарекомендував для формування спряжених лінійчатих поверхонь [3], тому встало актуальне завдання досліджувати можливість застосування кінематичного параметричного гвинта для спряжених криволінійних поверхонь. Методика складання моделювання просторового параметричного кінематичного гвинта заснована на теоремах професора А.М.Подкоритова.

Тривимірна модель кінематичного гвинта виконана в САПР Autodesk Inventor, що є системою параметричного моделювання, тобто дозволяє довільно міняти розміри проектованої конструкції, при цьому зберігаючи її конфігурацію і цілісність.

Параметричний кінематичний гвинт - це просторове зображення трьох гвинтових рухів, з яких два є гвинтовими, а третє - результуючим [1]. За допомогою кінематичного гвинта визначаються наступні параметри:

Алгоритм побудови:

1. АВ - відстань між мимобіжними осями і та j (задана вручну);
2. h_1 - крок гелікоїда Σ_A (заданий вручну);
3. h_2 - крок гелікоїда Σ_B (заданий вручну);

4. γ - кут між мимобіжними осями i та j (заданий вручну);
5. ω_A - швидкість обертання осі i (задана вручну);
6. ω_B - швидкість обертання осі j (задана вручну).

Необхідно визначити:

7. a - відстань між мимобіжними осями i та m (отримана автоматично);
8. b - відстань між мимобіжними осями j та m (отримана автоматично);
9. f - передатне відношення ($f=a/b$);
10. α - кут між мимобіжними осями i та m (отриманий автоматично);
11. β - кут між мимобіжними осями j та m (отриманий автоматично);
12. h_m - крок гелікоїда Φ (отриманий автоматично);
13. ω_m - швидкість обертання осі m (отримана автоматично).

Таким чином, для формування діаграми просторового параметричного кінематичного гвинта в системі САПР Autodesk Inventor достатньо задати параметри AB , h_1 , h_2 , γ , ω_A і ω_B (на рис.1), решта параметрів отримана автоматично (рис.1).

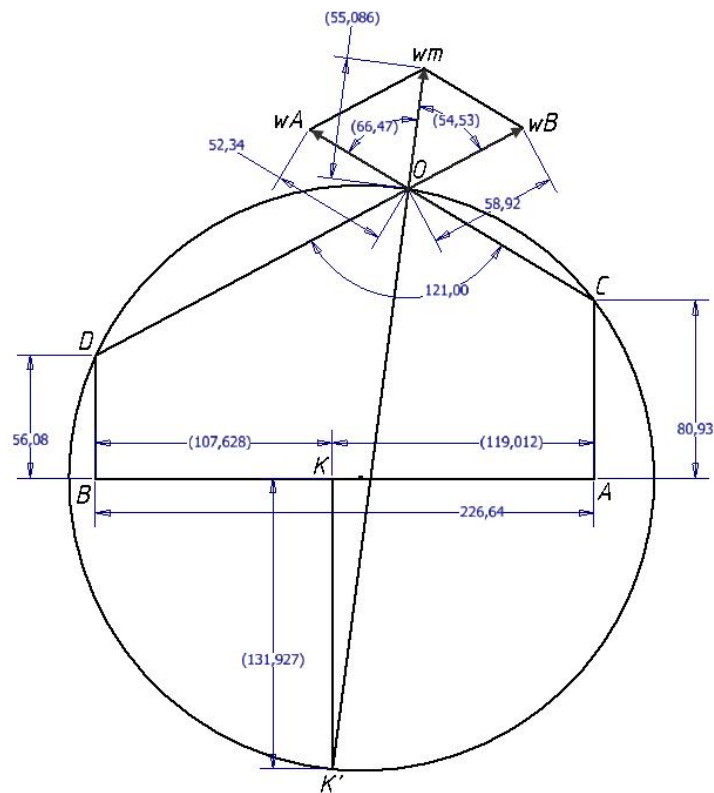


Рис.1 Параметрический кинематический винт

При формуванні просторової моделі кінематичного параметричного гвинта автоматично створена і відредагована таблиця параметрів тривимірної моделі (рис.2), в якій можна змінювати значення призначених для користувача параметрів.

При цьому кожен параметр моделі приймає значення відповідного призначеного для користувача параметра, вказаного в стовпці «Формула», при цьому початковими є наступні параметри: AB , γ , ω_A , ω_B , h_1 і h_2 .

При формуванні тривимірної моделі кінематичного гвинта автоматично створена і відредагована таблиця параметрів тривимірної моделі, в якій можна змінювати значення призначених для користувача параметрів. При цьому кожен параметр моделі приймає значення відповідного призначеного для користувача параметра. При розробці підпрограми до САПР

Autodesk Inventor по формуванню тривимірної моделі кінематичного параметричного гвинта, створена тривимірна модель може бути використана як шаблон (рис. 2), при цьому такі параметри як α , β , ω_m , f , h_m , a , b можуть бути обчислені по наступним формулам,

$$\omega_m = \sqrt{\omega_A^2 + \omega_B^2 - 2\omega_A\omega_B \cos(180 - \gamma)}$$

$$\alpha = \arcsin \frac{\omega_A \sin(180 - \gamma)}{\omega_m} = \gamma - \beta$$

$$\beta = \arcsin \frac{\omega_B \sin(180 - \gamma)}{\omega_m} = \gamma - \alpha$$

$$a = \frac{AB\omega_A \cos \alpha}{\omega_m}, \text{ если } h_1 = 0, h_2 = 0$$

$$b = \frac{AB\omega_B \cos \beta}{\omega_m}, \text{ если } h_1 = 0, h_2 = 0$$

$$f = \frac{a}{b}$$

$$h_m = b \cdot \operatorname{tg} \alpha = a \cdot \operatorname{tg} \beta, \text{ если } h_1 = 0, h_2 = 0$$

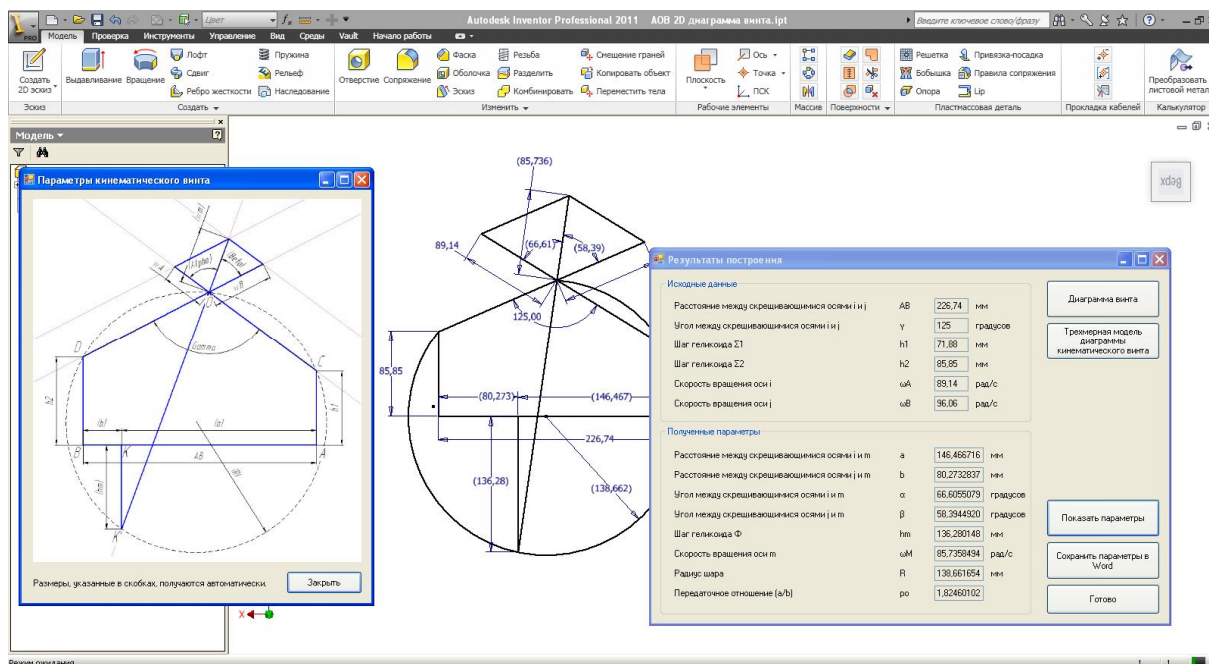


Рис.3. Диаграмма пространственного параметрического кинематического винта

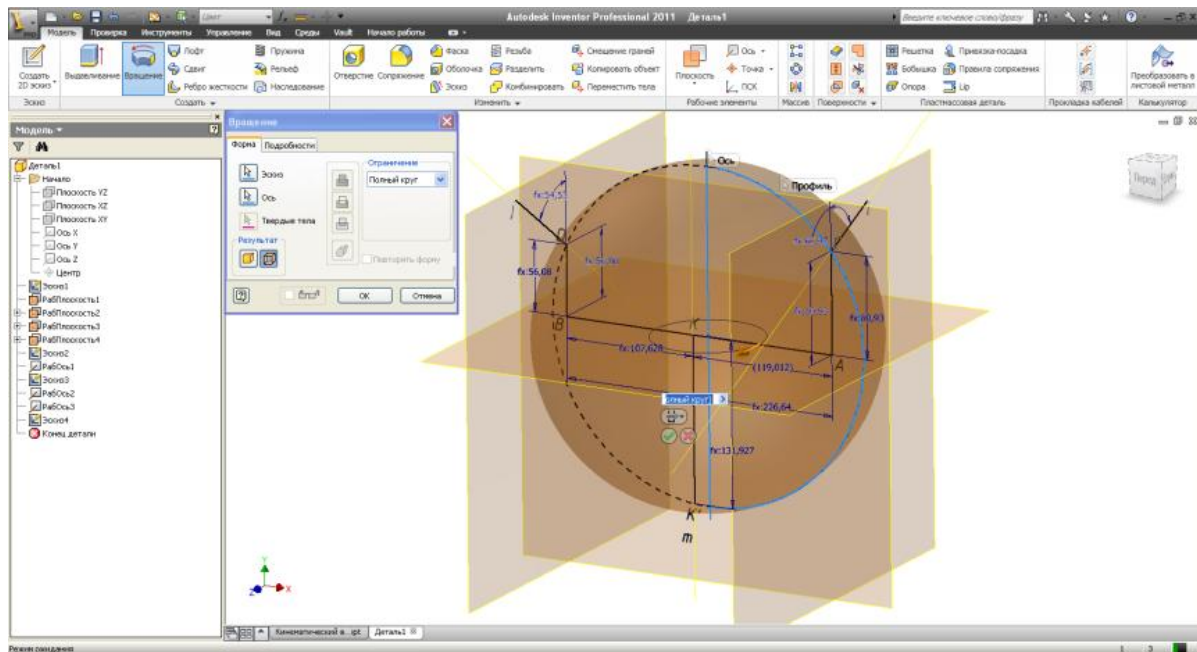


Рис.4. Тривимірна модель параметричного кінематичного гвинта.

Зображенню параметричного кінематичного гвинта відповідає сукупність послідовно виконуваних геометричних побудов. Представлений алгоритм програми (рис.5) описує формування діаграми просторового кінематичного гвинта і витягання з неї геометричних параметрів, необхідних для побудови твердотілого моделювання «Евольвентний гелікоїд – конус обертання».

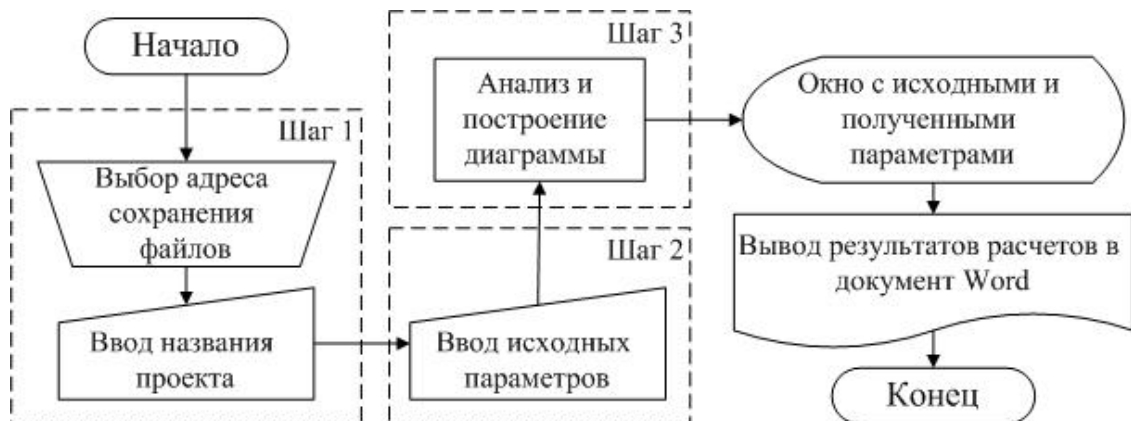


Рис. 5. Алгоритм программы

На базі розробленого просторового параметричного кінематичного гвинта побудовано твердотіле моделювання (рис.4,5), спряжених поверхонь «Евольвентний гелікоїд – конус обертання» за допомогою параметричного кінематичного гвинта.

Для евольвентного гелікоїда виконується співвідношення

$$a = h \cdot \operatorname{tg} \alpha \quad (1)$$

Приймаємо $a=30$ мм, $\alpha=35^\circ$, $\gamma=50^\circ$.

Із співвідношення (1) отримано значення h :

$$h = \frac{a}{\operatorname{tg} \alpha} = \frac{30}{\operatorname{tg} 35^\circ} = \frac{30}{0.7002} = 42.8449 \text{ мм}$$

$$\beta = \gamma - \alpha = 50^\circ - 35^\circ = 15^\circ$$

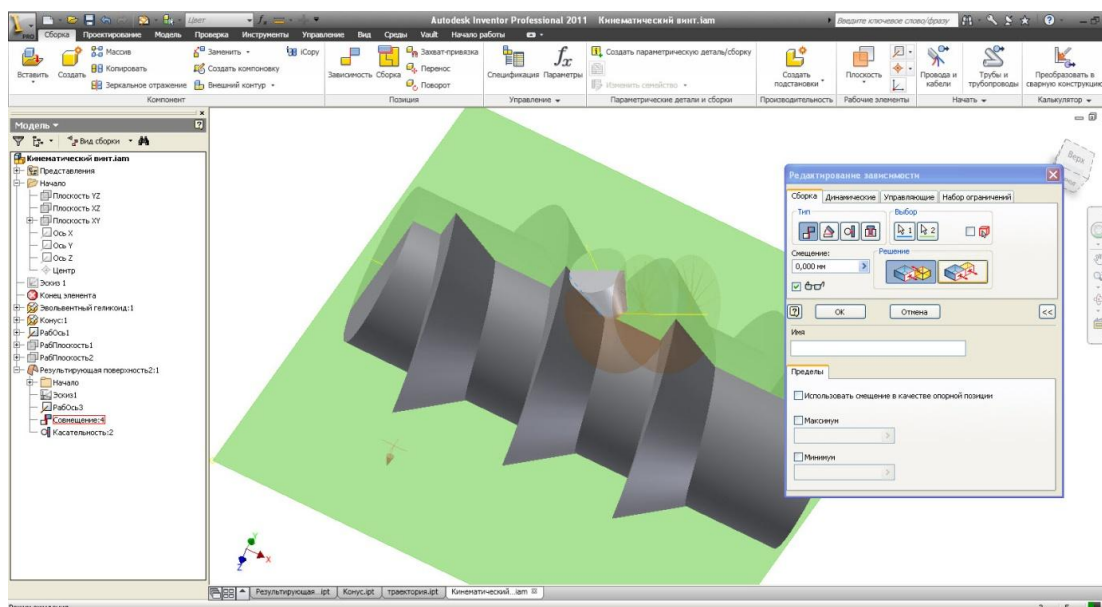


Рис.4. «Евольвентный геликоид – конус обертання» за допомогою параметричного кінематичного гвинта

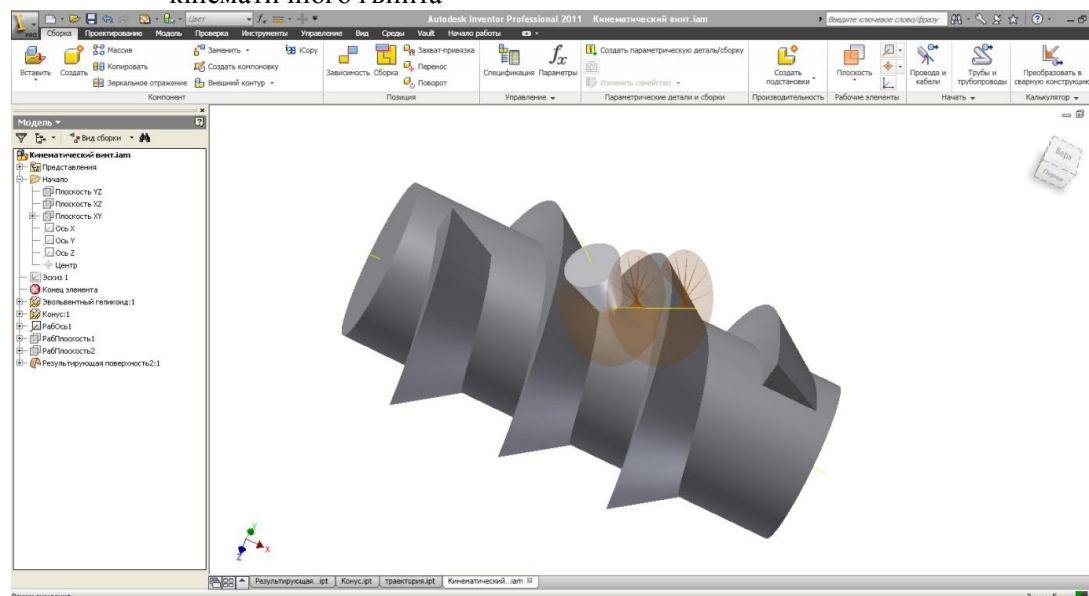


Рис.4. «Евольвентный геликоид – конус обертання» за допомогою параметричного кінематичного гвинта

У цей час, практично всі проектні рішення виконуються за допомогою САПР. Це дозволяє істотно скоротити час на проектування. Але не завжди за допомогою стандартних засобів комп'ютерного моделювання можна вирішити поставлене завдання. У зв'язку із цим виникає необхідність у створенні спеціальних підпрограм для побудови деяких видів поверхонь. Одними з таких поверхонь є спряжені поверхні, тому що в них криволінійні й напрямна, і утворююча.

Сосіб комп'ютерного моделювання геометричної просторової моделі параметричного кінематичного гвинта, дозволяють вирішити складні завдання конструювання спряжених

поверхонь, підвищити точність і продуктивність інженерної праці і створювати конкурентно-здатні вироби в машинобудуванні, літакобудуванні, кораблебудуванні.

Висновки та перспективи подальшого дослідження. У результаті проведених досліджень ми зробили комп'ютерне параметричне трьохмірне твердотільне моделювання спряжених поверхонь за допомогою просторового параметричного кінематичного гвинта. А також наблизились до рішення проблеми підвищення точності профілювання і підвищення продуктивності конструкторських робіт. Подальші дослідження проводитимуться у напрямі розширення побудови спряжених квазігвинтових поверхонь на базі параметричного кінематичного гвинта, також наблизились до вирішення проблеми виключення інтерференції при профілювання деталей в машинобудуванні.

1. А.Н. Подкорытов. Исключение интерференции сопряженных поверх-ностей зубчатых передач. INTERNATIONAL CONGRES - GEAR TRANSMISSIONS, Sofia - BULGARIA, 1995г., с.143-145.
2. А.Н. Подкорытов. Автоматизация, электронное моделирование и исследование интерференции сопряженных криволинейных поверхностей на базе ЭВМ. – Омск; Зап-Сиб.кн.изд, 1976г., 168с.
3. Николаев А.Ф. Диаграмма винта и ее применение к определению сопряженных линейчатых поверхностей с линейным касанием / Тр. семинара по теории механизмов и машин. / АН СССР. Ин-т Машиноведения, т. 10. – Вып. 37. – Москва, 1950 – С. 52-106.
4. Подкоритов А.Н., Исмаилова Н.П., Дюкре Л.Г. Метод формирования сопряженных винтовых нелинейчатых поверхностей семейством огибающих геликоидов // Геометричне та комп'ютерне моделювання. – Вип..17. – ХДУХТ. – Харків, 2007. – С.12-15.

УДК 629.439.001.76

Каганюк А.К., к.т.н. доц., Желобицкий Я.К.

Луцкий национальный технический университет

ВНЕДРЕНИЕ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ СТРУННОГО ТРАНСПОРТА ЮНИЦКОГО «SKYWAY» В СУЩЕСТВУЮЩУЮ ТРАНСПОРТНУЮ СИСТЕМУ

Каганюк А.К., Желобицкий Я.К. Внедрение инновационных технологий струнного транспорта Юницкого «Skyway» в существующую транспортную систему. В статье рассмотрены инновационные технологии струнного транспорта Юницкого «Skyway», существующие транспортные технологии, а также современные системы автоматического управления дорожным движением. Раскрыта важность реформации существующих транспортных технологий, и предложена возможность внедрения технологий транспорта Юницкого в них путём создания абсолютно новой системы функционирования транспорта. Проведены сравнительные характеристики технологии «Skyway» и ныне существующей транспортной системы. Также, проведены финансовые расчёты для создания струнного транспорта. Предложена потребность внедрения АСУ ДД в «Skyway».

Ключевые слова: струнный, Юницкий, «Skyway», транспортные технологии, АСУ ДД.

Каганюк О.К., Желобицкий Я.К. Впровадження інноваційних технологій струнного транспорту Юницького «Skyway» в існуючу транспортну систему. У статті розглянуто інноваційні технології струнного транспорту Юницького «Skyway», існуючі транспортні технології, а також сучасні системи автоматичного управління дорожнім рухом. Раскрыта важливість реформації існуючих транспортних технологій, і запропонована можливість впровадження технологій транспорту Юницького в них шляхом створення абсолютно нової системи функціонування транспорту. Проведено порівняльні характеристики технології «Skyway» і нині існуючої транспортної системи. Також, проведено фінансові расчёт для створення струнного транспорту. Запропоновано потреба впровадження АСК ДР в «Skyway».

Ключові слова: струнний, Юницький, «Skyway», транспортні технології, АСК ДР

Kaganyuk A.K., Zhelobitskiy Y.K. Introduction of innovative technologies String Transport Unitsky «Skyway» the existing transport system. The article examines innovative technologies String Transport Unitsky «Skyway», existing transport technologies and modern automatic control system traffic. Rasskryta importance of reformation of existing transport technologies and offered the possibility of introducing transport technologies Unitsky them by creating a completely new system of functioning of transport. Comparative characteristics of the technology «Skyway» and now the existing transport system. Also held Calculate your financing to create a string of transport. A need for the implementation of ATMS in «Skyway».

Keywords: string, Unitsky, «Skyway», transport technology, ATMS

Постановка научной проблемы. Транспортные пути - живительные артерии современного мира. Кровеносная система глобальной экономики должна поддерживаться в тонусе и обновляться. Сегодня уровень развития технологий достиг того, что передача информации в любую точку мира занимает считанные секунды, транспортировка же людей или грузов занимает значительно больше времени. Потребность человека именно в скоростных перевозках возрастает с каждым днем.

Помимо скорости, не менее важным фактором остаётся безопасность. Проблема аварийности в последнее десятилетие приобрела особую остроту в связи с несоответствием дорожно-транспортной инфраструктуры потребностям общества и государства в безопасном дорожном движении, недостаточной эффективностью функционирования системы обеспечения безопасности дорожного движения и крайне низкой дисциплиной участников дорожного движения. Любая недостаточная надёжность одного из составляющих приводит к дорожно-транспортному происшествию. Общее количество дорожно-транспортных происшествий ежегодно составляет приблизительно 200 тыс. и это количество растёт, особенно за последние годы.

Наравне с вышеупомянутыми факторами есть ещё один не менее важный приоритетный компонент – стоимость перевозок. Перемещение пассажиров и транспортировка грузов требует более современные технологии, которые должны учитывать как быстроту, дешевизну так и экологичные способы доставки.

В настоящее время современный мир находится на грани экологической катастрофы и перед всем человечеством стоит весьма важная задача по совершенствованию технологий и транспортных экологически безопасных потоков. Химические комбинаты, фабрики, заводы не обладают достаточно эффективными методами очистки вредоносных отходов, которые еще больше способствуют загрязнению окружающей среды и, тем самым приносят непоправимый вред природе, а соответственно и самому человеку. При создании новых транспортных потоков, разрушается поверхностный почвенный слой, нарушается равновесное состояние, возникает необходимость по созданию многофункциональной инфраструктуры. Страдает живой мир, рельеф местности, биогеоценоз и биоразнообразие прилегающей территории. Расходуется больше ценных территорий с точки зрения экономической и природной.

Не менее важными проблемами на сегодняшний день являются функционирование систем АСУДД которые можно классифицировать по следующим признакам:

- вместимость подвижного транспортного и пассажирского составов;
- протяженность транспортных потоков;
- инфраструктура технического обслуживания; и многое другое.

Весьма актуальным на сегодняшний день является создание совершенно новой транспортной технологии способной по иному взглянуть на решение поставленной проблемы которая позволит разрешить ранее перечисленные недостатки в области транспортных потоков как в области экологии так и сфере обслуживания. Поэтому, данная работа будет посвящена вопросам анализа современных технологий по разрешению проблем в транспортных потоках. В данном направлении большой вклад был сделан Белорусским ученым и исследователем Анатолием Юницким [2], которая воплощается в технологии «Skyway». В данном направлении работали и другие авторы [1,6].

На основании проведенного анализа сформулируем критерии, позволяющие сформулировать цели для решения поставленных задач:

- сравнительная характеристика технологии «Skyway» в сравнении с ныне существующей;
- потребность внедрения АСУ ДД в «Skyway»;
- экономический расчёт стоимости проекта относительно существующей технологии транспортной системы.

Для сравнительного анализа используем проектное решение современных струнных технологий Анатолия Юникого «Skyway» «скоростной RSW комплекс».

Изложение основного материала и обоснование полученных результатов. Компания Euroasian Rail Skyway Systems Holding Ltd. (RSW systems) владеет инновационной технологией транспорта нового поколения «SkyWay», имеющего неоспоримые преимущества над существующими видами транспорта.

Транспорт на основе этих технологий сможет заменить собой весь городской, пригородный, междугородний транспорт и транспортные грузоперевозки. А также может объединить весь мир в одну высокоскоростную транспортную сеть.

На рис. 1 представлен один из проектов струнного транспорта Юникого «Skyway» «скоростной RSW комплекс».[2]



Рис. 1 – Один из проектов струнного транспорта Юницкого «Skyway» «скоростной RSW комплекс»

Проведем сравнительная оценка характеристик технологий струнного транспорта Юницкого «Skyway» «городской RSW комплекс» и ныне существующих:

Основные характеристики:

- 1) Скорость движения — до 500 км/ч.
- 2) Вместимость подвижного состава:
 - пассажиров — до 500 человек;
 - грузов — до 100 тонн.
- 3) Уклон пути — до 10%, при специальном исполнении — до 30%.
- 4) Расстояние перевозки — до 10 000 км.
- 5) Объём высокоскоростных междугородных перевозок:
 - пассажиров — до 1 млн. пасс./сутки;
 - грузов — до 100 тыс. т/сутки.
- 6) Стоимость высокоскоростной трассы без учёта стоимости подвижного состава, пассажирских вокзалов, станций и инфраструктуры — от 3 млн. USD/км.
- 7) Себестоимость высокоскоростных перевозок в 5 раз ниже себестоимости перевозок высокоскоростной железной дорогой, поездами на магнитной подушке, самолётом.

Конкурентные преимущества:

- 1) Уменьшение капитальных затрат на строительство:
 - за счёт значительного уменьшения изъятия земли под трассу и инфраструктуру;
 - за счёт исключения земляных насыпей, выемок, тоннелей, мостов, путепроводов, многоуровневых развязок и водопропускных сооружений;
 - за счёт исключения ограждения линейной части высокоскоростной трассы;
 - за счёт того, что пересечённый рельеф местности, низкая прочность подстилающих грунтов и сложные географические и климатические условия не приводят к удорожанию рельсо-струнной эстакады;
 - за счёт уменьшения на порядок ресурсоёмкости рельсо-струнной эстакады в сравнении с традиционными транспортными эстакадами со сплошным полотном.
- 2) Снижение эксплуатационных издержек:
 - за счёт уменьшения расхода энергии и топлива;
 - за счёт снижения расходов на обслуживающий персонал и его заработную плату;
 - за счёт снижения объёмов путевых и ремонтно-восстановительных работ;
 - за счёт исключения необходимости в зимний период времени очищать путевую структуру от наледи и снега.
- 3) Снижение себестоимости высокоскоростных перевозок пассажиров и грузов в три и более раз.
- 4) Возможность щадящего освоения и заселения новых территорий в труднодоступных местах, комфортных для проживания (острова, горы, шельф моря и др).
- 5) Повышение надёжности и безопасности всепогодной и круглогодичной эксплуатации высокоскоростной транспортной системы в любых природно-климатических условиях.
- 6) Возможность совмещения путевой структуры и опор с воздушными и кабельными линиями электропередач и линиями связи (проводными, оптоволоконными, радиорелейными, сотовыми).

Экологичность:

- 1) Низкая ресурсоёмкость и энергозатратность на всех стадиях жизненного цикла высокоскоростной транспортной системы (проектирование, строительство, эксплуатация и демонтаж).
- 2) Высокоскоростная дорога в эстакадном исполнении не нарушает рельеф местности, биогеоценоз и биоразнообразие прилегающей территории.

3) Дорога не уничтожает плодородную почву и произрастающую на ней растительность.

4) Дорога не препятствует:

- движению грунтовых и поверхностных вод;
- перемещению людей, домашних и диких животных;
- работе сельскохозяйственной, строительной и специальной техники.

Надёжность и безопасность:

1) Наличие у высокоскоростного подвижного состава противосходной системы.

2) Десятикратный запас прочности несущих струнных элементов в транспортной эстакаде.

3) Устойчивость к гололёду, оледенению, снежным заносам, туманам, пыльным и песчаным бурям.

4) Надёжность работы при любых экстремальных температурах воздуха.

5) Высокая устойчивость к вандализму и террористическим актам.

6) Устойчивость к природными катаклизмам (при специальном исполнении и минимальном удорожании) [3]:

- землетрясения — магнитудой до 10 баллов по шкале Рихтера;
- наводнения, разливы рек — с глубиной воды до 10 м и более;
- ураганный ветер — со скоростью до 250 км/ч и более;
- цунами — с высотой волны до 20 м и более.

Потребность внедрения автоматической системы управления дорожным движением в инновационную технологию струнного транспорта «Skyway».

Как известно, АСУДД предназначены для:

- контроля интенсивности и загруженности полос ДД,
- управления светодиодным табло и распылителем реагентов,
- контроля за погодными условиями и видимостью на дорогах,
- контроля за состоянием дорожного покрытия,
- и т.д.

Исходя из описания технологии «Skyway» и проведённых исследований решено, что струнный транспорт Юницкого не требует внедрения альтернативных систем управления и координирования движения в свою технологию. Потому что данная система управления уже присутствует в технологической цепочке и позволяет в автоматическом режиме в полном объёме ее использовать.

Например, контроль над загруженностью полос прохождения в «Skyway» не нужен, т.к. поезда перемещаются по расписанию. Управление светодиодными табло для водителей отсутствует поскольку движение подвижного состава контролируется с пульта управления диспетчера транспортными потоками. Если система предусматривает наличие оператора, то вся необходимая информация поступает непосредственно на дисплей пульта управления. Такие возмущающие факторы, как контроль за погодными условиями, ограниченная видимость дорожного покрытия, уже не оказывает существенного влияния на безопасность движения, поскольку технологическая цепочка перемещения транспорта предусматривает совершенно иные методы перемещения. Исключения могут составить форс – мажорные обстоятельства. Данная технология позволяет значительно повысить безопасность перемещения подвижных объектов.

Струнные технологии SkyWay сочетает в себе свойства гибкой нити и жесткой балки, что существенно повышает износостойкость направляющих транспортного потока. В шасси подвижного транспорта встроены датчики, способные контролировать основные технологические параметры с последующей выдачей информации на монитор, или дисплей информационного табло. Данная технология позволяет экстраполировать ситуации преждевременных неисправностей, что в свою очередь повышает безопасность эксплуатации [4]

. Проведем экономическое обоснование о целесообразности использования разрабатываемого проекта относительно существующей технологии транспортной системы. Данные сведены в таблицу №1 [5]

Таблица №1. Сравнительная оценка экономической целесообразности разрабатываемого проекта .

В миллионах фунтов стерлингов

Виды работ	Ориентировочная стоимость работ, млн. фунтов												
	1 год				2 год				3 год				Итого
	I	II	III	IV	I	II	III	IV	I	II	III	IV	
Высокоскоростной междугородный грузопассажирский СТЮ (протяжённость демонстрационного участка трассы — 30 км; расчётная скорость движения — до 500 км/ч), всего	0,76	1,86	3,08	4,52	6,12	7,96	10,68	15,42	16,52	19,2	15,18	2,10	103,40
в том числе:													
1. Высокоскоростной (до 500 км/ч) междугородный грузопассажирский юнибус, всего	0,20	0,46	0,86	1,12	1,18	1,18	0,82	0,94	1,06	1,06	0,92	-	9,80
в том числе:													
1.1. проектирование высокоскоростного (до 500 км/ч) грузопассажирского юнибуса 4-го поколения (с учётом стоимости рабочих мест в Конструкторском бюро и программного обеспечения)	0,18	0,36	0,54	0,72	0,72	0,54	0,06	0,06	0,06	0,06	0,06	-	3,36
1.2. размещение заказов и единичное (индивидуальное) изготовление высокоскоростного (до 500 км/ч) грузопассажирского юнибуса (2 шт.)	-	-	0,12	0,18	0,24	0,30	0,42	0,54	0,66	0,66	0,54	-	3,66
1.3. разработка технологии, проектирование и изготовление технологического оборудования и оснастки для изготовления высокоскоростных междугородных грузопассажирских юнибусов и их эксплуатации	-	0,06	0,06	0,06	0,06	0,12	0,12	0,12	0,12	0,06	0,06	-	0,84
1.4. экспертиза документации, заключения, пуско-наладочные работы, испытания и сертификация высокоскоростного грузопассажирского юнибуса междугородного СТЮ	-	-	0,06	0,06	0,06	0,12	0,12	0,12	0,12	0,18	0,18	-	1,02

1.5. прочие расходы и непредвиденные затраты	0,02	0,04	0,08	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,08	-	0,92
2. Высокоскоростная (до 500 км/ч) рельсо-струнная транспортная эстакада, всего	0,14	0,46	0,72	1,04	1,52	2,42	4,42	9,02	10,90	14,04	10,74	0,80	56,22
в том числе:													
2.1. проектирование высокоскоростной (до 500 км/ч) рельсо-струнной транспортной эстакады 4-го поколения (предварительно напряжённой, неразрезной и статически неопределимой), с учётом стоимости рабочих мест в Конструкторском бюро и программного обеспечения	0,10	0,16	0,18	0,18	0,18	0,18	0,14	0,10	0,06	-	-	-	1,28
2.2. размещение заказов и изготовление (строительство) высокоскоростной (до 500 км/ч) рельсо-струнной транспортной эстакады (демонстрационная высокоскоростная междугородная трасса протяжённостью 30 км)	0,02	0,14	0,24	0,48	0,84	1,68	3,46	7,70	9,54	12,60	9,60	0,80	47,10

2.3. разработка технологии, проектирование и изготовление технологического оборудования и оснастки для изготовления рельсо-струнной путевой структуры и опор (промежуточных и анкерных) высокоскоростного (для 500 км/ч) междугородного грузопассажирского СТЮ	-	0,12	0,18	0,24	0,30	0,30	0,30	0,30	0,18	0,12	0,06	-	2,10
2.4. экспертиза документации, заключения, пуско-наладочные работы, испытания и сертификация рельсо-струнной путевой структуры и опор (промежуточных и анкерных) высокоскоростного (500 км/ч) междугородного грузопассажирского СТЮ	-	-	0,06	0,06	0,06	0,06	0,12	0,12	0,12	0,12	0,18	-	0,90
2.5. прочие расходы и непредвиденные затраты	0,02	0,04	0,06	0,08	0,14	0,20	0,40	0,80	1,00	1,20	0,90	-	4,84
3. Инфраструктура (пассажирские станции; стрелочные переводы; системы автоматического управления, безопасности, энергообеспечения и связи и др.), всего	0,14	0,46	0,70	1,16	1,80	2,32	2,48	2,24	1,46	1,10	0,66	-	14,52
в том числе:													
3.1. проектирование каждого концепта каждого элемента инфраструктуры для междугородного высокоскоростного (до 500 км/ч) СТЮ 4-го поколения, с учётом стоимости рабочих мест в Конструкторском бюро и программного обеспечения	0,12	0,30	0,36	0,72	1,20	1,50	1,50	1,20	0,30	-	-	-	7,20
3.2. размещение заказов и изготовление каждого концепта каждого элемента инфраструктуры междугородного высокоскоростного СТЮ (для 500 км/ч)	-	0,06	0,12	0,18	0,24	0,36	0,48	0,60	0,78	0,78	0,36	-	3,96
3.3. разработка технологии, проектирование и изготовление технологического оборудования и оснастки для изготовления каждого концепта каждого элемента инфраструктуры высокоскоростного (до 500 км/ч) междугородного грузопассажирского СТЮ	-	0,06	0,08	0,10	0,14	0,18	0,18	0,12	0,10	0,06	0,06	-	1,08
3.4. экспертиза документации, заключения, испытания, пуско-наладочные работы и сертификация каждого концепта каждого элемента инфраструктуры высокоскоростного междугородного грузопассажирского СТЮ	-	-	0,06	0,06	0,06	0,08	0,10	0,12	0,14	0,16	0,18	-	0,96
3.5. прочие расходы и непредвиденные затраты	0,02	0,04	0,08	0,10	0,16	0,20	0,22	0,20	0,14	0,10	0,06	-	1,32
4. Здания (офисные, лабораторные, производственные и др.), строительные сооружения, землеотвод, благоустройство, инженерные сети, всего	0,08	0,18	0,50	0,90	1,32	1,74	2,66	2,92	2,80	2,70	2,56	1,00	19,36
в том числе:													
4.1. проектирование каждого здания, сооружения, благоустройства, инженерных сетей), с учётом стоимости рабочих мест в Конструкторском бюро и программного обеспечения	0,06	0,14	0,30	0,50	0,60	0,40	0,30	0,14	-	-	-	-	2,44
4.2. размещение заказов на оборудование, строительные и строительно-монтажные работы	-	-	0,10	0,20	0,40	1,00	2,00	2,40	2,40	2,40	2,40	1,00	14,30
4.3. экспертиза документации, заключения, испытания, пуско-наладочные работы	-	-	-	-	0,02	0,04	0,06	0,08	0,10	0,10	0,02	-	0,42
4.4. прочие расходы и непредвиденные затраты	0,02	0,04	0,10	0,20	0,30	0,30	0,30	0,30	0,30	0,20	0,14	-	2,20
5. Маркетинг, создание клиентской базы и заказов, нормативно-правовая база, юридическое обеспечение, патентно-лицензионная работа, лицензионные платежи за использование ноу-хау и изобретений	0,20	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	0,30	3,50

Выводы и перспективы дальнейшего исследования. На основании проведенного анализа и экономической целесообразности можно сделать следующие выводы:

1. Рассматриваемая технология построения транспортных потоков обладает новизной.
2. Присутствует экономическая целесообразность в последующей разработке струнных технологий Анатолия Юницкого «Skyway».
3. Отсутствует необходимость в создании локальной системы АСУ ДД.
4. Применение данных технологий значительно повысит безопасность перемещения транспортных средств.

1. Томашевський В. М. Т-56. Моделювання систем. - К: Видавнича група ВНУ, 2005. ~ 352 С.: іл. ISBN 966-552-120-9. – ст. 52-90
2. Електронний ресурс https://skyway.capital/tehnologiya/vidi_transporta/
3. Електронний ресурс <http://skywayinvestgroup.com/aktsioneramhtml/tehnologiya-sky-way/>
4. Електронний ресурс <http://www.cbst.by/rus/activity/traffic/asu/assud/>
5. Електронний ресурс https://skyway.capital/tehnologiya/vidi_transporta/visokoskorostnoy_transport/
6. Стеценко, І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2010. – 399 с. ISBN 978-966-402-073-9. ст. 42-51.

УДК 629.113(07):004.01:004.04

Каганюк О.К., к.т.н., доц. Поліщук М.М., к.т.н.
Луцький національний технічний університет

ВИКОРИСТАННЯ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ В РОЗПІЗНАВАННІ ТРАНСПОРТНИХ ЗАСОБІВ ЗА ДОПОМОГОЮ РАДІОЛОКАЦІЙНИХ МЕТОДІВ

Каганюк О.К., Поліщук М.М. Використання інноваційних технологій в розпізнаванні транспортних засобів за допомогою радіолокаційних методів. В даній статті проводиться аналіз розпізнавання номерних знаків транспортних засобів з подальшою ідентифікацією з точки зору безпеки дорожнього руху. Необхідним, для вирішення поставленої задачі, є шумоподавляюча фільтрація з подальшим виявленням сегментації. А найбільш привабливим методом розпізнавання є кореляційний метод.

Ключові слова: номерний знак, відеоспостереження, LoG-фільтр, піксель, бінарне зображення, алгоритм, скелетизація, Fahmy.

Каганюк А.К., Полищук Н.Н. Использование инновационных технологий в распознавании транспортных средств с помощью радиолокационных методов. В данной статье проводится анализ распознавание номерных знаков транспортных средств с дальнейшей идентификацией с точки зрения безопасности дорожного движения. Необходимым в решении поставленной задачи есть шумоподавляющая фильтрация сигнала с последующей сегментацией. А наиболее приемлемым методом распознавания, является корреляционный метод.

Ключевые слова: номерной знак, видеонаблюдение, LoG-фильтр, пиксель, двоичное изображение, алгоритм, скелетизация, Fahmy.

Kaganyuk A.K., Polischuk N.N. The Use of innovative technologies in recognition of transport vehicles by radio-location methods. In this article an analysis is conducted recognition of registration numbers of transport vehicles with further authentication from point of safety of travelling motion. By a necessity in the decision of the put task est' shumopodavlyayuschaya filtration of signal with subsequent segmentation. And by the most acceptable method of recognition, there is a cross-correlation method.

Keywords: license plate, video surveillance, LoG, filter, pixel, binary image algorithm, skeletize, Fahmy.

Постановка наукової проблеми. Місцезнаходження транспортних засобів, цінних вантажів, рухомих об'єктів, розпізнавання номерних знаків набуває все більшої значення в повсякденному житті і є. вкрай **актуальні** при створенні сучасної інфраструктури на сьогоднішній день. Вони дозволяють управляти маршрутами автотransпортних засобів, забезпечувати безпеку автомашин і здійснювати їх пошук у разі викрадення, або інших життєвих ситуацій. Такі системи дозволяють в автоматичному режимі здійснювати пропуск на режимний об'єкт, автоматизувати процес видачі пропусків з подальшим відстеженням в умовах здійснення перевезень.

Сьогодні актуальним є безпека дорожнього руху транспортних засобів та ідентифікація автомобілів за їх реєстраційним номерним знаком. Розробка та створення систем контролю та розпізнавання номерних знаків мають все більше поширення у різних сферах народного господарства.[1].

Актуальність задачі в розпізнаванні номерних знаків транспортних засобів полягає в тому, що з кожним днем на шляхах стає більше автомобілів, що в свою чергу створює проблеми у великих містах, особливо в «часи пік», тому виникає потреба в створенні автоматизованих систем контролю та відстеження як транспортних засобів, так і номерних знаків рухомого транспорту.

Аналіз досліджень. У даний час існує досить велика кількість систем розпізнавання номерних знаків, але не всі з них є якісною та надійною продукцією. Системи з високою швидкістю і точністю розпізнавання є комерційними, засекреченими і дорогими, що не дозволяє здійснити їх масове впровадження [1,13]. Тому, наша задача полягає в тому, щоб визначити найбільш ефективні системи. Провести аналіз функціонування тих систем виявити основні методи ідентифікації, створити раціональний алгоритм і націлити зусилля на створення і вдосконалення розробляє свої системи.

Важливою проблемою в контролі номерних знаків є визначення найбільш ефективних методів обробки зображень та розпізнавання автомобільних номерних знаків, а також створення нових комбінованих або модифікованих методів для якісного розпізнавання системи в реальному часі.

Для досягнення поставленої *мети* необхідно вирішити наступні задачі:

1. Аналіз методів і здійснення обробки зображень при локалізації номерного знака.
2. Аналіз методів сегментації.
3. Дослідження методів розпізнавання образів.
4. Розробка і дослідження алгоритмів розпізнавання символів.

5. Вибір методів розпізнавання і оцінка їх ефективності.

6. Розробка системи для обробки зображення з камер відеоспостереження для розпізнавання номерного знака транспортного засобу.

В даній статті не буде можливим розглянути всі вказані методи і поставлені задачі, тому ми зупинимось на перших двох методах, це аналіз методів по обробці зображень і аналіз методів сегментації

Задача розпізнавання номерних знаків розбивається на три етапи: попередня обробка зображення, сегментація, розпізнавання символів.

Попередня обробка зображення необхідна для поліпшення візуальної якості зображення і, в кінцевому підсумку, для локалізації номерний пластини.

Сегментація проводиться з метою виділення символів для подальшого розпізнавання обраним методом.

У зв'язку зі складністю поставленої задачі що до розпізнавання автомобільних номерів в її загальній постановці, рішенням як всієї задачі в цілому, так і окремих її аспектів присвячено велику кількість робіт.

Виклад основного матеріалу. У світі вже існують розробки, які стосуються розпізнавання автомобільних номерних знаків. Розглянемо деякі з них.

Система «Авто-Інспектор» – програмно-апаратний комплекс, що забезпечує розпізнавання номерів рухомих автомобілів, надійно працює в широкому діапазоні зовнішніх умов, легко інтегрований з охоронним обладнанням, виконавчими пристроями і зовнішніми базами даних.

Ефективний для вирішення задач реєстрації, ідентифікації та забезпечення безпеки автомобілів, контролю транспортних потоків.

У системі «Авто-Інспектор» реалізовані функціональні можливості необхідні для ефективного вирішення завдань на різних об'єктах: від забезпечення збереження автомобілів в межах автостоянки до контролю за пересуванням транспорту в масштабах підприємства, окремої магістралі, цілого міста.

Основні функціональні можливості даної системи:

- розпізнавання реєстраційного номера автомобіля;
- виявлення появи автомобіля в кадрі;
- можливість адаптації до стандартів номерів будь-якої країни світу;
- успішна робота модуля в будь-яких погодних умовах;
- вибір кадру з оптимальним розміром і чіткістю реєстраційного номера;
- одночасне розпізнавання в зоні контролю декількох номерів (переклад відеозображення в текстовий формат);
- ефективна взаємодія з різними охоронними системами (охоронного телебачення, контролю доступу);
- запис кожного проїзду, в'їзду/виїзду транспортних засобів;
- зіставлення розпізнаного номера з інформацією бази даних (власної або зовнішньої) [2].

АПК «Автоураган» – апаратно-програмний комплекс автоматичного розпізнавання зображень державних реєстраційних знаків транспортних засобів.

Відеозображенням, що надходить на комп'ютер, дає можливість проводити розпізнавання реєстраційних знаків автомобілів; перевірку їх за підключеними базами даних з видачею повідомлення оператору та збереженням інформації про проїзд або з виконанням іншої призначеної дії.

Гнучкі налаштування ПО дозволяють використовувати АПК «Автоураган» для вирішення численних завдань пов'язаних з розпізнаванням автомобільних номерів.

Для ефективного використання систем по ідентифікації номерних знаків, необхідно використовувати модульний принцип, який може дозволити провести уніфікацію в будові таких систем.

Які переваги дає модульний принцип будови таких систем? Модульний принцип дозволяє утворювати як розподілену мережеву клієнт – серверну систему, так і одночасно локальну систему, де використовується обмежена кількість комп'ютерів, це по – перше, по – друге, модульний принцип дозволяє удосконалювати дану систему так і нарощувати, доповнюючи її іншими модулями з конкретними функціями для конкретного підприємства.[3].

У даній області ведуться також дослідження, наприклад, Fahmy [4], який запропонував метод двобічної асоціативної пам'яті нейронної мережі для читання номерних знаків. Цей метод підходить

для невеликої кількості моделей. Інші компанії, такі як Nijhuis і Ter Brugge et al [5] пропонували використання нечіткої логіки і нейронних мереж для номерних знаків транспортних засобів. Даний метод дозволяє використовувати нечітку логіку для сегментації і дискретно-часові клітинні нейронні мережі для виділення ознак. Lotufo, Morgan і Johnson [6] запропонували автоматичну систему розпізнавання номерного знака за допомогою оптичних методів. Такі компанії як розпізнавання символів. S.K. Kim, D.W. Kim і H.J. Kim [7] запропонували у розпізнаванні символів використовувати генетичний алгоритм сегментації для локалізації області з номерним знаком. Hontani [8] запропонував спосіб видалення символів без знання їх положення і розмірів зображення.

На національному рівні розробляються системи розпізнавання автомобільних номерних знаків компанією «ЕФ ЕФ», яка є провідним виробником на українському ринку. В даний час компанія займається випуском систем безпеки і відеоспостереження. Вони виробляють обладнання та програмно-апаратні комплекси, що покривають широкий спектр завдань в області безпеки, та керування технологічними процесами.

Система «НомерОК» проводить захоплення відеопотоку, розпізнавання автомобільних номерів в потоці, збереження події з записом номера, часу і кадру з номером. У програмі передбачена можливість внесення розпізнаних номерів у «білий» і «чорний» списки і додавання коментарів до розпізнаного номера [9].

При локалізації номерного знака є зображення з камери, які найчастіше зашумлені, розмиті. Для усунення адитивного та імпульсного шуму проводиться попередня обробка такого сигналу який згладжується фільтрами (гаусовський і медіанний фільтри). При фільтрації яскравість (сигнал) кожної точки вихідного зображення, викривленого перешкодою, замінюється деяким іншим значенням яскравості [10].

В обробці зображень використовується двовимірний дискретний функція Гауса з нульовим середнім:

$$G[i, j] = e^{-(i^2 + j^2)/2\sigma^2} \quad (1)$$

Фільтр, побудований на її основі, служить для згладжування. При цьому виконуються маніпуляції з параметром σ^2 (встановлене значення дисперсії зазвичай набагато більше обчисленого, що призводить до більш ефективного шумозаглушення, але й збільшує ступінь розмивання зображення).

При медіанній фільтрації використовується двовимірне вікно (маска фільтра), що має центральну симетрію. Центр вікна розташовується в поточній точці фільтрації. Форма вікна може бути різною. Розміри апертури оптимізуються в процесі аналізу обробки і залежать від детальності зображення. Відліки, що опинилися в межах вікна, називаються робочою вибіркою поточного кроку.

Значення елементів робочої вибірки упорядковуються за зростанням. Вибирається елемент, який займає центральне положення в цій послідовності – медіана. Якщо центральне значення є шумовим викидом, то фільтр забезпечить його придушення.

Для виконання подальших етапів необхідно виконати виділення на зображенні меж об'єктів – безперервних кривих, в яких спостерігається різкий стрибок яскравості.

Виділення меж здійснюється за допомогою градієнтних фільтрів першого і другого порядку. Використовуючи, наприклад, фільтр другого порядку – LoG-фільтр компанії (Marr-Hildreth), працює шляхом зв'язування зображення з лапласіаном функції Гауса. Він поєднує в собі виявлення меж зі згладжуванням.

Маска фільтра G буде визначатись із слідуючого масива.

$$G := \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 5 & -8 & 5 & 4 \\ 6 & -8 & -52 & -8 & 6 \\ 4 & 5 & -8 & 5 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Результат роботи фільтру LoG представлений на рис. 1.



Рис. 1. Зображення номерного знака автомобіля: а) вихідне зображення, б) з виділеними межами

Відповідно з алгоритмом сегментації, проводиться пошук координат передбачуваних символів на локалізованій зоні. На першому етапі сегментації обчислюються оцінки правдоподібності приналежності пікселів лініям символів. У найпростішому випадку використовуються або готові результати накладення фільтра Н, або безпосередньо яскравість пікселів. Результатом першого етапу є масив оцінок правдоподібності EZh_{xw} .

Другий етап – обчислення вектора середньої оцінки правдоподібності за стовпцями:

$$A_j = \frac{1}{h} \sum_{i=1}^h EZh_{ij} \quad (2)$$

Вектор А дозволяє виявити проміжки фону між символами, які проявляються у вигляді екстремумів, якщо розглядати А як функцію.

На третьому етапі проводиться виявлення вертикальних роздільників – меж між символами. Для цього кожному елементу зіставляється оцінка правдоподібності N_i . Чим вище ця оцінка, тим вища ймовірність того, що в даному місці знаходиться проміжок фону між символами. Індеси i обраних N_i – це координати вертикальних роздільників (межі) між символами:

$Dev = \{dev1, dev2, \dots, devk\}$, где k – кількість знайдених роздільників.

На четвертому етапі здійснюється уточнення вертикальних і горизонтальних меж символів. Для цього методом Отсу обчислюється порогове значення яскравості для фрагмента зображення між двома роздільниками. Поріг дозволяє бінаризованими фрагмент і в результаті знайти координати виділеного на фрагменті об'єкта [11, 14, 15].

Для спрощення задачі розпізнавання символів номерного знака застосовується морфологічна ерозія. Мета такої обробки – отримання скелета символу (зображення шириною в 1 піксель).

Ерозія бінарного зображення А структуруючим елементом В позначається $A \ominus B$ та задається виразом:

$$A \ominus B = \{z \in A | B_{\approx} \subseteq A\} \quad (3)$$

При виконанні операції ерозії структурний елемент проходить по всім пікселям зображення. Якщо в деякій позиції кожен одиничний піксель структурного елементу співпадає з одиничним пікселем бінарного зображення, то виконується логічне додавання центрального пікселя елемента з відповідним пікселем вихідного зображення (рис. 2).

У результаті застосування операції ерозії всі об'єкти, менші, ніж структурний елемент, стираються, а об'єкти, з'єднані тонкими лініями стають роз'єднаними, та розміри всіх об'єктів зменшуються [12, 14, 15]. Якщо структурний елемент являє собою єдиний піксель, то в результаті виходить якісний острів об'єкта.

Після попередньої обробки та сегментації можна приступати до вирішення задачі розпізнавання.

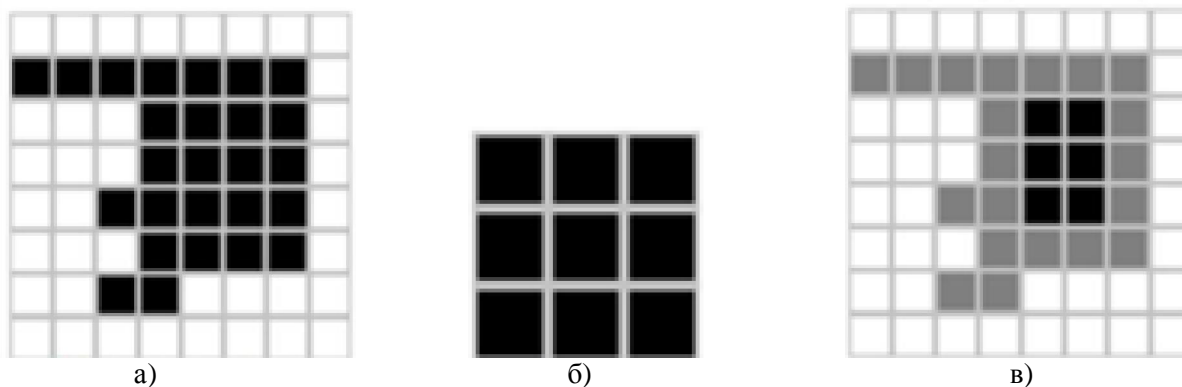


Рис. 2. Логічне додавання центрального пікселя елемента з відповідним пікселем вихідного зображення: а) бінарне зображення А, б) структурний елемент В, в) витончення зображення А структурним елементом В²

Висновки та перспективи подальшого дослідження. На підставі аналізу літературних джерел виділено основні етапи реалізації та алгоритми, які можуть бути використані при проектуванні заданої системи. Необхідними для вирішення цього завдання є шумоподавляюча фільтрація, виявлення меж, скелетизація та сегментація, а найбільш придатними методами розпізнавання – кореляційні, шаблонні методи і нейронні мережі.

1. Личканенко И.С. Методы обработки изображений и распознавания образов для задачи обнаружения номерных знаков транспортных средств // Информатика и компьютерные технологии – 2013. – Донецк: Донецкий национальный технический университет, 2013
2. Авто-Инспектор – система распознавания автомобильных номеров [Электронный ресурс]. – Режим доступа: <http://www.iss.ru/products/intelligent/auto>.
3. АПК «АВТОУРАГАН» [Электронный ресурс]. – Режим доступа: <http://www.recognize.ru/node/23>
4. Fahmy M.M.M., 1994, Automatic Number-plate Recognition : Neural Network Approach, Proceedings of VNIS'94 Vehicle Navigation and Information System Conference, 3 1 Aug-2 Sept, pp.291-296, 1994
5. Nijhuis J.A.G., Brugge Ter M.H., Helmholt K.A., Pluim J.P.W., Spaanenburg L., Venema L., Westenberg M.A., 1995, Car License Plate Recognition with Neural Networks and Fuzzy Logic, IEEE International Conference on Neural Networks, pp.2232-2236, 1995
6. Lotufo R.A., Morgan A.D., and Johnson AS., 1990, Automatic Number-Plate Recognition, Proceedings of the IEEE Colloquium on Image analysis for Transport Applications, V01.035, pp.6/1-6/6, February 16, 1990.
7. Kim S.K., Kim D.W., and Kim H.J., 1996, A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation, Proceedings of 3rd IEEE International Conference on Image Processing, V01.2., pp.661-664, 1996.
8. Hontani H., and Koga T., (2001), Character extraction method without prior knowledge on size and information, Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'01), pp. 67-72.
9. Система распознавания автомобильных номеров «НомерОк» [Электронный ресурс]. – Режим доступа: <http://avtonomerok.com>.
10. Грузман И.С., Киричук В.С. Цифровая обработка изображений в информационных системах: Учебное пособие. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.
11. Воскресенский Е.М., Царев В.А. Моделирование и адаптация систем распознавания текстовых меток на видеоизображениях. – Череповец: ИНЖЭКОН-Череповец. – 2009. – 154 с.
12. Математическая морфология [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/113626/>
13. Личканенко І.С. Дослідження методів та пошук ефективного алгоритму для задачі розпізнавання номерних знаків транспортних засобів електронне посилання <http://masters.donntu.org/2013/fknt/lichkanenko/diss/indexu.htm>
14. Фурман Я.А., Кривецкий А.К., Передреєв А.К. Введение в контурный анализ; приложение к обработке изображений и сигналов. – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2001. – 592 с.
15. Методы компьютерной обработки изображений / Ред. Со́йфеф В.А. – М.: ФИЗМАТЛИТ, 2001. – 784 с.

УДК 629.113(07):004.01:004.04

Каганюк О.К., к.т.н., Сопіжук Р.В.

Луцький національний технічний університет

РОЗРОБКА АПАРАТНО - ПРОГРАМНОГО ВИМІРЮВАЛЬНОГО КОМПЛЕКСУ ДЛЯ ДІАГНОСТИКИ ТА ДОСЛІДЖЕННЯ ТЕХНІЧНОГО СТАНУ ДВИГУНІВ ВНУТРІШНЬОГО ЗГОРЯННЯ ЗА ДОПОМОГОЮ МІКРОКОНТРОЛЕРА ELM327

Каганюк О.К., Сопіжук Р.В. Розробка апаратно-програмного вимірювального комплексу для діагностики та дослідження технічного стану двигунів внутрішнього згорання за допомогою мікроконтролера ELM327. В даній статті розглядається питання розширення можливостей діагностики технічного стану двигунів внутрішнього згорання шляхом автоматизації вимірювання тиску в циліндрах двигуна внутрішнього згорання. Задача вирішується з використанням датчиків тиску, мікроконтролеру та персональної обчислювальної машини.

Ключові слова: Датчик тиску, двигун внутрішнього згорання, мікроконтролер elm 327, обчислювальна машина.

Каганюк А.К., Сопіжук Р.В. Разработка и исследование аппаратно – программного измерительного комплекса для диагностики и исследования технического состояния двигателей внутреннего сгорания с помощью микроконтроллера ELM327. В данной статье рассматриваются вопросы расширения функциональных возможностей диагностики технического состояния двигателей внутреннего сгорания путем автоматизации измерения давления в цилиндрах двигателя внутреннего сгорания. Поставленная задача разрешается путем использования датчиков давления, микроконтроллера и персональной вычислительной машины.

Ключевые слова: Датчик давления, двигатель внутреннего сгорания, микроконтроллер elm 327, вычислительная машина.

Kaganyuk A.K., Spiguk R.V. Development and research of hardware - software measuring complex for the diagnosis and study the technical combustion engines by means of microcontroller ELM32. This article discusses the issues of expanding the functionality of diagnosing the technical condition of the engines internal zgoraniya by automating the measurement of pressure in the cylinders of an internal zgoraniya. Postavlenaya problem is resolved by the use of pressure sensors, a microcontroller and a personal computer.

Keywords: pressure sensor, microcontroller elm 327 computer.

Забезпечення якості машин неможливе без широкого використання на всіх стадіях їхнього життєвого циклу технічних засобів діагностування. Широка номенклатура машинобудівної продукції, велике число вимірюваних діагностичних параметрів, роблять особливо актуальним реалізацію різних алгоритмів та пристроїв діагностування.

Актуальним питанням стає технічна діагностика автомобільного транспорту яка являє собою своєрідним індикатором і гарантом якості та надійності нової техніки, тому її застосування у світі з кожним роком зростає.

У процесі експлуатації автомобілів поступово знижується ефективна потужність і збільшується питома витрата палива і виникає **проблема ефективної діагностики технічного стану двигуна.**

Сучасні інформаційні технології, завдяки спрощеній системі програмування мікроконтролерів, суттєво спростили вирішення задач з вимірювання фізичних величин, надавши можливість їх застосування широким колом спеціалістів різних галузей. Це дозволяє розширити коло задач для мікроконтролерів. Однією із таких задач є застосування мікроконтролерів для діагностики двигунів внутрішнього згорання (ДВЗ).

Важливим і відповідальним вузлом ДВЗ є циліндро-поршньова група (ЦПГ). Втрата компресії, відхилення її від норми в різних циліндрах приводить до швидкого зносу ЦПГ. Для діагностики стану ЦПГ застосовуються пристрої вимірювання компресії (компресометри). Однак компресометр не дає достатньої інформації для детальної діагностики стану ЦПГ двигуна. Так як зниження компресії може бути викликано не тільки зношенням гільз циліндрів, поршнів, компресійних кілець, але й іншими різнобічними причинами, основними з яких є: порушення теплових зазорів в клапанному механізмі, зношення направляючих втулок клапанів, прогар клапана чи поршня, негерметичність впускних (випускних) клапанів, дефекти прокладки головки блоку циліндрів, закоксовування поршневих кілець та інше. Для отримання повної діагностичної інформації ЦПГ необхідно мати залежності зміни тиску в камері згорання від кута повороту колінчатого валу.

Аналізуючи характер залежності можна провести детальне діагностування ДВЗ.

В літературі викладено питання щодо проведення тесту ДВЗ на падіння тиску в камері згорання шляхом локалізації місця із слабкою герметизацією і за цим по таблиці визначення

дефектної деталі. Аналіз зміни тиску на протязі робочого циклу ДВЗ не здійснюється, результат вимірювання не реєструється і не може бути використаний для прогнозування технічного стану ДВЗ, планування його капітального ремонту.

В літературі представлено програмно-технічний комплекс на базі багатоканального пристрою вводу-виводу та обробки аналогової і цифрової інформації – багатофункціональної плати збору даних L-783 виробництва ЗАО "Л-Кард" (Росія). Описаний вимірювальний комплекс є універсальним та використовується сумісно із програмним продуктом PowerGRAPH як для проведення науково-дослідних робіт в напрямку удосконалення робочого процесу ДВЗ, так і для проведення циклу лабораторних занять, з метою придбання студентами практичного досвіду та закріплення теоретичного матеріалу. Проте для вирішення завдання технічної діагностики ЦПП потрібно удосконалення його програмного забезпечення, доступ до якого закритий.

Мета даної роботи є поліпшення функціонально-якісних показників діагностичного обладнання двигунів внутрішнього згоряння за рахунок розробки апаратно-вимірювального комплексу контролю встановлення тиску у камері спалення.

Викладення основного матеріалу. Стан сучасної елементної бази і обчислювальної техніки визначає, що найбільш раціональною побудовою вимірювальних систем у складі лабораторних установок є апаратно-програмні вимірювальні комплекси (АПВК), або автоматизовані системи перевірки тиску

у камері спалення які задовольняють наступним умовам:

- використовують стандартну елементну базу;
- використовують стандартні інтерфейси і протоколи;
- мають зручний, наочний і "інтуїтивно наочний" графічний інтерфейс користувача;
- автоматично передають і обробляють інформацію.

Для задоволення перерахованих вище вимог АПВК доцільно будувати на основі персональної обчислювальної машини, до якої підключаються вимірювальні пристрої через стандартні роз'єми.

Виходячи з того, що процес наповнення камери спалення, в залежності від розміру припустимої площі поперечного перерізу, у першому періоді триває 0,4 від швидкості обертання колінчатого валу. Тому, для вимірювання динамічних характеристик достатня швидкодія датчика тиску становить не більше 0,1 с, а швидкодія вимірювального пристрою – на порядок вище - 0.01 с [1,2].

Для забезпечення одночасного вимірювання параметрів у кожній камері спалення, необхідно розміщувати не менше 4-х, а якщо буде потреба збільшення числа вимірів, до 8 датчиків з динамічним діапазоном не менш 50 дБ і часовим інтервалом не більше 0,5 мс (виходячи із максимальної швидкості наповнення камери спалення). Для виконання цієї умови необхідно мати 8-ми каналний аналого-цифровий перетворювач (АЦП) не менш 9 розрядів (виходячи із похибки вимірювання тиску не більше 10% для інженерного розрахунку) з тактовою частотою більше 2 кГц. При цьому швидкість передачі даних повинна бути не менше 126 кБит/с [3]. Таку швидкість передачі даних у ПЕОМ можна забезпечити по інтерфейсу USB. Обробку і відображення інформації можна здійснити за допомогою ПЕОМ. Структура варіанту АПВК показана на рисунку 1.

До складу АПВК необхідно включити:

- не менше чотирьох датчиків тиску, підключених до кожної камери спалення (рис.1.) з буферним підсилювачем швидкодією не більше 0,1 с.
- мікроконтролер, що включає в себе: комутатор, 8-канальний аналого-цифровий перетворювач (АЦП) із цифровим сигнальним процесором (ЦСП) і комплектом кабелів сполучення;
- автоматизоване робоче місце (АРМ), що включає ПЕОМ з програмним забезпеченням.

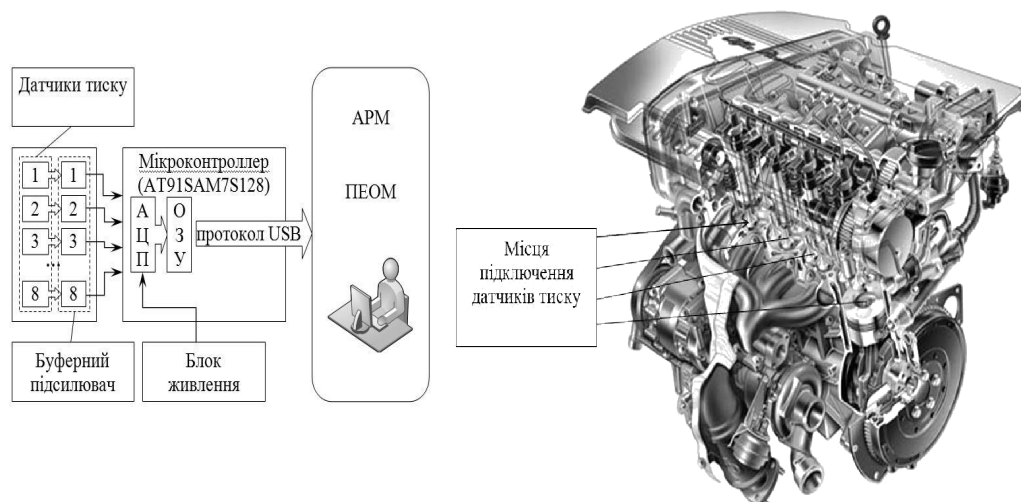


Рис. 1 – Схема апаратно-програмного вимірювального комплексу та схема розміщення датчиків тиску на двигуні внутрішнього згоряння

Для отримання кривої зміни тиску в циліндрі необхідно прогріти двигун до робочої температури, встановити в досліджуваний циліндр датчик тиску замість вивернутою свічки, а високовольтний провід цієї свічки встановити на розрядник. У випадку, коли двигун оснащений єдиним модулем запалювання на всі циліндри, можна зняти модуль і встановити додаткові високовольтні дроти між його виводами і свічками, дотримуючись при цьому запобіжних заходів. Якщо можливо, відключити роз'єм від форсунки циліндра, що діагностується, щоб виключити подачу палива. Синхронізацію при знятті кривої зміни тиску краще використовувати зовнішню, від датчика першого циліндра. Запустити двигун і зняти криву зміни тиску.

Програма виконує наступні завдання: відображає у вигляді часових графіків тиск у кожній камері спалення одночасно, вимірює часові та амплітудні параметри, запам'ятовувати отримані результати у файл, керує роботою мікроконтролера AT91SAM7S128 [4, 5].

Спочатку оператор встановлює параметри АЦП. Зовнішній вигляд діалогового вікна на Рис.2. Оператор встановлює навпроти позиції "Коеф.дел" встановлює коефіцієнт ділення, може встановити інший тактовий генератор, включити або виключити відповідні канали АЦП, встановити розрядність АЦП 10 розрядів або 8 разрядів. Після встановлення оператор повинен натиснути кнопку "Применить", після чого з'являється значення частоти АЦП напроти строчки "Частота АЦП". Для встановлення коефіцієнту множення амплітуди (рис. 2.) спочатку необхідно провести калібрування. Для калібрування у графі

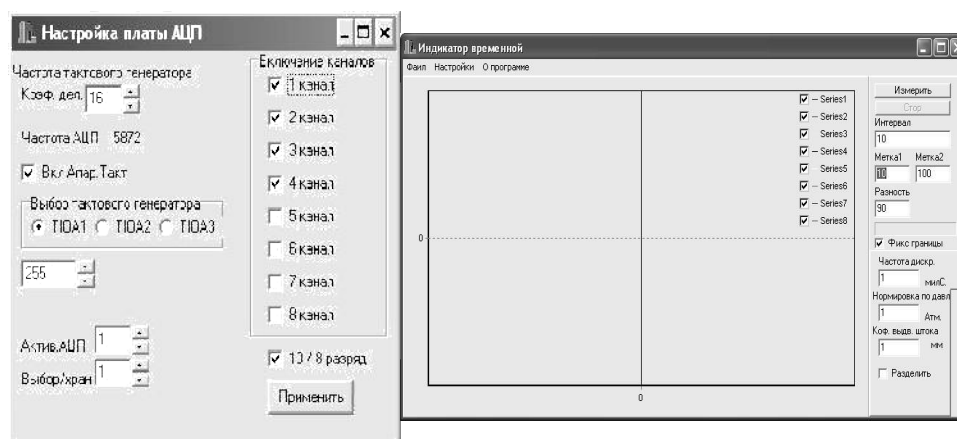


Рис. 2 – Зовнішній вигляд діалогового вікна налаштування параметрів АЦП для блоку перетворень та зовнішній вигляд основного інтерфейсу програми

“Нормировка по давлению” встановлюється одиниця. Після виміру отримуємо на екрані криву встановленого тиску. Максимальне значення у байтах та значення по манометру запам'ятовуються а потім знаходиться їх співвідношення. Це значення потім записується у графу “Нормировка по давлению”. В подальших вимірюваннях значення амплітуди буде відповідати значенням у атмосферах або у МПа.

Процес з'єднання з зовнішнім блоком перетворення можливо проконтролювати у вікні “Интерфейс работы с АЦП”. Зовнішній вигляд інтерфейсу програми зображено на рис. 3. Після натискання кнопки “Пуск” у нижньої частині вікна з'являється посилання програми про підключення або відключення пристрою.

Початок перетворення починається з натискання кнопки “Измерить” рис.3. Кількість вимірювань встановлюється у графі “Интервал”. Після накопичення заданого інтервалу з'являється графік встановлення тиску, зовнішній вигляд якого зображено на рис. 3.

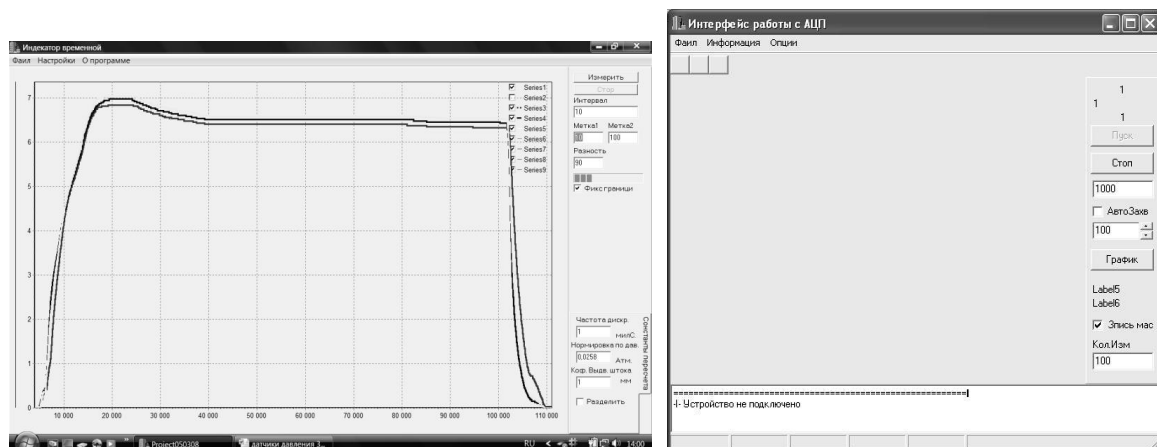


Рис 3 – Зовнішній вигляд процесу встановлення тиску та зовнішній вигляд інтерфейсу роботи з зовнішнім блоком перетворення

Після отримання кожного вимірювання оператор запам'ятовує отриманий графік натисканням “Файл” → “Сохранить rpn”. Це дає змогу у подальшому використовувати отримані результати вимірювань для проведення досліджень.

Отримання значень на графіку здійснюється за допомогою вертикальних лінії які знаходяться зліва на початку графіку. За допомогою курсору ці лінії встановлюються у заданих місцях оператором. Під підписами “Метка 1” та “Метка2” з'являються значення положення вертикальних ліній. У графі “Разность” автоматично розраховується відстань між двома вертикальними лініями. Таким чином здійснюється вимірювання часових параметрів перехідного процесу у камері спалення. На осі отриманого графіку відображається значення тиску. Графік встановлення тиску у кожній камері спалення має свій колір, та підписується у верхньому правому куті словами “Series1”. Напроти кожної назви графіку “Series...” є галочка, яка дозволяє включити або виключити відображення відповідного графіку. Таким чином можливо окремо розглядати кожний графік встановлення тиску у кожній камері спалення.

Діагностичний адаптер OBD-II ELM327

За допомогою ELM327 ви самі зможете провести діагностику двигуна вашого автомобіля! Тепер не потрібно втрачати час для поїздки на СТО, не треба витратити гроші кожен раз, коли у вас загорілася лампочка перевірте двигун або несправний який-небудь датчик! У 80% випадках проблема несуттєва і вирішувана в домашніх умовах. За допомогою цього сканера ви самі дізнаєтеся причину неполадки і зможете видалити цю помилку! Враховуючи вартість послуг з комп'ютерної діагностики на СТО, даний прилад окупається за кілька разів!

Підходить до всіх автомобілям із США з 1996 року (не має значення



виробник). І до всіх автомобілів з європейського ринку бензин з 2001 року, дизель з 2004 року; Але часто може підійти і до автомобілів з Європи старше 2001! Наприклад багато моделей Volvo, SAAB, Jaguar, Porsche, Lancia, Alfa Romeo, Fiat, Opel з Європи підтримують цей авто сканер з 1996 року. Можливості діагностичного адаптера ELM 327 не залежить від виду підключення (COM, USB або Bluetooth).

Цим він і заманює - один адаптер і багато машин. Та ще й програму можна використовувати одну. Зустрічаються винятки, звісно, коли адаптер не може прочитати машину. Найчастіше це трапляється через те, що машина не підтримує OBD-II. Але це все-таки виняток.

Можливості даного діагностичного адаптера:

- Читає помилки
- Стирає помилки
- Показує витрата палива
- Виводить параметри систем і датчиків автомобіля в реальному часі:
- (приклади)
- Обороти двигуна
- Навантаження двигуна
- Температура охолоджуючої рідини
- Стан паливної системи
- Швидкість руху автомобіля
- Короткостроковий витрата палива
- Довгостроковий витрата палива
- Абсолютний тиск повітря
- Випередження запалювання
- Температура всмоктуваного повітря
- Масова витрата повітря
- Положення дросельної заслінки
- Лямбда-зонд
- Тиск палива тощо.
- Будує графіки
- Виводить інформацію про автомобіль
- Замірявся час розгону і потужність
- Лічильник часу розгону: від 0 до 100 км / год
- Робить стоп кадр
- Записує дані
- Експорт даних і роздруківка

Софта в мережі досить багато, платні, безкоштовні додатки, тріольні, ламані. Але по істині заслуговують обговорення всього одна, дві програмки. Я напевно не відзначився оригінальністю, якщо напишу, що сама «багата» програма це Torque Pro, благо у нас є умільці які зламали цю програмку і вона може нам дістатися безкоштовно, завантажити її і почитати про неї можна на 4pda.ru в темі Torque Pro (OBD2 / автомобіль), так само можна було б розглянути ще пару програм таких як DashCommand, HobDrive і OBD DROIDSCAN PRO, але робити цього не буду, так як невеликий розповідь про Torque буде достатній.

Прив'язка пристрою проходить за лічені секунди, тільки не забуваємо в настройках Bluetooth виставити «Видимість для всіх». І ось після закінчення пари секунд вводимо той самий пароль який у багатьох не проходить, але у мене все нормально, паролем служить 4-х значне число тисячу двісті тридцять чотири. При закусці Torque, прогамма попросить дозвіл на підключення GPS для вимірів швидкості і місцезнаходження. Більш детальну інструкцію можна погортати на сайті elm327.uscoz.ru. Ну а я спробую трохи розповісти про функціонал і корисності програми.



Рис 4 – Головне меню програми та доступні датчики

Зверху є 4 іконки, вони сигналізують про: підключенні до GPS, телефон говорить нам незрозуміло про що, індикатор адаптера говорить про з'єднання телефону з адаптером, машинка позначає, що адаптер отримує інформацію від авто.

Ну звичайно ж відразу хочеться подивитися, що я зможу зробити і які датчики працюють. Йдемо в налаштування і дивимось інформацію про адаптері.

Далі переходимо до бортового комп'ютера з показанням різної інформації. На швидкості 120км / год.(Рис. 5)

Гріх було б не про сканувати авто на наявність помилок. Дана процедура займає не так багато часу. (Рис. 6)



Рис 5 – Покази різної інформації на швидкості 120км / год. та процедура сканування помилок

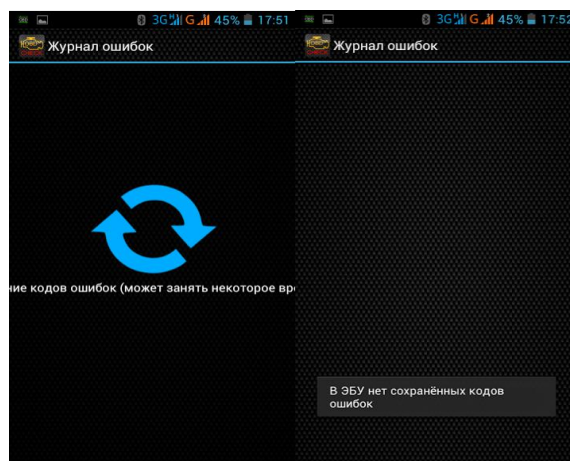


Рис 6 – Процедура сканування помилок

На робочий екран можна додати різні екрани. Перелік нижче на фото.

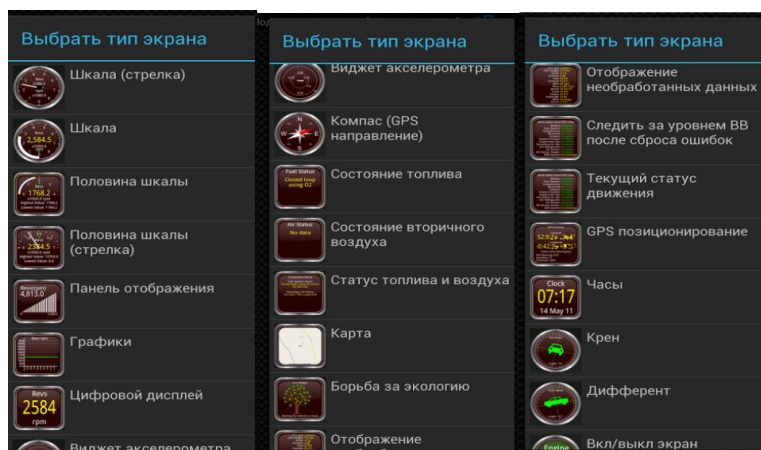


Рис 7 – Перелік типів екранів

Також програма може, записати трек вашої подорожі з накладенням на карту. Відображати миттєву витрату, вартість поїздки, напруга, температуру повітря тестую чого двигуна. На цьому напевно я закінчу зі скріншотами програми, так як огляд все таки адаптера а не софта.

Мені здається хороша річ для автомобіліста, за невелику ціну ми отримуємо дуже великий доступ до інформації про машину, це куди краще ніж платити за офіційні сканери наприклад для зняття помилки. Думаю така річ повинна бути у будь-якого поважаючого себе автолюбителя.

Висновки. Таким чином запропонований апаратно-програмний вимірювальний комплекс дозволяє:

1. спостерігати встановлення тиску у камері спалення;
2. , підвищити точність вимірювання часових параметрів встановлення тиску та його значення;
3. проводити дослідження і порівняльний аналіз отриманих графіків між камерами спалення;
4. надає можливість з більшою достовірністю діагностувати несправності циліндро-поршньової групи;
5. зменшити затрати на позапланові капітальні ремонти двигунів внутрішнього згоряння.

1. Губертус Гюнтер. Диагностика дизельных двигателей. Серия «Автомеханик». Пер. с нем. Ю. Г. Грудского. – М: ЗАО «КЖИ «За рулем», 2004 г.
2. Врублевский А.Н. Техническое решение для исследования рабочего процесса двигателя внутреннего сгорания / Врублевский А.Н., Дзюбенко А.А., Вахрушев В.И. // Промышленные измерения, контроль, автоматизация, диагностика. – 2008. – №4. – С 30-33.
3. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel. - М.: ИП РадиоСофт, 2002 - 176 с.
4. Шпак Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров. / - К.: «МК-Пресс», М: Издательский дом «Додэка XXI» 2007. - 400 с.

УДК 519.866:658.149.3

Рудь Н.Т.

Луцький національний технічний університет.

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ТРАНСФЕРУ ТЕХНОЛОГІЙ

Рудь Н.Т. Математичне моделювання процесів трансферу технологій. В статті запропонована модель трансферу технологій в регіоні у вигляді орієнтованого графа. Процес трансферу технологій описаний системою рівнянь балансу грошових потоків для суб'єктів інноваційної діяльності. Розв'язок системи рівнянь дозволяє оцінити динамічний індекс, який характеризує ефективність трансферу технологій.

Ключові слова: наука, виробництво, інноваційна інфраструктура, трансфер технологій, модель.

Рудь Н.Т. Математичне моделирование процессов трансфера технологий. В статье предложена модель трансфера технологий в регионе в виде ориентированного графа. Процесс трансфера технологий описан системой уравнений баланса денежных потоков для субъектов инновационной деятельности. Решение системы уравнений позволяет оценить динамический индекс, характеризующий эффективность трансфера технологий.

Ключевые слова: наука, производство, инновационная инфраструктура, трансфер технологий, модель.

Rud N. Mathematical modeling technology transfer processes. Innovation transformation of the region involves the development of their own approach to the transfer of innovative technologies, which should take into account the conditions, opportunities, prospects and reserves, i.e. the innovation potential of the region. It is necessary to provide for a close relationship and possibility of transfer to the programs and projects results implementation, basic research to scientific applications, and achievements of the past - to the scientific, technical and innovative programs and further - to the economic, environmental and social programs. Mediation in science and innovation sphere on the information transfer between the stages of the innovation process is an activity to ensure technology transfer. This article describes a model of transfer of technologies in a region, which is represented as oriented graph. The process of transfer of technologies is given by a system of balance equations of monetary transfers for the participants of the innovative activity. The solution of the system of equations helps to estimate the dynamic index, which characterizes the efficiency of the transfer of technologies.

Keywords: science, industry, innovative infrastructure, transfer of technologies, model.

Постановка проблеми. Зростаюча актуалізація трансферу та комерціалізації технологій визначається тим, що сьогодні наукові установи та промислові підприємства все більше зацікавлені в прикладному використанні результатів своїх наукових та дослідно-конструкторських робіт (НДДКР). Результативність роботи лабораторій, інститутів, університетів, відділів НДДКР промислових підприємств у всьому світі все більшою мірою оцінюється з точки зору економічної ефективності комерціалізації та використання технологій як на внутрішньому, так і на міжнародному ринках.

Реальність сучасних взаємовідносин між наукою та промисловістю, знаннями та досвідом, технологічним процесом та продукцією, яка випускається, містить багато складностей та вимог, що потребують окремого розгляду та вивчення. Саме з погляду на перспективність цього напрямку діяльності слід відокремити ряд питань, щоб дозволяли виділити завдання на шляху до встановлення механізму співпраці наукового та виробничого секторів регіонів держави. Діагностика, системний аналіз, формування стратегії й тактики, все це є невід'ємними частинами здійснення трансферу.

Аналіз останніх досліджень і публікацій. Сьогодні аспекти трансферу технологій достатньо досліджуються вітчизняними й закордонними вченими. До найбільш значимих праць у цій області можна віднести роботи наступних авторів: Н. Арзамасцева, Н.І. Чухрай, В.В. Тітова, В.Г. Зінова, П.М. Цибульова, Л. Мінделі, А. Фурсенко, Н. Фонштейн та ін. Серед закордонних вчених слід особливо зазначити Х. Бремера, Р. де Вола, М. Портера, Р.Фостера.

Трансфер від одної стадії до іншої – це теоретичне позначення напрямку процесу. В дійсності, на думку Б. Санто [1], трансферу, як функціональної ланки немає, все виконується само собою, якщо система дійсно працює. Важливішим є надання системі відповідної свободи дій і сприятливих умов. На думку І. Дежиної і В. Салтикова [2, с.190], комерціалізація – це самий важливий елемент інноваційного процесу. Крім того, Жиц Г.І. та Квашніна М.Н. відмічають, що слабкою ланкою в системі практичної реалізації досягнень науки є комерціалізація результатів НДДКР [3].

Актуальність дослідження процесів трансферу технологій підтверджують роботи таких вчених, як Чухрай Н.І. [4], Фонштейн Н.М. [5], Шапошнікова А.А. [6], Мухіна А.П. [7], Лаптева Г.Д. [18], Тітова В.В. [9].

Незважаючи на наявність публікацій за даною тематикою, питання застосування і організації трансферу технологій і цілісний підхід до цієї наукового осмислення процесу трансферу технологій і механізму його реалізації в науково-інноваційній сфері регіону в сучасних умовах зумовлює необхідність здійснення досліджень даного питання.

Мета статті – моделювання процесу трансферу технологій для оцінки його ефективності.

Виклад основного матеріалу. Одне із основних завдань реалізації інноваційних процесів в регіонах є встановлення ефективного зв'язку науки з виробництвом і дієвих механізмів доведення науково-технологічної продукції до рівня товару. Як відомо, найбільш ефективною моделлю взаємодії науки і виробництва, з'єднуючою ланкою між розробником і споживачем інновацій, між науковою ідеєю і її практичним втіленням, є трансфертні механізми.

Трансфер технологій, якщо він добре організований, успішно й інтенсивно функціонує, стає основним способом реалізації стратегічних альтернатив інноваційного розвитку регіону. При цьому активізується цілеспрямована підприємницька діяльність, зменшується тривалість інноваційного процесу, вмикаються механізми планомірного пошуку і генерування інноваційних ідей, сприяння у їх реалізації. У той же час, високоорганізована передача технологій можлива лише за наявності розвиненої системи інфраструктурного забезпечення в регіоні [10].

Для оцінки інноваційного розвитку економічної системи запропонований узагальнений показник: $I^{\circ} = \varphi I^{\mu}$, де φ – динамічний індекс, що характеризує ефективність трансферу інноваційних технологій; I^{μ} – ресурсна складова для здійснення інноваційних процесів.

У [11] приведена методика оцінки інноваційного потенціалу I^{μ} , розглянуті приклади застосування даної методики до аналізу різних економічних систем. Наступним завданням є оцінка динамічного індексу φ , що характеризує ефективність трансферу технологій.

На рис. 1 представлена модель трансферу технологій стосовно мезоекономічної системи (регіон).

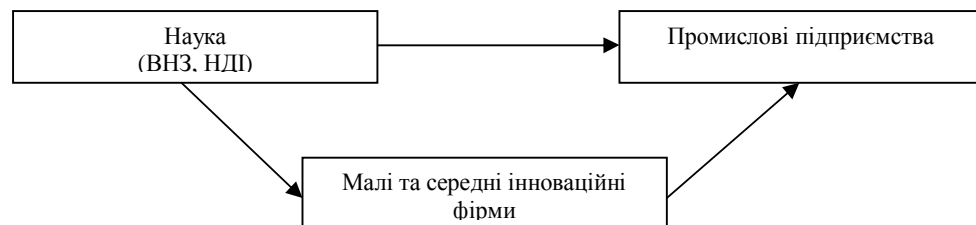


Рис. 1. Схема трансферу технологій у регіоні

Рис. 1 показує дві можливості:

- прямого впровадження результатів наукових досліджень науково-дослідних інститутів і вузів на промислових підприємствах;
- просування інновацій на промислові підприємства через інноваційні структури.

Для другого варіанту можна виділити такі стадії інноваційного процесу:

1) *етап наукової розробки проекту* – науково-технологічна ідея розробляється в дослідницькій організації – ВНЗ або НДІ. Розробка ідеї, як правило, йде в рамках виконання теми, що фінансується із держбюджету, або, маючи конкретного замовника промисловості, в рамках виконання договірної НДР;

2) *етап виробництва дослідних зразків чи малих серій інноваційної продукції* виконується малими і середніми інноваційними підприємствами, що тісно взаємодіють з науковими організаціями і ставлять своєю метою комерціалізацію результатів наукових досліджень;

3) *етап серійного виробництва* виконується крупними промисловими підприємствами. При цьому, метою інноваційного процесу є досягнення саме цієї стадії, на якій здійснюється

серійний випуск інноваційної продукції і її пропозиція на ринку товарів та послуг.

Інноваційний процес (рух від ідеї до результату у вигляді інноваційної продукції) є процесом проходження вищеперерахованих стадій. При цьому основна мета інноваційного розвитку полягає в досягненні максимальної ефективності цього процесу, яка визначається кількісно повнотою та швидкістю руху по інноваційному циклу. На рис. 2 представлений орієнтований граф, що ілюструє рух ресурсу за стадіями трансферу технологій. Ребра і вершини цього графу представлені суцільними або пунктирними лініями. Вершини, позначені пунктиром, відповідають основним стадіям інноваційного процесу. Із рисунку видно, що ці вершини є точкамигалуження, після яких інноваційний процес може йти різними шляхами. Вершини, зображені суцільною лінією, відповідають проміжним або кінцевим фінансовим результатам інноваційного процесу. Відповідно, по ребрах, зображених суцільною лінією здійснюється перехід продукту від однієї стадії до іншої, а по ребрах, зображених пунктирною лінією, – створення доданої вартості. Таким чином, сам трансфер технологій здійснюється по ребрах, зображених суцільною лінією, а супроводжуюча його зміна фінансового потоку – по ребрах, зображених пунктирною лінією.

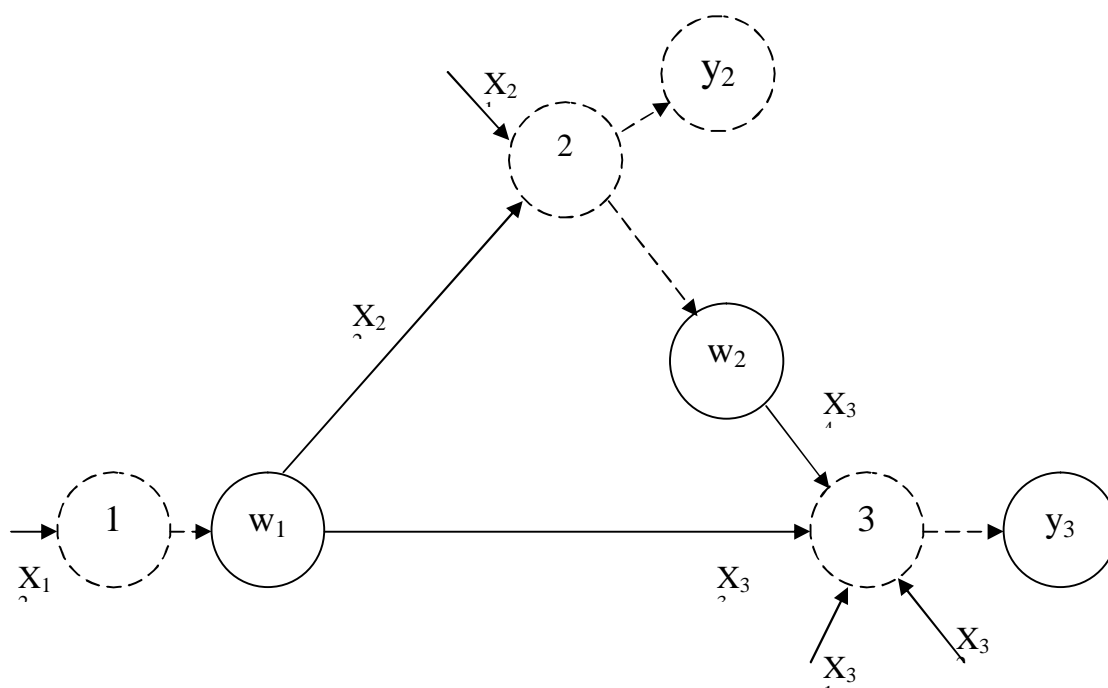


Рис. 2. Графічна інтерпретація розрахунків з трансферу інновацій

Для характеристики ефективності процесу трансферу технологій доцільно використовувати економічні показники – витрати, прибуток, виручку. Складемо для кожного з вузлів, зображених пунктирною лінією, рівняння балансу. У результаті отримаємо систему рівнянь:

$$\begin{cases} x_1 + p_1 = w_1, \\ x_2 + p_2 = w_2 + y_2, \\ x_3 + p_3 = y_3. \end{cases} \quad (1)$$

де x_i ($i = 1, 2, 3$) – витрати i -того вузла;

p_i ($i = 1, 2, 3$) – сума прибутку, який може бути отриманий від реалізації наукових розробок і інноваційної продукції для відповідних вузлів;

w_i ($i = 1, 2$) – проміжні результати інноваційного циклу, готові до трансферу технологій;

y_1 ($i = 2, 3$) – дохід від реалізації інноваційної продукції.

Розглянемо детальніше складові кожного вузла моделі.

1. Наука (НДІ і ВНЗ) – перше рівняння системи (1):

$x_1 = a_1 w_0$ – витрати на інноваційну складову наукової діяльності; w_0 – сумарний по регіону обсяг НДР і ДКР, що виконується науковими організаціями; a_1 – коефіцієнт ефективності вкладення коштів у наукові дослідження і розробки ($0 < a_1 < 1$);

w_1 – вартість об'єктів інтелектуальної власності (патенти, ліцензії, технічні звіти, статті і ін.), створені ВНЗ і НДІ у результаті наукової діяльності.

2. Малі і середні інноваційні фірми — друге рівняння системи (1):

x_2 – витрати малих і середніх підприємств на випуск інноваційної продукції, що включають наступні складові:

$b_2^1 x_2^1$ – витрати на освоєння і випуск власних розробок інноваційної продукції (x_2^1 – витрати на власні розробки інноваційної продукції; b_2^1 – коефіцієнт ефективності впровадження власних розробок, $b_2^1 \geq 0$);

$b_2^2 x_2^2$ – витрати на випуск інноваційної продукції при освоєнні об'єктів інтелектуальної власності, придбаних у НДІ і ВНЗ ($x_2^2 = c_{12} w_1$ – витрати на купівлю об'єктів інтелектуальної власності у НДІ і ВНЗ, c_{12} – коефіцієнт передачі, що характеризує сприйнятливність підприємств малого і середнього бізнесу в регіоні до науково-технічних розробок, здійснюваних у НДІ і ВНЗ, $0 \leq c_{12} \leq 1$, b_2^2 – коефіцієнт ефективності освоєння придбаних розробок, $b_2^2 \geq 0$);

w_2 – обсяг реалізованої продукції малими і середніми інноваційними підприємствами, яка може використовуватися промисловими підприємствами для серійного виробництва;

y_2 – обсяг реалізованої «під замовлення» малими і середніми підприємствами інноваційної продукції, не призначеної для подальшого використання у серійному виробництві.

3. Промислові підприємства – третє рівняння системи (1):

x_3 – витрати промислових підприємств на випуск інноваційної продукції, що включають наступні складові:

$b_3^1 x_3^1$ – витрати на освоєння і випуск власних розробок інноваційної продукції (x_3^1 – витрати на власні розробки інноваційної продукції, b_3^1 – коефіцієнт ефективності освоєння власних розробок, $b_3^1 \geq 0$);

$b_3^2 x_3^2$ – витрати на освоєння інноваційних технологій, придбаних за кордоном (x_3^2 – витрати на купівлю інноваційних технологій за кордоном, b_3^2 – коефіцієнт ефективності освоєння імпортованих розробок, $b_3^2 \geq 0$);

$b_3^3 x_3^3$ – витрати на випуск інноваційної продукції при освоєнні об'єктів інтелектуальної власності, придбаних у НДІ і ВНЗ ($x_3^3 = c_{13} w_1$ – витрати на купівлю об'єктів інтелектуальної власності у НДІ і ВНЗ, c_{13} – коефіцієнт передачі, що характеризує сприйнятливність промислових підприємств у регіоні до науково-технічних розробок, здійснюваних в НДІ і ВНЗ, $0 \leq c_{13} \leq 1$, b_3^3 – коефіцієнт ефективності освоєння розробок, придбаних у НДІ і ВНЗ, $b_3^3 \geq 0$);

$b_3^4 x_3^4$ – витрати на випуск інноваційної продукції при освоєнні об'єктів промислових зразків, придбаних у малих і середніх підприємств ($x_3^4 = c_{23} w_2$ – витрати на купівлю промислових зразків у малих і середніх підприємств, c_{23} – коефіцієнт передачі, що характеризує сприйнятливність крупними підприємствами розробок наукових організацій і підприємств малого і середнього інноваційного бізнесу, $0 \leq c_{23} \leq 1$, b_3^4 – коефіцієнт ефективності освоєння розробок, придбаних у малих і середніх підприємств, $b_3^4 \geq 0$);

y_3 – дохід промислових підприємств від випуску інноваційної продукції. Трансфер інноваційних технологій створює ефект мультиплікатора, який визначається таким співвідношенням:

$$y = \mu y'' \quad (2)$$

де $y = y_2 + w_2 + y_3$ – сумарний обсяг інноваційної продукції, виготовлений у регіоні малими і середніми, а також крупними промисловими підприємствами; y' – обсяг інноваційної продукції у регіоні у відсутності трансферу технологій ($c_{12} = c_{13} = c_{23} = 0$), $\mu = 1/(1-k)$ – мультиплікатор трансферу технологій; k – динамічний індекс, що характеризує ефективність трансферу технологій ($0 \leq k \leq 1$).

Індекс k позначає граничну схильність до трансферу технологій у регіоні (показує, яка частина інноваційної продукції отримана в результаті трансферу технологій). Вираз для k може бути отриманий як різниця (1) – (2) в припущенні, що $p_2 = p_3 = 0$:

$$k = \frac{y''}{y} = \frac{y''}{y' + y''}, \quad (3)$$

де $y' = x_2^1 b_2^1 + x_3^1 b_3^1 + x_3^2 b_3^2$ – обсяг інноваційної продукції, отриманої в результаті власних розробок і імпорту технологій;

$y'' = c_{12} w_1 b_2^2 + c_{13} w_1 b_3^3 + c_{23} w_2 b_3^4$ – обсяг інноваційної продукції, отриманої за рахунок трансферу технологій.

Таким чином, у відсутності трансферу технологій функціональний індекс $k=0$. Це означає, що інноваційна продукція локалізована на окремих підприємствах (або взагалі відсутня), немає руху в рамках інноваційного циклу, мультиплікатор дорівнює одиниці, інноваційна система регіону працює у край неефективно. На жаль, подібна або близька ситуація спостерігається в даний час у ряді регіонів України.

Разом з тим, ефективний трансфер технологій, при якому коефіцієнти передачі c_{12}, c_{13} і c_{23} близькі до одиниці, функціональний індекс $k > 0$, а мультиплікатор $\mu > 1$, дозволяє істотно збільшити обсяг інноваційної продукції у регіоні.

1. Санто Б. Сила инновационного развития / Б. Санто // Инновации. – 2004. – №2(69). – С.5–15.
2. Дежина И. Механизмы стимулирования коммерциализации исследований и разработок / И. Дежина, Б. Салтыков // Общество и экономика. – 2004. – №7–8. – С. 189–194.
3. Жиц Г.И. Коммерциализация новшеств и инновационный трансфер: некоторые подходы к трактовке понятий / Г.И. Жиц, М.Н. Квашнина // Инновации. – 2006. – №11(98). – С.47–51.
4. Чухрай Н.І. Трансфер і комерціалізація технологічних інновацій / Н.І. Чухрай // Економіка України. – 2002. – №3 (17). – С. 160–166.
5. Фонштейн Н.М. Трансфер технологий и эффективная реализация инноваций / Н.М. Фонштейн. – М.: Акад.нар.хоз. при правительстве РФ, 1999. – 294 с.
6. Шапошников А.А. Трансфер технологий: Определение и формы / А.А. Шапошников // Инновации. – 2005. – № 1(78). – С. 57–60.
7. Мухин А.П. Коммерциализация научно-технических разработок: учебно-практическое пособие / А.П. Мухин, Н.В. Арзамасцев, В.П. Ващенко, И.Д. Вершинин и др. – М.: АМРИП, 2001. – 173с.

8. Лаптев Г.Д. Трансфер технологий как экономическая категория / Г.Д. Лаптев. Режим доступа: [Электронный ресурс]. – <http://eng.ippt.demosite.ru>.
9. Титов В.В. Трансфер технологий / В.В. Титов. Режим доступа: [Электронный ресурс]. – [http://civt - 213a.ksu.ru/content/fag](http://civt-213a.ksu.ru/content/fag).
10. Соловйов В.П. Питання ефективної організації трансферу технологій в Україні / В.П. Соловйов // Інвестиції та інноваційний розвиток (науково-практичний бюлетень). – 2008 – №3. – С.14–19.
11. Максимов Ю. Инновационное развитие экономической системы: оценка инновационного потенциала / Ю. Максимов, С. Митяков, О. Митякова, Т. Федосеева // Инновации. – 2006. – №6. – С. 53–57.

УДК 681.513.2

Рязанцев О.І., Кардашук В.С., Бортник К.

Східноукраїнський національний університет імені Володимира Даля

СИСТЕМА АВТОМАТИЗАЦІЇ КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ ВИРОБНИЦТВА ЖЕЛАТИНУ

Рязанцев О.І., Кардашук В.С., Бортник К. Система автоматизації керування технологічним процесом виробництва желатину. Розглянуто модернізовану систему автоматизації керування виробництва желатину. Проведена декомпозиція технологічного процесу та визначений перелік вхідних та вихідних сигналів. Запропоновані програмно-апаратні засоби реалізації системи керування. Для розроблення програмного забезпечення використано бібліотеку алгоритмічних елементів, що описують елементи керування. Наведені переваги використання модернізованої системи.

Ключові слова: система керування, програмно-логічний контролер, графічний інтерфейс.

Рязанцев А.И., Кардашук В.С., Бортник К. Система автоматизации управления технологическим процессом производства желатина.

Рассмотрена модернизированная система автоматизации управления производства желатина. Проведена декомпозиция технологического процесса и определен перечень входных и выходных сигналов. Предложены программно-аппаратные средства реализации системы управления. Для разработки программного обеспечения использовано библиотеку алгоритмических элементов, которые описывают элементы управления. Наведены преимущества использования модернизированной системы.

Ключевые слова: система управления, программно-логический контролер, графический интерфейс.

Ryazantsev A.I., Kardashuk V.S., Bortnik K. The automation system of process control the production of gelatine.

We consider the modernized system automation of the production of gelatine. Spend the decomposition process and a list of input and output signals. Proposed software and hardware implementation of the control system. For software development use the library of algorithmic elements that describe the controls. Hover advantages of modernized system.

Keywords: System management, software and program logic controller, graphical interface.

Актуальність проблеми. Ріст промислового виробництва, технічна модернізація технологічних процесів (ТП) супроводжується вирішенням питань пов'язаних з впровадження у виробництво новітніх апаратних засобів систем автоматизації керування (САК) на базі сучасних програмно-логічних контролерів (ПЛК). Робота з модернізації САК ТП виробництва желатину проводилася в рамках науково-технічної програми співпраці між кафедрою комп'ютерної інженерії Східноукраїнського національного університету імені Володимира Даля та науково-виробничим підприємством «Уніконт» (м. Северодонецьк), що займається розробкою і впровадженням ПЛК і робочих станцій (РС) для інформаційних систем та САК ТП різноманітного призначення [1].

Аналіз досліджень об'єкта керування. За результатами дослідження виконана декомпозиція об'єкту керування, для розроблення програмного забезпечення системи визначений перелік вхідних та вихідних сигналів, визначені функції системи. Кількість аналогових сигналів від об'єкту керування – 49. Кількість вихідних дискретних сигналів – 24. САК забезпечує автоматичне програмне керування ТП. Кількість вихідних інформаційних сигналів стану, що відображаються на інформаційному пульті оператора-технолога – 31.

Рішення задачі. Модернізована САК ТП виробництва желатину створена на діючому виробництві в рамках технічної модернізації об'єкта керування та виконує інформаційні та керуючі функції. Рішенням даної проблеми, згідно аналізу стану об'єкту керування, є використання ПЛК Уніконт-743/1 на базі процесорного модуля Advantech PCA-657F з необхідним набором модулів. Схему взаємодії компонентів системи та склад її основних модулів наведено на рис. 1.

САК ТП виробництва желатину є програмно-технічним комплексом, який складається з технічних засобів та програмного забезпечення і забезпечує автоматизацію керування та контроль стану вихідних параметрів (температури, стану кранів, концентрацію речовини та ін.). Керування процесом на нижньому рівні здійснюється за допомогою ПЛК з необхідним набором апаратних та програмних модулів, який забезпечує збір, первинну обробку інформації, видачу сигналів керування.



Рис. 1. Структура системи автоматизації керування

Перелік та характеристика модулів ПЛК наведена в табл. 1.

Таблица 1

Перелік модулів ПЛК

Назва	Опис та характеристика модуля
БПр PCA-657F	Блок процесорний Advantech PCA-657F.
АЦП-60	Аналогово-цифровий перетворювач. Кількість каналів – 60.
МН-12	Модуль нормалізації. Кількість каналів – 12.
МНТП/ТО-8	Модуль нормалізації термopара/термоопір. Кількість каналів – 8.
МВДС-64КО	Модуль введення дискретних сигналів з контролем обриву лінії. Кількість каналів – 64.
МФДС-64	Модуль формування дискретних сигналів. Кількість каналів – 64.

Діючі контури контролю та виконавчі пристрої розташовані безпосередньо на місці розташування об'єкту керування. Для забезпечення централізованого контролю за станом параметрів технологічного процесу, виміром, відображенням, реєстрацією, сигналізацією відхилень параметрів і показників стану обладнання, а також функцій програмного і безпосереднього управління ТП, передбачається застосування ПЛК у промисловому виконанні IP54.

ПЛК здійснює введення технологічних параметрів і реалізацію керуючих впливів на виконавчі механізми котла згідно алгоритму керування. Крім того, за допомогою вбудованого пульта в ПЛК є можливість керувати ТП обслуговуючим персоналом за місцем розташування котла, що підвищує надійність функціонування системи в цілому.

Система дозволяє представляти інформацію технологу або начальникові цеху про стан технологічного об'єкта, формування звіту подій на об'єкті, зберіганні інформації, контроль роботи обслуговуючого персоналу та є відкритою для зв'язку по локальній мережі Ethernet (RS-485) підприємства з подальшими об'єднанням в мережу з іншими відділеннями желатинового цеху.

САК ТП забезпечує традиційні для систем такого роду функції:

1. Автоматичний контроль, керування та візуалізацію з використанням графічного інтерфейсу.
2. Автоматичний контроль стану модулів системи та виконавчих механізмів (датчики, крани).
3. Підвищення якісних показників ведення процесу за рахунок збільшення оперативності і точності вимірів, контроль дій оператора-технолога.
4. Ведення архіву процесу, стану системи та дій технологічного персоналу.

5. Діагностику ПЛК, облік аварійних та попереджувальних повідомлень системи з реєстрацією відповідних дій персоналу та ряд ін.

Основним інструментом технолога-оператора при роботі є система відображення, в якій відображаються результати роботи і переключення між контурами керування (контур нагріву котла, кранів подачі реагентів та ін.).

Для реалізації функцій керування в системі відображення екран оператора розділений на технологічні зони у відповідності з функціональним призначення. Система відображення на робочій станції дозволяє здійснювати переключення між вікнами мнемосхем та контролювати стан ТП та проводити діагностику модулів ПЛК. Перегляд даних здійснюється за допомогою закладок та написів на технологічній схемі. Для розробки системи відображення використано Photon Application Builder (PhAB) фірми QSSL. Цифровий ПІД-регулятор температури має паспорт в якому відображені діючі значення коефіцієнтів настроювання. Фрагмент мнемосхеми ТП представлено на рис. 2.

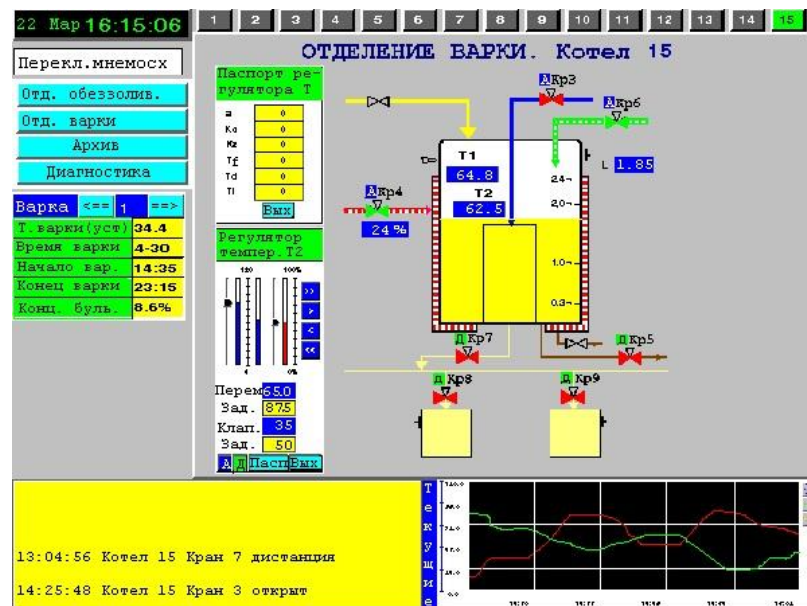


Рис. 2. Фрагмент мнемосхеми технологічного процесу

При розробленні програмного забезпечення використана бібліотека алгоритмічних модулів, що описує функції кожного з апаратних модулів ПЛК.

Таблиця 2

Склад програмних модулів ПЛК

Позначення	Характеристика
ASPIIN	Модуль ініціалізації АЦП.
ASPID	Модуль АЦП з діагностикою.
ASPIL	Модуль перетворення цифрового коду в фізичну величину.
RAS16	Модуль розпаковки 16-розрядний.
S_FILTER	Модуль фільтрації аналогового сигналу.
MVDCK	Модуль МВДС з діагностикою.
MFDSZD	Модуль формування стану дискретних каналів та діагностики його працездатності.
RPID	Модуль ПІД-регулювання.

Збір інформації з датчиків, керування виконавчими пристроями здійснюється з використанням модулів зв'язку з об'єктом [2]. Зв'язок з датчиками здійснюється через модуль МНТП/ТО за допомогою стандартних рівнів сигналів. Попереднє з'єднання з МНТП/ТО, при подальшій подачі сигналів на АЦП, забезпечує якісний прийом сигналів в умовах промислових

перешкод. Виходи датчиків подаються до блоку МНТП/ТО, де проводиться нормалізація сигналів, а потім в блок АЦП для перетворення аналогового значення сигналів в цифровий код.

Програмний елемент обробки аналогового сигналу ASPIL (рис. 3, табл. 3-4) забезпечує прийом цифрового коду з виходу модуля АЦП, що відповідає значенню напруги, перетворення його в фізичну величину та діагностику модулю.

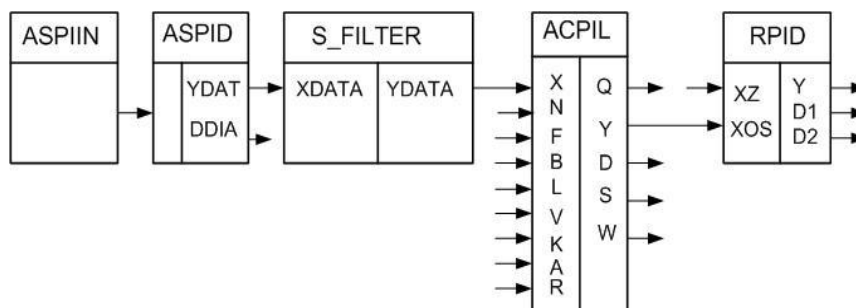


Рис. 3. Прийом аналогового сигналу та видача керування

Таблиця 3

Вхідні параметри елементу обробки аналогового сигналу ACPIL

Позн.	Вид сигналу	Призначення
X	Аналоговий (int)	Код в діапазоні 0-4096, що вводиться з АЦП (12 розрядів).
N	Дискретний	Признак: датчик включений/виключений (0/1).
F	Аналоговий (int)	Параметр фільтру (0,3 – 1).
H	Аналоговий (int)	Нижня границя достовірності (в одиницях виміру фізичної величини).
B	Аналоговий (int)	Верхня границя достовірності (в одиницях виміру фізичної величини), $B > H$.
L	Аналоговий (int)	Нижня уставка сигналу попередження (в одиницях виміру фізичної величини), $L \geq H$.
V	Аналоговий (int)	Верхня уставка сигналу попередження (в одиницях виміру фізичної величини), $V \leq B$.
K	Аналоговий (int)	Умова видачі значення в мережу, $K > 0,1 \%$.
A	Аналоговий (int)	Мінімум діапазону виміру (в одиницях виміру фізичної величини).
R	Аналоговий (int)	Максимум діапазону виміру (в одиницях виміру фізичної величини), $R > A$.

Таблиця 4

Вихідні параметри елементу обробки аналогового сигналу ACPIL

Позн.	Вид сигналу	Призначення
Q	Аналоговий (int)	Значення параметру для видачі в мережу.
Y	Аналоговий (int)	Виміряне значення параметру.
D	Дискретний	Признак: 1 – датчик виключений або порушена одна з технологічних границь.
S	Дискретний	Признак: 1 – значення параметру менше або дорівнює мінімуму уставки сигналізації ($Y \leq L$).
W	Дискретний	Признак: 1 – значення параметру більше або дорівнює максимуму уставки сигналізації ($Y \geq R$).

Прийнятий з АЦП цифровий код перевіряється на допустимість та розраховується по формулі:

$$z = F \cdot x + (1 - F) \cdot G, \quad (1)$$

де x - значення коду після фільтру в діапазоні 0-4096, що вводиться з АЦП (12 розрядів);
 F - параметр фільтру (0,3 – 1);
 G - значення параметру після фільтру попереднього кроку (початкове значення $G=1000$).

Розрахунок значення в одиницях параметру виміру:

$$Y = A + (R - A) \cdot \frac{z}{4096}, \quad (2)$$

де R - максимум діапазону виміру (в одиницях виміру фізичної величини), 10В;
 A - мінімум діапазону виміру (в одиницях виміру фізичної величини), 0В.

При розходженні сигналу від датчика та завдання ПІД-регулятора виробляється вплив на відповідний кран об'єкта керування [3].

Модуль MVDCK забезпечує введення стану дискретних сигналів від виконавчих пристроїв та їх діагностику. Елемент RAS16 забезпечує перетворення вхідної аналогової інформації в n -розрядний позиційний код.

При прийомі дискретних сигналів вхідний сигнал представляється виразом:

$$X = X_{n-1} * 2^{n-1} + X_{n-2} * 2^{n-2} + \dots + X_0 * 2^0, \quad (3)$$

де n - кількість розрядів (1-16); $X_i \in \{0,1\}$.

Специфіка визначення середніх і сумарних значень в задачах керування обумовлена дискретним характером вхідної інформації. З цією метою в ланцюжку прийому аналогового сигналу запропоновано використати програмний модуль S_FILTER та алгоритм фільтрації аналогового сигналу, який забезпечує обчислення середнього значення останніх n -значень. Перед початком роботи значення кількості вимірів n заноситься як параметр налаштування.

Програмне забезпечення реалізоване на мові програмування Watcom C++ з застосуванням Photon microGUI. ПЛК та РС функціонують під керуванням операційної системи реального часу QNX v.4.25. САК ТП – одна з традиційних областей застосування сімейства операційної системи QNX, а використання інтернет-технологій на сучасному етапі для віддаленого моніторингу та керування ТП є доповненням керування на нижньому рівні.

Зв'язок між ПЛК та робочою станцією здійснюється за допомогою мережевого інтерфейсу Ethernet (RS-485).

Програмна реалізація системи забезпечує:

1. Функціонування та реакцію системи на подію з тактом роботи 200 мс.
2. Регулювання температури по закону ПІД-регулювання, динамічне налаштування коефіцієнтів системи.

3. Видачу керуючих впливів 24 В в такті не більше 200 мс для виконавчих механізмів.

4. Передачу оперативної інформації по мережі Ethernet (RS-485) на робочу станцію для підсистеми відображення трендів, архівів, підсистеми діагностики стану модулів ПЛК та ряд ін.

Розроблене програмне забезпечення ПЛК та методи керування дозволили:

- реалізувати видачу керуючих впливів для виконавчих механізмів 24 В в такті не більше 200 мс з похибкою виміру параметрів 0,1 %;

- отримати до 30 % економії електроенергії та до 20 % витрати води та ряд інших переваг.

Висновки. Модернізована АСК ТП виробництва желатину дозволила використати сучасні засоби автоматизації ТП. Основні теоретичні положення і результати розроблення АСК ТП перевірялись з використанням програмних засобів на базі операційної системи реального часу QNX v.4.25 фірми QSSL (QNX Software System Ltd.), бібліотеки алгоритмічних модулів «Уніконт», мови програмування та транслятора Watcom C++, графічної оболонки Photon

microGUI, системи відображення Photon Application Builder фірми QSSL, SCADA-системи „Уніконт-М”, SCADA-системи „Кварц”, ПЛК Fastwell на базі процесора CPU686 фірми Octagon Systems.

1. Рязанцев О. І. Реалізація функцій керування на нижньому рівні в системі автоматизації керування процесом аерозольного нанокаталізу / О.І. Рязанцев, В.С. Кардашук // Вісник Східноукраїнського національного університету ім. Володимира Даля. – № 15 (186), ч. 2, 2012. – С. 170–175.
2. Рязанцев А.И. Экспериментальные исследования интегральных преобразователей напряжения и их применение в цепях гальванической развязки /Рязанцев А.И., Иванов А.Н., Кардашук В.С. // Проектирование радиоэлектронных и лазерных устройств и систем. ФГБОУВПО "Воронежский государственный технический университет". – Воронеж, 2013. – С. 106-116.
3. Кардашук В.С. Математична модель підсистеми керування температурою для процесу знешкодження відходів хлорорганічного синтезу / Кардашук В.С., Пономарчук О.В., Маринич М.В. // 1-а міжнародна науково-практична конференція «Теоретичні і прикладні аспекти комп'ютерних наук та інформаційних технологій» TACSIT-2015, 15-16 травня, м. Сєверодонецьк. – С. 21 – 24.

УДК 629-1.47

Федік Л.Ю., Балинський А.О.

Луцький національний технічний університет

ОСОБЛИВОСТІ КОНСТРУКЦІЇ І КЕРУВАННЯ АВТОМОБІЛЯМИ АМФІБІЯМИ

Федік Л.Ю., Балинський А.О. Особливості конструкції і керування автомобілями амфібіями. У статті викладені основні види автомобілів амфібій. А також їх застосування, конструктивні особливості і керування. Зокрема таких видів: GMC ACKWX 353, DUKW-353, Spira4u, WaterCar Parther, ГАЗ-46 "МАН", ЗИС-485 «БАН», ЛуАЗ-967, ВАЗ-Е2122, УАЗ-3907 «Ягуар», «Амфікара», Hydrocar, «АКВАДОМ», «Aqyada», «Watercar», Rinspeed Splash, Сплеск, Panther.

Ключові слова: автомобіль, амфібія, швидкість, двигун, потужність, плаваючі.

Федик Л.Ю., Балинский А.А. Особенности конструкции и управления автомобилями амфибиями. В статье изложены основные виды автомобилей амфибий. А также их применения, конструктивные особенности и управления. В частности таких видов: GMC ACKWX 353, DUKW-353, Spira4u, WaterCar Parther, ГАЗ-46 "имел", ЗИС-485 «БАН», ЛуАЗ-967, ВАЗ-Е2122, УАЗ-3907 «Ягуар», «Амфикара», Hydrocar, «АКВАДОМ», «Aqyada», «Watercar», Rinspeed Splash, Всплеск, Panther.

Ключевые слова: автомобиль, амфибия, скорость, двигатель, мощность, плавающие.

Fedik L.Y., Balynskyy A.O. Features konstruksyy and control cars amfybyyamy. The article describes the main types of amphibious vehicles. As well as their applications, design features and controls. In particular the following species: GMC ACKWX 353, DUKW-353, Spira4u, WaterCar Parther, GAZ-46 "MAN" ZIS-485 "BAS" luaz-967, VAZ-E2122, UAZ-3907 "Jaguar", "Amfikara" Hydrocar, «AQUADOM», «Aqyada», «Watercar», Rinspeed Splash, Splash, Panther.

Tags: car, amphibian, speed, engine, power, floating.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими та практичними завданнями. Історія появи автомобілів бере свій початок у далекому XVIII столітті. На даний момент існує велика кількість їх видів.

Автомобілі амфібії являють собою транспортний засіб, що має особливість пересуватися як по землі, так і по воді. Тому їх застосування доцільне у військовій промисловості, наприклад, плаваючі танки і БТРи.

Аналіз останніх досліджень і публікацій, в яких започатковано розв'язання даної проблеми і на які спирається автор. Дослідження особливостей конструкції і принципів роботи автомобілів амфібій займалися фірми-розробників і ентузіасти: Спіра, WaterCar, Гббс, Rinspeed, КАМИ, Gibbs Technologies, Rinspeed і ін.

Виділення невирішених раніше частин загальної проблеми, котрим присвячується стаття. Огляд існуючих моделей автомобілів амфібій і їх особливості керування.

Формулювання цілей статті (постановка завдання). Огляд літературних джерел про конструктивні особливості, принципи роботи і застосування автомобілів амфібій.

Виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів.

Автомобіль амфібія GMC ACKWX 353 (рис.1) була поставлена в серійне виробництво корпорацією General Motors навесні 1942 року. Корпус машини являв собою плоскодонний човен понтонного типу, зварений з листової сталі товщиною близько двох міліметрів. Човен не був несучою конструкцією - всередині корпусу знаходилося звичайне шасі з незначними змінами вузлів, викликаними специфікою застосування.

У днищі човна цієї конструкції машини були зроблені виїмки для коліс осей карданних валів і гребного гвинта. Корпус амфібії був розділений перегородками на три частини - носову, десантну і кормову. У носовій розташовувалися двигун і радіатор, доступ до яких був можливий через два спеціальних люка. Перший люк забезпечував обслуговування радіатора, а також глушника і слугував для виходу нагрітого повітря з моторного відділення. Другий люк - забезпечував безпосередній доступ до двигуна. У носовій частині також розташовувалося відділення управління - панель приладів, кермо, сидіння шофера і праве сидіння для його помічника або командира.

Це відділення спереду було захищене вітровим склом, з боків - престижними брезентовими

боковинами, а зверху міг натягатися тент. На додаток до звичайних органів керування GMC ACKWX 353 у цьому відділенні знаходилися важелі для включення гвинта, клапанів насосів, а також тумблери для включення підкачки шин. На відміну від сухопутного базового шасі задні колеса були односхилими. Підвищення прохідності і кліренсу забезпечувалися десятишаровими покриттями 11,00-18 замість 7,5-20 звичайних вантажівок. Централізована підкачка шин, яка була застосована на цьому автомобілі, зробила GMC DUKW першою серійною американською машиною з подібною системою.

Проте, ходова частина і підвіска амфібії не відрізнялися від базової вантажівки. Десантне відділення на 25 осіб мало внутрішні розміри 3780x2080x710 мм (коли розміри амфібії були 9450x2440x2690 мм), екіпаж складався з двох чоловік. Замість десанту ця амфібія могла перевозити 2340 кг вантажу. Морське застосування цих машин зумовило і появу в характеристиках нетипових водних параметрів: висота надводного борту (від ватерлінії до палуби) в носовій частині 584 міліметра, в кормовій частині - 457 мм, висота осадку до передніх колес - 1,12 м, по задніх колесах - 1,24 м. Радіус циркуляції на воді 6,2 м, запас ходу на воді 62 км. Гранична висота хвиль у прибіжній зоні близько трьох метрів. На воді амфібія приводилася в рух трилопатеvim гвинтом (діаметр 635 мм). На початковому етапі створення амфібії виконали роботи по збільшенню швидкості руху на воді з 8 до 10 км/год. За рахунок раціонального розміщення гребного гвинта в тунелі, кращого підбору параметрів гребного гвинта, екранування колісних шин і інших заходів. Для відкачування води, що потрапила в корпус машини, було встановлено два насоси: шестеренний і відцентровий, що приводилися в дію від валу гребного гвинта.

Бак для пального розташовувався праворуч у задній частині вантажного відділення. У кормовій частині знаходилася лебідка для самостійного витягування амфібії на важкопрохідних ділянках дороги. Проте, і без лебідки прохідність амфібії була дуже не погана: крутизна подоланого підйому, особливо актуальна при виїзді на берег, становила 31 °, радіус повороту на суші - 11 м. Машин амфібії GMC DUKW 353 було виготовлено близько 21000 одиниць. Ці спеціальні автомобілі були основоположниками більшості американський вантажівок різних напрямків, а ті у свою чергу - основоположниками практично всіх вантажівок і армійської техніки CPSP [2].

Довгоподібний GMC ACKWX 353 став основою для тривісної армійської амфібії DUKW-353, рис. 2. Випуск цієї машини почався у квітні 1941 р. концерном «Дженерал Моторс» спільно з нью-йоркською суднобудівної фірмою «Спаркмен і Штефенс». Вперше серійно випустили вантажний автомобіль амфібію, всі мости якої були оснащені односхилими колесами, що рухаються слід у слід і не створюють додаткового опору руху. Колеса при цьому оснащені спеціальними десятишаровими еластичними шинами, що допускають експлуатацію зниженого тиску. А це у свою чергу різко підвищило опорну поверхню колеса і прохідність на слабких ґрунтах. У цій машині вперше було сконструйоване і застосоване до шасі № 2005 принципове нововведення - централізоване керування тиском повітря в шинах.

За даними офіційного сайту Берегової охорони США, у віданні якої знаходяться всі земноводні транспортні засоби країни, на 25 червня 2002 року тільки в США 75 амфібії DUKW-353 використовувалися в комерційних цілях і 140 - під державною юрисдикцією на несудноплавних водних шляхах. Навіть Радянська влада замовляла для Червоної Армії із США по ленд-лізу в 1943-44 роках - 586 амфібії.

Особливостями цієї машини-амфібії є її вантажопідйомність на суші 2250 кг, найбільша швидкість руху по суші 80 км/год, по воді - 10 км/год, маса 6,5 т, тип і модель двигуна GMC (6 циліндрів, 4,4 л, карбюраторний), потужність - 91,5 к.с., запас ходу 354 км (по суші) і 80 км (по воді) [3].



Рис. 1. Автомобіль GMC ACKWX 353



Рис. 2. Автомобіль DUKW-353

Компанія Спіра, в Китаї, розпочала серійний випуск експериментального триколісного автомобіля на електричному та газовому двигунах Spira4u, рис.3. Електрична версія важить близько 236 кг, а газова (з двигуном внутрішнього згорання) - близько 200 кг. У автомобіля композитне шасі, корпус зі скловолокна і панелі з м'якої піни. Саме завдяки спіненим матеріалам автомобіль може пересуватися не тільки по дорогах, але і по воді.

У газовій версії використовується двигун об'ємом 150 см³ з китайського мотоцикла Wangye. Його максимальна швидкість - 85 км/год. Особливістю автомобіля є оснащення автоматичною коробкою передач, низька витрата палива - 2,94 л на 100 км і високий запас ходу - 320 км. За додаткову плату можна встановити додатковий двигун на задні колеса та додатковий акумулятор, який подвоїть діапазон пробігу до 230 км. З одним електродвигуном автомобіль розганяється до максимальної швидкості в 100 км/год.

Триколісний мікроавтомобіль амфібія вирішує відразу кілька проблем - економічні, екологічні та зручність пересування на дорогах у місті. На ньому можна їздити по шосе і плавати по воді, а паркувати незвичайний автомобіль можна, навіть, у вертикальному положенні [4].



Рис. 3. Автомобіль-амфібія Spira4u

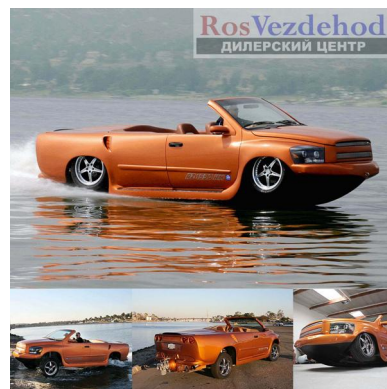


Рис. 4. Автомобіль-амфібія WaterCar Panther

Автомобілі амфібії, що здатні пересуватися по дорогах загального користування як і прості автомобілі, ставляться на облік в ДАІ, а для переміщення по воді – у ДУМС (державній інспекції по маломірних суднах). Конструкція днища виконана таким чином, щоб розвивати максимальну швидкість на воді. Переходячи з суші на воду амфібії Watercar з легкістю переходять на швидкість глисування і за лічені секунди можуть перетворюватися в катери, які здатні швидко і легко пересуватися по воді.

За основу, у моделі WaterCar Panther, був узятий легендарний джип Wrangler, рис. 4. У базовій

комплектації встановлені вінілові сидіння та приладова панель джипа Wrangler, доповнена приладами для роботи водомета і індикатором переходу з "водного" режиму в "дорожній".

Сидіння розроблені таким чином, щоб зробити поїздки максимально комфортно незалежно від нерівностей поверхні дороги. Вони виготовлені з нержавіючої сталі і морського вінілу з метою, отримання по справжньому морського інтер'єру. З метою ліквідації процесу корозії, майже вся машина виготовлена з нержавіючої сталі, а решта - або з епоксидної смоли, або з інших нержавіючих матеріалів. "Пантера" спроектована для тривалого застосування в агресивній, солоній воді. Його вага становить 1338 кг. Може розвивати швидкість руху по воді до 80 км/год, на землі – до 190 км/год. Застосовується двигун Honda 3,7-літровий VTEC (250 к.с.)

Найшвидшим авто вважається WaterCar Python, яке виготовляє компанія WaterCar з класу автомобіля Chevrolet Корвет. Даний автомобіль розганяється до швидкості 100 км/год за 4,5 секунди, а максимальна його швидкість на асфальті становить 160 км/год. Вага машини 1725 кг. На воді можна спокійно відкрити двері автомобіля і при цьому салон автомобіля буде абсолютно сухим. Двигун у автомобіля амфібії використовуються V8 від Corvette, від LS1 до новітнього LS9 6,2 літра у якого потужність складає 640 к.с, також як у ZR1. Щоб перетворитися в човен, потрібно всього 2-3 секунди. Під час виробництва, водієві пропонують вибрати колір кузова, з 60000 кольорів і колір салону з 4000000 кольорів.

На воді автомобіль амфібії пересувається за допомогою водомета, а його колеса прибираються завдяки гідравлічному механізму. У автомобіля, рама виготовлена з прямокутного профілю і встановлена в кузов автомобіля, а дно - з якісного склопластику, у вигляді букви V. Вартість даного автомобіля амфібії становить 150000 доларів.

Автомобіль Aquada, що виготовляється фірмою Gibbs Technologies, виглядає як автомобіль Mazda 5 з незвичайними морськими бамперами. Він також без дверей, відмінно їздить по дорогах, і плаває по воді. Водійське сидіння і всі органи керування знаходяться посередині, а пасажирські місця розташовані з боків. Під час руху по воді, пасажирські місця можна підняти вгору. На воді як у більшості автомобілів амфібій колеса прибираються в арки натисненням однієї кнопки. Двигун - 2,5-літра V6 Land Rover Freelander потужністю 175 к.с, задній привід. Двигун з легкістю розганяє амфібію на трасі до 160 км/год, а на воді максимальна швидкість - 48 км/год. Час трансформації в катер, складає всього 6 секунд. Вартість даного автомобіля амфібії в Англії від 139,000 до 260,000 доларів США.

Майже всі виробники автомобілів-амфібій, такі як: Гіббса, КАМІ, Aquada і WaterCar, щоб не створювати опору під час руху автомобіля, зробили можливість піднімання його колес, а дно як у звичайних човнів.

Серед розробників автомобілів амфібій є й такі, які прагнуть зробити, щось нове та незвичайне. Так Rinspeed створив унікальний автомобіль Сплеск на підводних крилах, які піднімають автомобіль на 0,5-1 метри над водою! Автомобіль-амфібія на підводних крилах Rinspeed Сплеск може досить швидко і з легкістю пересуватися по землі, а при необхідності може в будь-який момент звернути з дороги і продовжити рух по воді. Ця модель на сьогоднішній день є однією з самих незвичайних автомобілів амфібій. Оскільки може пересуватися по воді за допомогою підводних крил.

За допомогою системи підводних крил, яка піднімає концепт на цілих пів метра від поверхні води, і тим самим дає перевагу на воді розвивати швидкість до 84 км/год. На звичайних дорогах, його максимальна швидкість 200 км/год, до сотні розгін складає всього 6 секунд. Для використання підводних крил, потрібна глибина води від 1 метра, якщо глибина менше 1 метра можна пересуватися за допомогою гвинта, зі швидкістю 50 км/год. Підводні крила можуть піднімати авто при швидкості 30 км/год. У автомобіля амфібії підводні крила знаходяться в порогах, тим самим вони захищені і з легкістю можна повернути їх на 90 градусів вниз, а заднє крило під час їзди, виконує роль анти-крила. Тим самим притискає задню частину амфібії до землі. У авто без пасажирів вага становить всього 800 кг, двигун 750 к.с., 140 м³. Автомобіль обладнаний турбіною, яка працює на звичайному природному газі. Під час використання турбіни, потужність передається по системі на задні колеса.

Автоамфібія Hydra Spyder виготовляються американською компанією C.A.M.I. Ця компанія,

також виробляє автобуси С.А.М.І Terra Wind і Hydra Terra. Автомобіль амфібія, представлений як спортивне авто і був показаний у фільмі з агентом 007 "І цілого світу мало". Автомобіль Hydra Spyder може спокійно вмістити в себе чотирьох чоловік, і ще тягнути одну людину на водних лижах. У базовій комплектації на амфібії встановлений 6-літровий двигун V8, LS2 Корвет потужністю 400 к.с, також є моделі з турбонадувом, у них двигун 502 к.с. CDI Chevy 500 к.с, з переднім приводом.

Ця модель амфібії оснащена двома передніми спортивними сидіннями і плоским заднім. Вантажопідйомність її становить 3300 кг. Ця вага розподіляється на перед автомобіля на 53% і на зад 47%. При в'їзді у воду водієві достатньо натиснути на кнопку і автомобіль автоматично, за допомогою пневматичної системи, з легкістю прибере колеса, змінить тип приводу і зможе розігнатися на воді до швидкості 80 км/год.

Оскільки, для військових водоплавна машина - річ часом незамінна, то за радянських часів у нашій країні було побудовано чимало амфібій, а деякі навіть потрапили в серійне виробництво [5].

Малий автомобіль водоплавний (скорочено МАВ), рис. 5, почали випускати на Горьківському автомобільному заводі в 1953 році. Машина призначалася для забезпечення дій розвідувальних підрозділів, переправи десантників і для проведення інженерних робіт на воді. ГАЗ-46 оснащувався чотирициліндровим двигуном від ГАЗ-М20 «Перемога», від ГАЗ-69 була запозичена трансмісія і підвіска, а для руху по воді використовувався гребний гвинт. У цілому модель була скопійована з американського Форд ГРА. Випуск «МАВ» тривав до 1958 року, а потім виробництво було передане на завод УАЗ. Однак, там не знайшли коштів для випуску амфібії.



Рис. 5. Автомобіль ГАЗ-46 «МАВ»



Рис. 6. Автомобіль ЗІС-485 «БАВ»

«БАВ» 1950 року, на відміну від «Мава» - це великий водоплавний автомобіль, рис. 6. Конструкцію цієї машини радянські інженери запозичили у американської амфібії GMC DUKW-353. Машина оснащувалася двигуном у 110 к.с., точно таким же, як і на БТР-152. ЗІС-485 перебував у виробництві 12 років і слугував для переправи особового складу десанту і автомобілів. У «БАВ» розміщувалися 25 осіб або 25 тонн вантажу, включаючи автомобілі і артилерійські знаряддя.

Повнопривідний плаваючий автомобіль-транспортер особливо малої вантажопідйомності ЛуАЗ-967 був створений на замовлення ВДВ для евакуації поранених, підвозу боєприпасів і військово-технічного майна, буксирування, а також установки окремих видів озброєння, рис. 7. Необхідність такої моделі з'явилася в роки Корейської війни, де вона проявила себе як масивна амфібія. ЛуАЗ-967 відрізнявся маленькими габаритами, оснащувався двигуном об'ємом менше одного літра, а на воді приводився в рух колесами - гребні гвинти у нього були відсутні. Характерна особливість конструкції - відкидна рульова колонка, розташована по центру салону: при необхідності водій міг управляти ЛуАЗом-амфібією в напівлежачому стані.



Рис. 7. Автомобіль ЛуАЗ-967



Рис. 8. Автомобіль ВАЗ-Е2122

ВАЗ-Е2122 був сконструйований в 1976 році на замовлення Міноборони СРСР із застосуванням вузлів і агрегатів «Ниви», рис. 8. Від своїх аналогів водоплавна «Нива» відрізнялася унікальним дизайном. Передня частина автомобіля, нагадує Lamborghini LM002. Автомобіль оснащувався 1,6-літровим двигуном, мав унікальну прохідність і міг пересуватися по воді зі швидкістю 4-5 км/год. Роком пізніше був представлений другий варіант ВАЗівської амфібії. Від першого він відрізнявся поліпшеними системами охолодження, посиленням кузова і зміненим місцерозміщенням сидінь. Однак, ні перша, ні друга модифікація ВАЗ-Е2122 так і не отримали масового випуску.

УАЗ-3907 «Ягуар» - це ще один перспективний автомобіль амфібія, якому не вдалося потрапити в серійне виробництво, рис. 9. Водоплавна машина була виготовлена на базі агрегатів УАЗ-469. Оригінальну конструкцію мав водозмінюючий кузов і герметичні двері. Перед заднім мостом були встановлені два гребних гвинта, а функцію рулів виконували передні колеса. До 1989 року було виготовлено 14 радянських «Ягуарів» і автомобіль був прийнятий на озброєння. У ході випробувань УАЗ-3907 проплив по Волзі від Ульяновська до Астрахані і назад. Але в 1991 році було прийнято рішення про припинення підготовки до серійного виробництва УАЗ-3907 [6].



Рис. 9. Автомобіль УАЗ-3907 «Ягуар»



Рис. 10. Автомобіль «Амфікара»

Автомобіль "гідрокар" (Hydrocar) розроблений Ріком Доббертіном. Виглядає дуже радикально, рис.11. Поки це тільки модель, але сам розробник вже прославився на весь світ, завдяки беспрецедентній подорожі. Яку він здійснив у 1993 р на саморобному автомобілі-амфібії, що був виготовлений, з молочної цистерни. Маса машини становила більше 8 т, висота більше 3 м і довжина близько 10 м. Найкраща швидкість на землі була 120,5 км/год, а на воді 7-8 вузлів (близько 12 км/год).



Рис. 11. Автомобіль Hydrocar



Рис. 12. Автомобіль «Aquada»

За три роки дослідник подорожував у цьому автомобілі по 38 штатах Америки, а також інших країнах: Гренаді, Венесуелі, в Гондурасі, Нікарагуа, Канаді, Мексиці та інших. Пройшовши шлях понад 53 тис.км по дорогах і близько 5000 км по воді.

На рис. 12 - свіжа розробка автомобіля амфібії "Aquada", розроблена компанією Gibbs Technologies. Бензиновий двигун потужністю 175 к.с. дозволяє розігнатися до 169 км/год на суші і до 48,2 км/год по воді - цього достатньо, щоб буксирувати людину на водних лижах.



Рис. 13. Автомобіль "Ватеркар"



Рис. 14. Автомобіль Rinspeed Splash

Автомобіль "Watercar"-2004, рис. 13, розроблений однойменною американською компанією, являє собою автомобіль амфібію на основі спортивного автомобіля. "Watercar" обійшов "АКВАДОМ", який до того мав рекорд швидкості серед автомобілів амфібій. Показники швидкості досить значні: на суші - 200 км/год, а на воді - 72 км/год (що непогано навіть за мірками швидких катерів).

Останнім рекордсменом серед автомобілів амфібій є "Сплеск", швейцарської компанії Rinspeed, рис. 14. У 2006 р. голова компанії перетнув на ньому Ла-Манш, потрапивши в книгу рекордів Гіннеса. Система на підводних крилах піднімає автомобіль більш ніж на півметра над поверхнею води. Модель може розвивати швидкість до 200 км/год, рухаючись по дорозі. При цьому час розгону становить до 100 км/год і складає 6 секунд. Сплеск здатний розвивати швидкість 45 вузлів (84 км/год) [7].

Американська компанія WaterCar запустила у виробництво і свою першу розробку – автомобіль амфібію Panther. Її розробники називають найшвидшою машиною подібного типу в світі, рис. 15. Під час пересування по землі новинка може розганятися до 128 км/год, а по воді - до 70 км/год.



Рис. 15. Автомобіль «Panther»

Кузов автомобіля амфібії виготовлений зі скловолокна, яке заповнене трьома квадратними метрами пінополістиролу. У цій моделі використане легке металеве шасі, світлотехніка від позашляховика Jeep Wrangler, складаючі колеса (на перетворення з дорожньої машини в човен затрачається близько 15 секунд) і шестициліндровий бензиновий двигун Honda. Maca WaterCar Panther досягає 1360 кг. Під час руху по воді машина може розвинути швидкість до 70,5 км/год.

Віддача мотора новинки становить 250 к.с. На землі він приводить у рух задні колеса, а у режимі човна - водометний рушій, розроблений WaterCar самостійно. Трансмісія - чотириступінчаста механічна, об'єднана з роздавальною коробкою [8].

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку.

Отже, в основному машини амфібії розробляють на замовлення військових. Для цивільного машинобудування це не характерні конструкторські рішення. Найчастіше цим займаються фірми-ентузіасты, хоча іноді на ринок і надходять невеликі партії експериментальних машин, наприклад: ГАЗ-46, ЗІС-485, УАЗ-3907, ВАЗ-2122, Amphicar і інші.

Література

1. <http://ru.wikipedia.org/wiki/Машина-амфибия>
2. <http://ru.wikipedia.org/wiki/DUKW-353>
3. <http://truck-auto.info/gmc/566-duk-w-353.html>
4. <http://greenbelarus.info/articles/22-01-2015/tayskaya-kompaniya-spira-zapustila-v-seriynoe-proizvodstvo-gibridnyy-avtomobil>
5. <http://rosvezdehod.ru/amfibii-avtomobili.htm>
6. <http://rosvezdehod.ru/amfibii-avtomobili.htm>
7. <http://rosvezdehod.ru/amfibii-avtomobili.htm>
8. <http://motor.ru/news/2013/06/28/amphibious/>

УДК 377.015.62

Бондаренко Т.С., к.п.н., ст.викл.

Українська інженерно-педагогічної академія

СИСТЕМА ХМАРНОГО МОНІТОРИНГУ ЯКОСТІ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ КВАЛІФІКОВАНИХ РОБІТНИКІВ

Бондаренко Т.С. Система хмарного моніторингу якості професійної підготовки майбутніх кваліфікованих робітників. Розглянуто технології проведення моніторингових досліджень якості професійної підготовки майбутніх кваліфікованих робітників, які базуються на використанні хмарних сервісів пошукової системи Google. Дано визначення системи хмарного моніторингу, описана її структура, яка складається з шести підсистем, для кожної з підсистем і системи в цілому запропоновано використання окремого хмарного сервісу.

Ключові слова: якість професійної підготовки, моніторинг, хмарні технології, хмарні сервіси, форма, анкета, опитувальник, автоматизація збору та обробки результатів моніторингу.

Бондаренко Т.С. Система облачного мониторинга качества профессиональной подготовки будущих квалифицированных рабочих. Рассмотрены технологии проведения мониторинговых исследований качества профессиональной подготовки будущих квалифицированных рабочих, основанные на использовании облачных сервисов поисковой системы Google. Дано определение системы облачного мониторинга, описана ее структура, состоящая из шести подсистем, для каждой из подсистем и системы в целом предложено использование отдельного облачного сервиса.

Ключевые слова: качество профессиональной подготовки, мониторинг, облачные технологии, облачные сервисы, форма, анкета, опросник, автоматизация сбора и обработки результатов мониторинга.

Bondarenko T.S. The system of cloud monitoring of the quality of professional training of future skilled workers. The technology of monitoring research quality of the training of future skilled workers based on the use of cloud services of search engine Google are described. The article presents the definition of cloud monitoring, described its structure, consisting of six subsystems, for each of the subsystems and the overall system proposed the use of a single cloud service.

Key words: quality of training, monitoring, cloud technologies, cloud services, shape, form, questionnaire, automate the collection and processing of monitoring results.

Постановка і актуальність проблеми. "Кадри вирішують все!" У цій крилатій фразі як не можна більш влучно відображена одвічна актуальність проблеми професійної підготовки. У кожен конкретний період розвитку суспільства ця проблема мала свої характерні особливості так як і поняття якості професійної підготовки в його класичному розумінні - сукупність властивостей продукції, що зумовлюють її придатність задовольняти певні потреби у відповідності до її призначення.

Сьогодні в умовах жорсткої конкуренції, насиченості ринку працевлаштування професійними кадрами, постійного оновлення та вдосконалення виробничих процесів ця проблема стає особливо актуальною. Професійна освіта в сучасних умовах розглядається в якості вирішального чинника підвищення ефективності економічних перетворень. Одним із шляхів вирішення даної проблеми у відповідності з основними принципами Болонського процесу є система забезпечення якості освітніх послуг на основі його моніторингу.

Питання моніторингу в освіті розроблялися В. Аванесовим, Т. Волобуєвою, Г. Єльніковою, О. Касьяною, І. Лапиною, О. Локшиною, В. Приходьком, З. Рябовою та ін.

Проблему моніторингу якості професійної підготовки розглядали Андрєєв В.І., Бенькович Н.В., Байдацька Н.М., Хриков Є. та ін.

Постановка задачі. Тут варто більш детально зупинитися на визначенні основної задачі моніторингу якості професійної підготовки. Відповідно до визначення [6] моніторинг - безперервний процес спостереження та реєстрації параметрів об'єкту в порівнянні із заданими критеріями. Разом з тим в деяких дослідженнях [1; 4] акцент у вирішенні завдання моніторингу зміщений у бік оцінки стану об'єкту з метою прийняття рішень із його управління. Такий підхід може забезпечити частинний розв'язок задачі на певному етапі. Але на наш погляд загальне завдання моніторингу - сприяти стабільному функціонуванню об'єкту на тривалому проміжку часу в умовах зовнішніх та внутрішніх збурень. Виходячи з цього завдання моніторингу якості професійної підготовки має вирішуватися не як окреме завдання підвищення якості підготовки на певному проміжку часу, а як завдання постійного забезпечення незмінно високого рівня цієї

підготовки при зміні вимог до якості професійної підготовки і умов, в яких дана підготовка здійснюється.

Природно, що при цьому завдання моніторингу не зводиться до оцінки рівня індивідуальних досягнень учнів, процедур ліцензування, атестації та акредитації. Це завдання має вирішуватися системно і в комплексі з іншими заходами щодо підвищення якості професійної підготовки. При цьому слід враховувати, що моніторинг - це один із шляхів підвищення ефективності та якості освітнього процесу. Система моніторингу повинна надавати допомогу у вирішенні даного завдання, а не створювати проблеми, які пов'язані зі збором обробкою і аналізом інформації про об'єкт моніторингу.

Організація системи моніторингу та технології, що визначають процес її функціонування, безпосередньо впливають на її ефективність. У зв'язку з цим необхідно розробити структуру системи і вибрати технології вирішення завдань моніторингу, які з одного боку мінімізують фінансові та часові затрати, а з іншого боку забезпечать зручність експлуатації системи і якість її функціонування.

Виклад основного матеріалу. Моніторингове дослідження якості професійної підготовки майбутніх кваліфікованих робітників є багатовимірним та розподіленим у просторі і часі. Для його проведення необхідно мати інструментарій, який дасть змогу перекрити значний простір та забезпечить контрольованість процесу. На сьогодні таким вимогам у повній мірі відповідають хмарні технології. Сама ідея хмарних технологій (доступ до хмарного ресурсу у будь-якому місці і у будь-який час) як можна краще перекриває потреби моніторингу якості професійної підготовки, бо значна частина роботи зі збору інформації (анкетування, опитування, діагностика, контроль), її зберігання, оброблюванню та надання суб'єктам моніторингу може бути ефективно виконана з використанням хмарних сервісів.

Система хмарного моніторингу якості професійної підготовки майбутніх кваліфікованих робітників складається з наступних підсистем: налаштування параметрів моніторингу; організація збору інформації; отримання інформації про об'єкти моніторингу; обробка інформації засобами системи моніторингу; подання результатів моніторингу користувачу; менеджмент системи моніторингу.

На відміну від складу системи моніторингу, який наведено в роботі [3, 260] в даному випадку в систему додано два блоки: підсистема налаштування параметрів моніторингу та підсистема менеджменту.

Підсистема налаштування параметрів моніторингу є надзвичайно важливою для ефективної роботи всієї системи. Якщо немає чіткого визначення цілі моніторингу, якщо не обрані параметри за якими буде вестись спостереження та не задані критерії їх оцінювання, то всі подальші роботи з моніторингу, як би вони бездоганно не були виконані, не дадуть очікуваного ефекту. Треба зауважити, що деякі дослідники розглядають цей етап як підготовчий (наприклад, [2,91; 5, 13]). Проте ми вважаємо, що корегування заданих на цьому етапі параметрів необхідно проводити регулярно.

Ще одна складова системи моніторингу, без якої її ефективна робота буде неможлива – підсистема менеджменту. Враховуючи те, що моніторинг це *постійне* спостереження за станом об'єкту, наявність менеджменту системи моніторингу необхідна для планування і координації всіх робіт, пов'язаних з моніторингом.

Зупинимось більш детально на визначенні системи хмарного моніторингу (СХМ). Всі підсистеми даної системи, за виключенням підсистеми налаштування параметрів моніторингу, побудовані на основі використання хмарних сервісів. Поняття хмарний моніторинг можна визначити як специфічну технологію побудови систем моніторингу на основі використання хмарних сервісів. Системи моніторингу, які створені з використанням хмарних сервісів, мають певні особливості в структурі і функціонуванні. В такому випадку дані зберігаються на віддаленому мережевому ресурсі, доступ до якого може здійснюватися з будь-якого комп'ютера, підключеного до Інтернету. При цьому зберігається можливість авторизації доступу та контролю за процесом редагування або перегляду даних. Це дає можливість керівнику мережевої системи моніторингу розподіляти права доступу до окремих ресурсів системи та здійснювати контроль за діяльністю користувачів системи. Розглянемо використання хмарних сервісів для кожної з підсистем.

Підсистема організації збору інформації. Основна частина інформації в даній системі збирається у вигляді анкет, опитувальників, результатів тестування та співбесіди. Для вирішення завдань розсилки, заповнення та збору заповнених документів у підсистемі використовується форма сервісу Docs.Google. Сама форма являє собою серію питань з різнотипними відповідями. Її побудова включає введення питання, вибір і визначення належного типу відповіді, збереження заповненої форми в базі результатів в хмарному сховищі даних Google Диск.

На формі можна розмішувати до восьми типів різних питань (один із списку, декілька із списку та ін.). Практика використання Google-форм показала, що цей набір типів запитань є достатнім для створення різноманітних анкет, опитувальників та тестів навчальних досягнень для моніторингових досліджень якості професійної підготовки.

Для відправки форми респонденту використовується опція **Відправити цю форму електронною поштою**. У додаткових налаштуваннях можна задати пункт **Вбудувати**, де надається код для вбудовування форми в сайт СХМ. Опція **Змінити підтвердження** дозволяє ввести повідомлення, яке видається користувачеві при натисканні кнопки **Відправити**. Публікація **Зведення відповідей** дасть респонденту можливість побачити основні результати на поточний момент часу.

Підсистема отримання інформації про об'єкти моніторингу. Вище було сказано про збереження результатів заповнення форм в базі результатів в хмарному сховищі даних Google Диск. Дані моніторингу зберігаються і обробляються в так званій хмарі, яка являє собою, з точки зору користувача, один великий віртуальний сервер. Істотною перевагою хмарних сховищ для створення СХМ є можливість організації спільного доступу користувачів до файлів, які там зберігаються.

Підсистема обробки інформації засобами системи моніторингу. Дані після відправки форми потрапляють в електронну таблицю Google Таблиця. Таблиця з'являється в менеджері файлів Docs.Google; її ім'я береться від назви форми плюс слово 'response'. У таблицю додається колонка **Відмітка часу кожного запису**, що є також «індексом» набору даних. За замовчуванням, інформація також сортується по цьому полю, так що ми отримуємо хронологічний список всіх записів. При обробці даних електронна таблиця дозволяє легко пересортувати інформацію за будь-яким стовпцем, не розбиваючи самі записи.

Використовуючи інструмент **Показати зведення відповідей** ми отримаємо вкладку з діаграмою з кожного питання форми. Крім того, поруч з діаграмою результати будуть представлені у вигляді чисел і відсотків. В таблиці є набір функцій (також як і в таблиці Excel) для статистичної обробки даних. При бажанні таблиця може бути експортована для обробки в додаток MS Excel.

Підсистема подання результатів моніторингу користувачу. Для розсилки респондентам результатів моніторингу використовуються режими спільного доступу користувачів до файлів, що зберігаються на Google Диску.

Підсистема менеджменту системи моніторингу. Підсистема менеджменту в СХМ організована з використанням хмарного сервісу Google Календар. З його допомогою складається розклад робіт, пов'язаних з вирішенням задач СХМ. Використовуючи календар можна відправляти нагадування про події через SMS та по e-mail (у цьому випадку до нагадування може бути додано посилання на форму, яку необхідно заповнити).

Для об'єднання всіх розглянутих підсистем в єдину систему використовується хмарний сервіс Google Сайти. З його допомогою можна організовувати тематичні чати (обговорення результатів моніторингу та пов'язаних з цим проблем); виставляти анкети, опитувальники, тести для збору інформації; організувати спільну роботу зі збору та обробки даних; вести облік заповнення інформаційної бази СХМ; проводити он-лайн опитування та он-лайн анкетування; призначати час для он-лайн обговорення результатів моніторингу.

Висновок. Запропоновані технології проведення моніторингових досліджень на базі хмарних сервісів мають цілий ряд переваг серед яких автоматизовані процеси створення та розсилки анкет і опитувальників, автоматизований збір та обробка результатів дослідження, управління подіями моніторингу за допомогою хмарного сервісу Google Календар, створення для проведення моніторингового дослідження Google-сайту та розміщення на ньому всіх необхідних матеріалів з теми дослідження (інструкцій, пояснень, анкет, опитувальників тощо).

Перспективи подальших досліджень. На даний час розроблено технічне завдання на СХМ якості професійної підготовки майбутніх кваліфікованих робітників Люботинського професійного ліцею залізничного транспорту та закінчується робота над технічним проектом системи. В перспективі планується завершити робочий проект системи і перейти до її дослідної експлуатації.

1. Бенькович Н.В. Система мониторинга качества профессиональной подготовки будущих экономистов в высшем учебном заведении [Електронний ресурс]. - Режим доступу: <http://www.tisbi.ru/assets/Site/Science/Documents/23-BENKOVICH.pdf>
2. Мариновська О. Моніторинговий супровід інноваційної діяльності / Оксана Мариновська // Вісник. Тестування і моніторинг в освіті – 2012. – № 7–8. – С. 89–96.
3. Моніторинг та аудит інформаційних технологій у системі управління організації / В. К. Галіцин, Р. Л. Ус // Моделювання та інформаційні системи в економіці. - 2012. - Вип. 86. - С. 256-262.
4. Коробович Л.П. Забезпечення педагогічних умов успішного моніторингу результативності навчального процесу у ВНЗ // НАРОДНА ОСВІТА. Електронне наукове фахове видання. - 2012. - № 1(16). - Режим доступу: <http://archive.nbu.gov.ua/e-journals/NarOsv/2012-16/index.htm>
5. Приходько В.М. Парадигми моніторингу якості освіти і педагогічного процесу: (У запитаннях і відповідях). Науково-методичний посібник. – Запоріжжя, 2010. – 215 с.
6. Словарь бизнес-терминов [Електронний ресурс]. - Режим доступу: <http://dic.academic.ru/dic.nsf/business/7942>

УДК 378:37.01/.09:004+338.48:008

Зубехіна Т. В.

ПВНЗ «Академія рекреаційних технологій і права»

АНАЛІЗ РЕЗУЛЬТАТІВ ПЕДАГОГІЧНОГО ЕКСПЕРИМЕНТУ ЩОДО ФОРМУВАННЯ ІНФОРМАЦІЙНОЇ КУЛЬТУРИ БАКАЛАВРІВ З ТУРИЗМУ

Зубехіна Т. В. Аналіз результатів педагогічного експерименту щодо формування інформаційної культури бакалаврів з туризму. У статті представлено результати педагогічного дослідження процесу формування інформаційної культури бакалаврів з туризму у процесі фахової підготовки; охарактеризовано структурні компоненти інформаційної культури бакалаврів з туризму, критерії та показники їх сформованості; визначено діагностичний інструментарій, з допомогою якого здійснено порівняльний аналіз сформованості інформаційної культури бакалаврів з туризму у контрольній та експериментальній групах.

Ключові слова: структурні компоненти, інформаційна культура, бакалавр туризму, діагностичний інструментарій, педагогічний експеримент.

Зубехина Т. В. Анализ результатов педагогического эксперимента по формированию информационной культуры бакалавров туризма. В статье представлены результаты педагогического исследования процесса формирования информационной культуры бакалавров туризма в процессе профессиональной подготовки; охарактеризованы структурные компоненты информационной культуры бакалавров туризма, критерии и показатели их сформированности, определено диагностический инструментарий, с помощью которого осуществлен сравнительный анализ результатов сформированности информационной культуры в контрольной и экспериментальной группах.

Ключевые слова: структурные компоненты, информационная культура, бакалавр туризма, диагностический инструментарий, педагогический эксперимент.

Tetiana V. Zubekhina. Analysis of pedagogical experiment on formation of information culture Bachelors of Tourism. The article describes the results of pedagogical research process of formation of information culture bachelors of tourism in the process of professional training; described structural components of information culture Bachelor of Tourism, criteria and indicators of their formation; the diagnostic tools with which comparative analysis of formation of information culture of Bachelor of Tourism in the control and experimental groups.

Keywords: structural components, information culture, bachelor of tourism, the diagnostic tools, pedagogical experiment.

Постановка проблеми у загальному вигляді. Інформатизація суспільства та комп'ютеризація усіх сфер людської діяльності стали провідними тенденціями світового розвитку. Нині значна частина населення світу задіяна в процесі виробництва та споживання інформації. Інформаційні технології все більше проникають в різні сфери життя, науки, освіти, виробництва, що вимагає відповідних знань та вмінь їх використання. В зв'язку з цим, важливе місце займають проблеми, пов'язані з оптимізацією навчального процесу у вищих навчальних закладах, до випускників яких пред'являються нові вимоги. Вони, насамперед, пов'язані з умінням фахівців з туризму орієнтуватися у величезній кількості інформації, впроваджувати нові інформаційні технології на практиці, з максимальною користю використовувати відомості з різних інформаційних джерел.

Аналіз останніх досліджень та публікацій. У сучасних психолого-педагогічних дослідженнях приділяється увага проблемі формування інформаційної культури особистості. Питання визначення суті поняття культура та її місця в освіті вивчали В. П. Андрущенко, А. О. Веряєв, Л. В. Гуперський, Н. Б. Крилова, В. М. Розін, І. К. Шалаєв та ін. Сучасні дослідження Л. Г. Агафоновой, С. І. Байлик М. П. Мальської, С. В. Мельниченко, О. П. Олійник, В. В. Худо спрямовані на дослідження теоретичного аспекту проблеми без урахування особливостей професійної діяльності у сфері туризму і специфіки професійної підготовки бакалаврів з туризму.

Формулювання цілей статті. Мета статті – здійснити експериментальний аналіз результатів педагогічного дослідження щодо формування інформаційної культури бакалаврів з туризму.

Виклад основного матеріалу дослідження. В інформаційному суспільстві важливого значення набувають не лише міцні фундаментальні знання фахівців з туризму, але і їхня здатність оперативно реагувати на запити мінливої сфери туризму, постійно накопичувати професійні знання і ефективно використовувати різні джерела інформації для вирішення професійних завдань. Зростає потреба у висококваліфікованих фахівцях, які володіють інформаційною культурою, тому оволодіння знаннями та вміннями використання інформаційних технологій стало важливою складовою професійної підготовки бакалаврів з туризму. Отже, потрібно формувати в них інформаційну культуру, яка є відображенням рівнів організації інформаційних процесів, сформованості вмінь

створювати, зберігати та використовувати на практиці отриману інформацію. *Інформаційна культура бакалаврів з туризму* розглядається нами як інформаційний компонент професійної культури особи, котра здобуває теоретичні знання та практичні уміння і навички, достатні для успішного виконання професійних обов'язків, які передбачають спроможність вільно орієнтуватися в інформаційному просторі, брати участь у його формуванні та сприяти інформаційним відносинам під час розробки туристичного продукту й координації туристичних поїздок клієнтів.

Структурними компонентами інформаційної культури бакалаврів з туризму є: мотиваційний, когнітивний, діяльнісний і рефлексивно-аналітичний. Відповідно до структурних компонентів визначено критерії і показники рівня сформованості інформаційної культури.

Мотиваційний компонент визначається мотиваційно-ціннісним критерієм, когнітивний – гносеологічним, діяльнісний – операційним, рефлексивно-аналітичний – особистісним.

О. Значенко стверджує, що кожен структурний компонент інформаційної культури може бути описаний через технічний, операційний, програмний, гігієнічно-ергономічний, навчальний та методичний критеріальні показники [1].

Для проведення педагогічного експерименту була визначена репрезентативна вибірка зі студентів Тернопільського національного педагогічного університету імені Володимира Гнатюка та Луцького інституту розвитку людини Університету «Україна». Експериментальну (72 осіб) і контрольну (74 особи) групи склали студенти 3-4 курсів напряму підготовки 6.140103 «Туризм».

Студенти контрольної групи навчалися за стандартною методикою викладання дисциплін, процес навчання студентів експериментальної групи супроводжувався створенням у вищому навчальному закладі інформаційного простору навчально-виховного процесу.

До основного засобу організації у ВНЗ інформаційного простору ми відносимо систему управління навчанням Moodle, яка включає такі інтернет-ресурси, як: аудіо-відео матеріали, інтернет-сайти, віртуальні співтовариства та комп'ютерне тестування.

Сучасна система професійної підготовки вимагає добору таких засобів навчання, які були б орієнтовані на розвиток творчих здібностей студентів, індивідуалізацію навчального процесу, формування навичок самостійного оволодіння і застосування знань. Для цього використовуються елементи дистанційного навчання, перевагами якого є: економія часу викладача і студента; дозволяє студенту відчувати самостійність і відповідальність, що підвищує його мотивацію до навчання; розвиває у студентів навички до безперервної освіти і підвищення кваліфікації у майбутній професійній діяльності [2].

Першим і найбільш поширеним засобом організації у ВНЗ інформаційного простору навчально-виховного процесу є Інтернет-ресурси. Структура Інтернету складається з сервісів та служб, останні в свою чергу поділяються на комунікативні та інформаційні. Інформаційні служби досить популярні і сучасній системі вищої освіти, оскільки вони насичені навчальними матеріалами і легкі у використанні. На сьогоднішній день і студенти, і викладачі повинні вміти добре орієнтуватися у величезній кількості мультимедійних підручників, інтернет-ресурсів, програмних засобів для турфірм, готелів, ресторанів, які забезпечують формування інформаційного простору.

Для підвищення рівня сформованості інформаційної культури бакалаврів з туризму ми створили електронні навчально-методичні комплекси таких навчальних дисциплін «Основи інформаційної культури бакалаврів з туризму», «Організація сфери екскурсійних послуг», «Організація туристичних подорожей» на базі системи Moodle, яка дає змогу ефективно обмінюватися навчальним контентом, взаємодіяти он-лайн та використовувати сучасні Інтернет-технології.

За допомогою цієї платформи забезпечується безперервний навчальний процес, тобто є можливість організовувати навчання під час вимушених канікул у весняному семестрі, при проходженні студентами практики за кордоном, уможливорює самостійну роботу студентів влітку. Студентам заочної форми навчання є можливість готуватися до екзаменаційних сесій протягом року в зручний для них час, не відриваючись від роботи.

В систему Moodle були завантажені мультимедійні матеріали, що містять відео- та аудіо фрагменти. Аудіо-відео матеріали – це робочі матеріали, які стануть у нагоді при вивченні тем «Об'єкти екскурсії «Старе місто», «Найвизначніші природні пам'ятки України» з дисципліни «Організація сфери екскурсійних послуг» та «Топ-10 найоригінальніших архітектурних споруд світу», «Найпопулярніші курорти України та світу» з дисципліни «Організація туристичних

подорожей». За допомогою аудіо-відео матеріалів студенти зможуть прослухати, побачити, а потім і опрацювати інформацію та краще підготуватись до певного завдання.

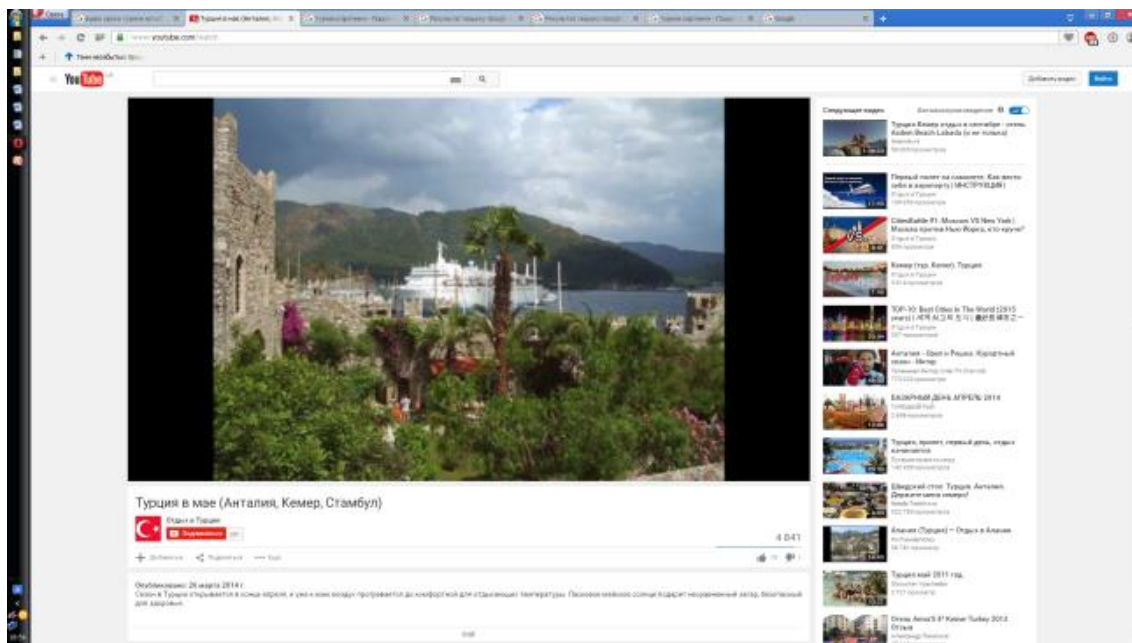


Рис. 1. Зразки відео лекцій.

Веб-сайти активно використовуються постачальниками послуг та посередниками туристського ринку при взаємодії між собою. Варто згадати про широкі можливості бронювання готельних номерів та квитків на літак через Інтернет. При вивченні навчальних дисциплін «Організація туристичних подорожей», «Організація сфери екскурсійних послуг» та «Основи інформаційної культури бакалаврів з туризму» важливо навчити студентів користуватися інтернет-ресурсами та системами, тому що вони є невід'ємною складовою туристичного бізнесу. Крім наукового значення, сайти є ще й пізнавальними, тому що саме на їхніх сторінках можна знайти інформацію про те, що не написано в путівниках, що може допомогти майбутнім туристам краще зорієнтуватися в незнайомому місці: правильно вибрати готель і маршрут поїздки, щоб не витратити зайвих грошей і не пропустити цікавих пам'яток.

Використовуючи електронний навчально-методичний комплекс навчальної дисципліни, студенти можуть знайти завдання для індивідуальних та семінарських занять, самостійних та контрольних робіт, тематику рефератів. При цьому вони повинні завантажити результати виконання знову в систему на перевірку в межах вказаних термінів. В системі передбачена можливість зручного оформлення питань для самоконтролю, проведення повноцінного проміжного тестування студентів.

Для активізації роботи між студентом та викладачем використовується форум – основний інструмент взаємозв'язку, що дає змогу студентам діставати консультації, ставити питання, на які викладач може відповідати у зручний для нього час. Запитання-відповіді в текстовому форматі на бажання викладача можуть бути доступними й для інших студентів (навіть наступних груп). Це дає можливість створити перелік питань, які часто виникають у студентів стосовно вивчення курсу, з вичерпними відповідями на них.

Нині найпоширенішим засобом організації інформаційного простору у ВНЗ є віртуальні співтовариства – це сукупність зв'язків між студентами та викладачами, що мають щось загальне (спільні цілі, інтереси, погляди, зайняття, хобі, проблеми і так далі) і, що здійснюють взаємодію у віртуальному просторі. Віртуальні мережеві співтовариства належать до категорії саморегульованих соціальних структур, що саморозвиваються. Загальний принцип, що визначає характер розвитку соціальної системи кіберпростору, полягає в її прагненні забезпечити свою цілісність і підвищити ефективність комунікаційних процесів.

Для визначення ефективності формування інформаційної культури бакалаврів з туризму визначено критерії, показники та рівні (репродуктивний, конструктивний, продуктивний).

Мотиваційно-ціннісний критерій сформованості інформаційної культури бакалаврів з туризму у процесі фахової підготовки характеризувався наявністю позитивної мотивації та ціннісних орієнтацій у студентів до навчання за допомогою інформаційних технологій, що лежать в основі формування інформаційної культури бакалаврів з туризму. Для оцінки сформованості мотивації у бакалаврів з туризму використовувався тест-опитувальник «Методика діагностики особистості на мотивацію до успіху Т. Елерса»).

Показниками гносеологічного критерію оволодіння бакалаврами з туризму професійними знаннями про можливості використання Інтернет технологій у майбутній діяльності були повнота, системність і міцність знань студентів, які визначалися за результатами відповідей студентів на тестові завдання з використання інформаційних технологій в туризмі.

Операційний критерій сформованості умінь та навичок студента щодо використання електронних засобів та інформаційних технологій в процесі формування інформаційної культури характеризувався сформованістю умінь та навичок студента щодо використання електронних засобів та інформаційних технологій в процесі формування інформаційної культури.

Показником особистісного критерію є здатність студентів до осмислення, саморозвитку, усвідомленого й творчого ставлення до можливостей інформаційних технологій у формуванні інформаційної культури. Для визначення рівня рефлексії використовувався «Тест на виявлення рівня рефлексії» Верігіної Н. А. Для визначення рівня сформованості здібностей до саморозвитку і самоосвіти використовувався однойменний тест.

Продуктивний рівень сформованості інформаційної культури бакалаврів з туризму характеризується високими професійними знаннями, професійними знаннями про можливості використання Інтернет технологій у майбутній діяльності. Конструктивний – достатніми професійними знаннями, професійними знаннями про можливості використання Інтернет технологій у майбутній діяльності; рівень характеризується поверхневими професійними знаннями про можливості використання Інтернет технологій у майбутній діяльності.

Педагогічний експеримент щодо формування інформаційної культури бакалаврів з туризму у процесі фахової підготовки включав констатувальний та формувальний етапи.

Отримані у результаті експерименту дані свідчать про те, що на початок педагогічного експерименту, в експериментальній групі продуктивний рівень сформованості за всіма компонентами формування інформаційної культури бакалаврів з туризму у процесі фахової підготовки становив 11,11% студентів, наприкінці – відповідно 25,56% студентів, конструктивний рівень становив на початок – 28,06% студентів, наприкінці – 40,83% студентів, репродуктивний рівень становив на початок – 60,83% студентів, наприкінці – 33,61% студентів (табл. 1).

У контрольній групі на початок педагогічного експерименту продуктивний рівень сформованості за всіма компонентами формування інформаційної культури бакалаврів з туризму у процесі фахової підготовки становив 10,54% студентів, наприкінці – 15,95% студентів, конструктивний рівень становив на початок – 28,11% студентів, наприкінці – 35,95% студентів, репродуктивний рівень становив на початок – 61,35% студентів, наприкінці – 48,11% студентів

Отже, наприкінці педагогічного експерименту у експериментальній групі збільшилася кількість студентів з продуктивним (14,45%) та конструктивним (12,77%) рівнями, значно зменшилася з репродуктивним (27,28%) рівнем. Динаміка змін у контрольній групі наступна: збільшення кількості студентів, які мають продуктивний і конструктивний рівні (відповідно 5,41% студентів і 7,84% студентів), зменшення репродуктивного рівня – на 13,24% студентів.

Таблиця 1

Динаміка рівнів сформованості за всіма компонентами формування інформаційної культури бакалаврів з туризму у процесі фахової підготовки

№ п/п	Рівень	Кількість студентів експериментальних груп, n=72	Кількість студентів контрольних груп, n=74
		Педагогічний експеримент	

		початок		кінець		у % різниця	початок		кінець		у % різниця
		в абсолютних цифрах	у % від загальної кількості	в абсолютних цифрах	у % від загальної кількості		в абсолютних цифрах	у % від загальної кількості	в абсолютних цифрах	у % від загальної кількості	
1	Продуктивний	11	11,11	25	25,56	14,45	11	10,54	16	15,95	5,41
2	Конструктивний	28	28,06	41	40,83	12,77	28	28,11	36	35,95	7,84
3	Репродуктивний	61	60,83	34	33,61	27,28	61	61,35	48	48,11	13,24

Висновки та перспективи подальших досліджень. Отже, основним засобом організації у вищому навчальному закладі інформаційного простору є система управління навчанням Moodle, яка включає такі інтернет-ресурси, як: аудіо-відео матеріали, інтернет-сайти, віртуальні співтовариства та комп'ютерне тестування. Створення інформаційного простору навчально-виховного процесу підготовки бакалаврів з туризму експериментальної групи дозволило ефективно сформувати у них інформаційну культуру., що підтвердили результати педагогічного експерименту. Зокрема, вони свідчать про суттєві позитивні зміни рівнів сформованості інформаційної культури бакалаврів з туризму у процесі фахової підготовки у студентів експериментальної групи порівняно зі студентами контрольної групи.

Перспективою подальших досліджень вбачаємо у здійсненні аналізу європейського досвіду формування інформаційної культури фахівців з туризму у вищих навчальних закладах.

1. Значенко О. П. Формування інформаційної культури майбутніх учителів гуманітарних дисциплін: автореф. дис. канд. пед. наук: спец. 13.00.04 / О. П. Значенко. – Київ : бв., 2005. – 24 с.
2. Кільченко А. В. Використання електронних інформаційних систем відкритого доступу для планування наукових досліджень в галузі освіти / А. В. Кільченко // Інформаційні технології і засоби навчання. – 2015. – №5 (49). – Режим доступу до журналу: <http://journal.iitta.gov.ua>.

УДК 378.004

Мельник В.М., Бортник К.Я., Чеб С.С., Міскевич О.І.

Луцький національний технічний університет

ДЕЯКІ ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ КОМУНІКАЦІЙНИХ ІНТЕРНЕТ-ТЕХНОЛОГІЙ В СИСТЕМІ ОСВІТИ УКРАЇНИ

Мельник В.М., Бортник К.Я., Чеб С.С., Міскевич О.І. Деякі перспективи впровадження комунікаційних інтернет-технологій в системі освіти України. Розглядаються деякі важливі перспективи розвитку комунікаційних технологій Інтернет для навчального процесу та дистанційного навчання в системі освіти. В зв'язку з цим наводяться деякі важливі аспекти раціонального їх використання та проводиться коротка характеристика призначення електронних порталів дистанційної освіти вузів. Висвітлюються деякі важливі проблеми впровадження інформаційно-комунікаційних технологій в систему навчання вітчизняних загальноосвітніх закладів.

Мельник В.Н., Бортник Е.Я., Чеб С.С., Мискевич О.И. Некоторые перспективы внедрения коммуникационных интернет-технологий в системе образования Украины. Рассматриваются некоторые важные перспективы развития коммуникационных технологий Интернет для процесса обучения и дистанционного обучения в системе образования. В связи с этим наводятся некоторые важные аспекты рационального их использования и проводится короткая характеристика назначения электронных порталов дистанционного обучения вузов. Обсуждаются некоторые важные проблемы внедрения информационно-коммуникационных технологий в систему обучения отечественных общеобразовательных заведений.

Melnyk V., Bortnyk Ka., Heb S., Mickiewicz O. Some prospects for implementation of Internet technologies and communication in education Ukraine. Some important perspectives of growth of Internet communication technologies for education process and distant education process in education system are discussed in the article. In addition with this some important aspects of their rational using are pointed in it and a short characteristic of electron portals appoint for distant education in university collages is presented. There also some important problems of information and communication technologies introducing in education system of country common education schools are considered.

Освітня й виховна діяльність в навчальному закладі повинна знаходитись у тісному взаємозв'язку, доповнюючи і збагачуючи одна одну. Навчання і виховання – це складові одного цілого процесу, процесу розвитку особистості. В зв'язку з стрімким плином науково-технічного прогресу та невідпинної модернізації у всіх сферах діяльності людини важливого значення набуває впровадження сучасних комунікаційних технологій. Досить важливу роль в навчально-виховному процесі відіграють комунікаційні Інтернет-технології, що покликані швидкодіюно вирішувати самі різноманітні його аспекти. Системи дистанційного навчання, навчальні Web-конференції, мережеві семінари та інші сучасні засоби електронно-комунікаційної передачі знань витісняють звичайні засоби та комунікативні можливості передачі знань в багатьох сторонах організації навчального процесу, оскільки являються набагато доступнішими, зручнішими і легшими у використанні.

Стрімкий розвиток мережевих інформаційних технологій поряд з помітним зниженням тимчасових і просторових бар'єрів в розповсюдженні інформації відкрив нові перспективні можливості у сфері освіти. Можна з упевненістю стверджувати, що в сучасному світі має прояв тенденція злиття освітніх і інформаційних технологій та формування на цій основі принципово нових – інтегрованих технологій навчання, заснованих, зокрема, на технологіях комунікативної передачі інформації через глобальну мережу Інтернет. Зокрема, зручність і гнучкість гіпертекстового представлення матеріалу, оперативний доступ до інформації, розташованої в різних регіонах України і країнах зарубіжжя, висока оперативність оновлення і інші переваги дозволили достатньо швидко впровадити їх в практичну діяльність багатьох вищих навчальних закладів. Аналіз освітніх ресурсів в цілому показує, що найчастіше такі технології використовуються в учбовому процесі інститутів і університетів гуманітарного напрямку. Пов'язано це з типом та характером представлення навчального матеріалу. В основному – це текстовий матеріал з графічними, зрідка анімованими ілюстраціями.

Існує і інша проблема в технічній освіті. Майбутній інженер-педагог повинен не тільки знати необхідні теоретичні положення дисциплін, що вивчаються, але і мати, наприклад, поняття про типічні процеси, що відбуваються в тих чи інших механізмах, пов'язаних з предметним

вивченням, набуті і перейняти навички вимірювань, обробки та співставлення результатів. То ж одним з серйозних перспективних завдань є розробка технологій віддалених (дистанційних) лабораторних практикумів.

Ще одним найважливішим завданням сучасного етапу розвитку українського дистанційного навчання є інтеграція і уніфікація освітніх ресурсів, технологій і середовищ. За останні роки створено велику кількість досить непоганих комунікативних розробок для їх практичного впровадження в Інтернет, проте вони параметрально не узгоджуються одна з одною, відрізняються операційними системами, способом подачі інформації, а зміст дисциплін досить часто просто не дозволяє використовувати їх в рамках єдиної освітньої програми.

Однак сучасний повноцінний учбовий процес уже практично неможливий без інтерактивних засобів навчання. Аналізуючи сьогоденні освітні Інтернет-технології, можна, на нашу думку, сформулювати три основні аспекти раціонального розвитку їх в системі освіти України:

- перспективи розробки гіпертекстових навчально-методичних помічників з близькоспорідненим (у ідеалі – з єдиним) інтерфейсом користувача і з урахуванням реальних технічних можливостей сьогоденного студента;
- перспективи розробки лабораторних практикумів віддаленого доступу для інженерних спеціальностей;
- перспективи розробки технологічного середовища для системи дистанційної та відкритої освіти, що органічно об'єднує накопичений провідними вузами досвід і типовий інструментарій в користуванні для формування єдиного освітнього середовища України [5].

Поряд з цим, необхідно враховувати виявлення декількох чітко визначених підходів в області Інтернет-освіти [1,4]. Так для одної категорії суб'єктів навчання Інтернет – це організована система самоосвіти, для якої необхідний відповідний вибір засобів навчання, відбір змісту матеріалу і його логічна чи спеціально впорядкована організація представлення. Для іншої ж – систематична освіта, що організовує взаємодію викладача і студента в комунікації навчання. А ще для іншої – це можливість творчого самовираження. Інтернет-навчання може також виступати як допоміжний, переважно інформаційний ресурс. Втім, багато хто схильний до тенденції об'єднувати вищеперечислені підходи.

Досить важливим є також чітке визначення форми навчання, для якої передбачається використовувати Інтернет-технології: для очної чи дистанційної. Якщо для очної форми, то досить важливо визначити, які види інтеграції при цьому передбачаються. Якщо для дистанційної – то яка модель дистанційного навчання буде взята до уваги для його практичної реалізації. Дані питання являються досить важливими. Від їх рішення та дидактичного обґрунтування залежить ефективність використання вже існуючих ресурсних комунікаційних систем та послуг Інтернет в цілому, а також загальна процесуально вигідна організація інформаційно-освітнього простору.

На сьогоденний день більшість українських вузів мають свої власні Інтернет портали дистанційної освіти, де зайшовши на них студенти можуть ознайомитись з лекційним матеріалом навчальної дисципліни, переглянути відеоуроки, опрацювати лабораторні чи практичні роботи, пройти самотестування і інше. Для прикладу розглянемо деякі з вітчизняних освітніх порталів дистанційного навчання вузів України.

Електронний портал дистанційного навчання Луцького національного технічного університету (ЛНТУ), <http://elearning.lutsk.ua/>. [2] (рис. 1) – це електронний портал, побудований на базі системи CMS Moodle, яка є одним із різновидів освітньої CMS системи, що використовується більш ніж в 20-ти вузах України. Портальна система електронного навчання ЛНТУ призначена для багатогранної допомоги студентам в організації як стаціонарного навчального так і самоосвітнього навчальних процесів, для того щоб стати професійними фахівцями з обраними ними спеціальностями.

Інше призначення цієї системи – це допомога викладачам вузу в організації процесу навчання з студентами через Інтернет-мережу.

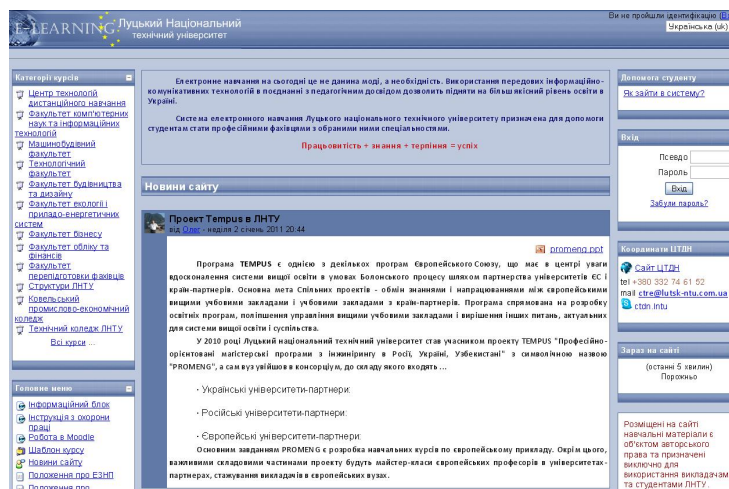


Рис. 1 Електронний портал дистанційного навчання
 Луцького національного технічного університету

До списку сайтів дистанційного навчання, побудованих на системі CMS Moodle відноситься і сайт дистанційного навчання Харківського національного економічного університету (ХНЕУ), <http://www.ikt.hneu.edu.ua/> [3] (рис. 2). Даний сайт являється також основним джерелом здобуття навчального матеріалу згідно обраної спеціальності студентами вузу через Інтернет-комунікацію з ним та дуже вагомою допомогою для контингенту його викладачів.



Рис. 2 Сайт дистанційного навчання
 Харківського національного економічного університету

Поряд з орієнтацією на систему CMS Moodle, інші освітні портали дистанційної освіти України будуються на таких системах, як Joomla, ATutor, що використовується як «движок» для сервера дистанційного навчання Тернопільського національного технічного університету імені Івана Пулюя, <http://dl.tntu.edu.ua>, та Ilias – що використовується у Київському національному університеті імені Тараса Шевченка.

Інтернет технології в системі освіти, можна відмітити, стають з кожним днем все більше необхідною невід'ємною частиною в організації начально-виховного процесу в закладах освіти України, оскільки і сучасне суспільство дедалі інтенсивніше набирає риси інформаційного, що одночасно вимагає нового, більш якісно та багатогранно організованого рівня організації

освітнього процесу через залучення новітніх інноваційних методів. Сам підхід такої організації освіти стимулює потребу залучення людей високої вузькопрофесійної освіченості у вищезгадану систему, де буде також стояти потреба в перекваліфікації працівників в порівнянні з швидким розвитком технологій електронно-дистанційної комунікації. Нинішній вищий навчальний заклад вже в багатьох аспектах не в змозі оперативно змінювати навчальні курси, адаптуючи їх до швидкодіючого наукового розвитку, а поряд з тим і реагувати на зміну запитів споживачів освітніх послуг, проте сучасне виробництво потребує більше освічених людей, ніж років 10 тому. Досвід університетів Західних країн свідчить, що університет уже не прив'язаний до місцевості (новітні університети відмовилися від географічних назв). Зникає поняття циклічності навчання, тобто вступити в деякі вищі навчальні заклади можна будь-якого дня. Замість факультету тепер є список курсів, необхідних для здобуття кваліфікації. Університет стає глобальною організацією, а в центрі безпосереднього освітнього процесу вже не «професор, який збирає навколо себе аудиторію», а «студент, якого обслуговують професори» [4]. На фоні цих тенденцій вимоги суспільства до освіти помітно змінилися. Однак не все так добре як здається на перший погляд. Якщо матеріально-технічна база вузів України знаходиться на достатньому рівні, то це не можна сказати про матеріально-технічне забезпечення загальноосвітніх закладів. Існує багато проблем, пов'язаних з впровадженням у навчально-виховний процес вітчизняних загальноосвітніх навчальних закладів інформаційно-комунікаційних технологій. Зідно державної цільової програми впровадження у навчально-виховний процес загальноосвітніх навчальних закладів інформаційно-комунікаційних технологій «Сто відсотків» [1] виділяють такі основні проблеми, які пов'язані з їх впровадженням:

- незадовільний стан забезпечення комп'ютерною технікою загальноосвітніх навчальних закладів. Спостерігається певна диспропорція у забезпеченні комп'ютерною технікою сільських (97,8 відсотка) та міських (93,4 відсотка) шкіл. На початок 2009/10 навчального року на 32 учні загальноосвітніх навчальних закладів припадав один персональний комп'ютер (середній аналогічний показник у Європі з 3 до 5);
- використання морально застарілої комп'ютерної техніки. В середньому в Україні потребує оновлення 40 відсотків існуючої у загальноосвітніх навчальних закладах комп'ютерної техніки, що становить понад 5 тисяч навчальних комп'ютерних комплексів;
- незадовільний стан підключення до Інтернету. На початок 2010 року показник підключення загальноосвітніх навчальних закладів у цілому в Україні становить 85 відсотків, а в сільській місцевості – тільки 54 відсотки. Більшість закладів підключено за технологією комутованого з'єднання, що унеможливує ефективне використання інформаційних ресурсів Інтернету в навчально-виховному процесі;
- недостатній рівень підготовки вчителів загальноосвітніх навчальних закладів з питань використання інформаційно-комунікаційних технологій в навчально-виховному процесі;
- недосконалість інформаційно-технологічної, навчально-методичної та організаційної підтримки процесів розроблення, накопичення та обміну інформаційними освітніми ресурсами;
- відсутність системи дистанційного навчання для дітей з обмеженими можливостями та дітей, які перебувають на довготривалому лікуванні та відсутність навиків користуватися комп'ютерною технікою;
- відсутність служби технічної підтримки, яка забезпечувала б працездатність програмних та апаратних засобів у загальноосвітніх навчальних закладах.

Висновок

Отже, стрімкий розвиток мережевих інформаційних технологій, окрім помітного зниження тимчасових і просторових бар'єрів в розповсюдженні інформації, відкрив нові перспективи у сфері освіти. Можна з упевненістю стверджувати, що в сучасному світі має місце тенденція злиття освітніх і інформаційних технологій і формування на цій основі принципово нових інтегрованих технологій навчання, заснованих на застосуванні інноваційних, зокрема, комунікаційних Інтернет-технологій. З їх використанням на сьогодні, незважаючи на існуючі проблеми, уже з'явилася можливість необмеженого, зручного і дуже дешевого тиражування навчальної інформації та швидкої і адресної її доставки не тільки в межах вузу, але і в більш глобальному масштабі користування. Все це і породжує основні напрямки та особливості розвитку комунікаційних інтернет-технологій для застосування їх в організації навчального процесу. То ж базовим

залишається забезпечення високого рівня базових знань в сфері електронно-комунікативних технологій, що і залишається основним завданням нашої держави в області системи освіти на найближчі роки.

1. Розпорядження Кабінету Міністрів України № 1722-р від 27 серпня 2010 року «Про схвалення Концепції Державної цільової програми впровадження у навчально-виховний процес загальноосвітніх навчальних закладів інформаційно-комунікаційних технологій «Сто відсотків».
2. Електронний портал дистанційного навчання Луцького національного технічного університету. [Електронний ресурс]. – <http://elearning.lutsk.ua/>
3. Сайт дистанційного навчання Харківського національного економічного університету. [Електронний ресурс]. – <http://www.ikt.hneu.edu.ua/>
4. Система дистанційного навчання «Віртуальний університет». [Електронний ресурс] . – <http://vu.net.ua/>
5. Каталог українських рефератів. Інтернет технології у дистанційному навчанні. [Електронний ресурс] . – <http://ua.textreferat.com>

УДК 004.72(002.6 + 371.33)

Ліщина Н.М.

Луцький національний технічний університет

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЧНИХ ПЛАТФОРМ ДЛЯ СТВОРЕННЯ СИСТЕМ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ УНІВЕРСИТЕТУ

Наталія Ліщина. Аналіз сучасних технологічних платформ для створення системи електронного документообігу університету. Розглядаються проблеми запровадження у вищих навчальних закладах електронного документа та електронного документообігу. Здійснено опис основних технологічних платформ провідних виробників. Розглядаються можливості вирішення сучасних проблем запровадження корпоративного електронного документообігу на базі застосування порталних технологій, зокрема Microsoft SharePoint та Lotus Notes/Domino. Виділено особливості служби Microsoft SharePoint Online як частини пакету Office 365.

Ключові слова: електронні документи, електронний документообіг, Microsoft SharePoint, Lotus Notes/Domino.

Наталия Лищина. Анализ современных технологических платформ для создания системы электронного документооборота университета. Рассматриваются проблемы внедрения в высших учебных заведениях электронного документа и электронного документооборота. Осуществлено описание основных технологических платформ ведущих производителей. Рассматриваются возможности решения современных проблем внедрения корпоративного электронного документооборота на базе применения порталных технологий, в частности Microsoft SharePoint и Lotus Notes / Domino. Выделены особенности службы Microsoft SharePoint Online как части пакета Office 365.

Ключевые слова: электронные документы, электронный документооборот, Microsoft SharePoint, Lotus Notes / Domino.

Nataliia Lishchyna. Analysis of current technology platforms to create electronic document management system of the university. The problems of the introduction in universities electronic document and electronic document circulation are considered. Description of the main technology platforms leading manufacturers is made. Possibilities of the decision of modern problems of introduction of corporate electronic document circulation on the basis of application of portal technologies, in particular Microsoft SharePoint and Lotus Notes / Domino are considered. Service features Microsoft SharePoint Online as part of Office 365 are excrete.

Keywords: electronic documents, electronic document circulation, Microsoft SharePoint, Lotus Notes / Domino.

Постановка проблеми. Проблеми запровадження у вищих навчальних закладах (ВНЗ) електронного документа та електронного документообігу стають все більш актуальними. Введення системи електронного документообігу (СЕД) зможе допомогти ВНЗ організувати їх роботу з документами і керувати ними протягом усього їх життєвого циклу [3].

На відміну від документів на паперових носіях зі своїми жорсткими рамками, статичною формою і обмеженими можливостями перехід до динамічних цифрових електронних документів забезпечує особливі переваги при створенні, спільному використанні, поширенні та збереженні інформації.

Електронні документи, як динамічні сховища інформації, можуть одночасно використовуватися співробітниками однієї робочої групи, відділу або підприємства загалом. Доступ до них здійснюється протягом кількох секунд. Прискорений доступ до інформації разом зі значною економією коштів може забезпечити й стратегічно важливі конкурентні переваги [1].

Таку роботу з електронними документами забезпечують інформаційні системи, які значаться як системи електронного документообігу.

На основі сучасних систем електронного документообігу забезпечується процес створення, управління доступом і розповсюдженням великих обсягів документів у комп'ютерних мережах, а також здійснюється контроль над потоками документів в організації. Часто ці документи зберігаються в спеціальних сховищах або в ієрархії файлової системи. Типи файлів, які, як правило, підтримуються в системах електронного документообігу, включають текстові документи, електронні таблиці, аудіо, відео дані, графічні об'єкти та документи Web [1].

За допомогою СЕД автоматично відслідковуються зміни в документах, терміни виконання та рух документів, а також контролюються всі їх версії і підверсії. На основі комплексної СЕД охоплюється весь цикл діловодства підприємства чи організації – від постановки завдання на створення документа до його списання в архів, забезпечується централізоване збереження документів у будь-яких форматах, у тому числі, складних композиційних документів.

Мета статті: на основі порівняльного аналізу систем електронного документообігу виявити їх основні проблеми для удосконалення процесу впровадження електронного документообігу в університеті.

Основна частина. У зв'язку з глобальною інформатизацією вищої освіти в Україні постає питання створення ефективної системи документаційного забезпечення ВНЗ. Як відомо, основа документаційного забезпечення вищої освіти — документи, які мають правове значення, оскільки є засобом засвідчення та доведення певних фактів. Вони також використовуються як джерела та носії інформації. В управлінській діяльності документ постає як предмет, і як результат праці.

Найпроблемнішою є організація розроблення та впровадження уніфікованих електронних форм службових документів різних груп навчальної документації у вищих навчальних закладах, оскільки в них переплітаються дві сфери — освітня та трудова, що мають свої законодавчі особливості та вимоги до документів.

Вирішуючи певні проблеми роботи з паперовими документами, при комп'ютеризації електронних форм службових документів різних груп навчальної документації виникають нові проблеми, що в більшості випадків можна подолати.

Впровадження електронного документообігу у вузах зумовило необхідність розроблення уніфікованих електронних форм службових документів різних груп навчальної документації, аналогів паперових форм.

Водночас, наприклад звітні електронні форми, на відміну від паперових, дають змогу оперативно зробити зведений підсумок успішності як за окремими студентами, так і у цілому за академічними групами, окремими кваліфікаціями, спеціальностями, за кафедрами. Слід зауважити, що обрахування успішності ускладнюється у зв'язку з впровадженням у вузах кредитно-модульної системи навчання, а тому функціонування відповідних електронних форм є необхідною умовою для забезпечення організації навчального процесу.

При виборі системи електронного документообігу слід враховувати всю множину чинників, і остаточне рішення бажано приймати на основі комплексного аналізу можливостей СЕД залежно від вимог і специфіки замовника. Критерії, за якими були розглянуті системи, досить універсальні і відповідають завданням автоматизації документообігу як в комерційних, так і в державних установах.

Оскільки за експертними оцінками в найближчі два роки очікується стрімке зростання переходу на відкрите програмне забезпечення (ВПЗ), то окрему увагу приділено СЕД Alfresco, найбільш поширеній на заході системі електронного документообігу саме через відкритість коду (повний Open Source), що спонукає до зниження витрат на ліцензії.[2].

Alfresco — це система управління корпоративними інформаційними ресурсами (ECM) та документообігом, один із лідерів на ринку вільного програмного забезпечення серед програм для організації електронного документообігу. Використовується для управління документами, записами, web-публікацією, груповою роботою в організації. Це система з відкритим кодом, тобто розповсюджується вільно, проте існує і платна версія.[1].

Модульний дизайн дозволяє, при необхідності, вибирати між функціональністю та продуктивністю. Alfresco працює як на Windows, так і на Unix. Система включає в себе вбудований OpenOffice для конвертації різних типів документів, отримування текстових даних для індексування. В комплекті також є Tomcat, який можна замінити на будь-який потрібний web-контейнер. Alfresco має свою власну базу даних користувачів. Однак, можливе автоматичне створення користувачів під час першого входу або синхронізації із зовнішнім джерелом: LDAP, каталогом Microsoft Active Directory, доменом компанії. Система працює з документами будь-якого формату: Microsoft Office, Open Office, pdf тощо. Якщо необхідного формату немає в списку підтримуваних — можна додати свій модуль конвертації в один з підтримуваних і ланцюг конвертації буде побудований в усіх необхідних форматах.

Також, можна повністю відмовитися від стандартного web-інтерфейсу і створити власний. В такому випадку, Alfresco буде використовуватися тільки в якості сховища. Для інтеграції з іншими ПО, підтримуються різні типи аутентифікації. Наприклад, користувач може отримати доступ в систему за допомогою Single sign-on. Якщо ж користувач не авторизований, то система автоматично подасть запит на авторизацію. Варто відзначити, що модель даних підтримує велику кількість успадкування (за допомогою аспектів), тобто у будь-який час можна до об'єкта додати будь-який аспект, і об'єкт придбає усі властивості даного аспекту. Доступ до даних можна гнучко налаштовувати. Система авторизації користується наступними термінами: об'єкт даних, дозвіл, користувач, група, роль. Ролі призначаються для користувачів і груп під час роботи програми.

Розгляд основних технологічних платформ провідних виробників для побудови СЕД і корпоративних порталів, представлених на ринках України, дозволяє, серед безлічі продуктів, виділити три основні групи:

1. Продукти на основі SharePoint від Microsoft.
2. Продукти, побудовані на платформі Lotus Notes/Domino від IBM.

3. Інші продукти, що розвиваються самостійно і що пропонують свої рішення, у тому числі і безкоштовні, але доки не займаючи істотні частки ринку. Оскільки третя група продуктів займає незначну частку ринку, то це означає, що рішення або не мають необхідної функціональності, або погано адаптовані для вітчизняних умов і користувачів. Тому обмежимося порівнянням перших двох платформ: SharePoint від Microsoft і Lotus Notes/Domino від IBM.

Сімейство продуктів SharePoint. SharePoint - ця скорочена назва продуктів і технологій Microsoft SharePoint. Їх можна використати з метою створення сайтів, для спільної роботи і обміну даними з іншими користувачами, для управління документами впродовж усього їх життєвого циклу, публікації звітів. SharePoint включає наступні продукти і технології: SharePoint Foundation, SharePoint Server, Microsoft SharePoint Server 2010, SharePoint Online, SharePoint Designer, SharePoint Workspace, EOS for SharePoint 2010.

Microsoft SharePoint Foundation – безкоштовний додаток до Windows Server, що надає базову інфраструктуру для спільної роботи – редагування, зберігання документів, контроль версій тощо. Також він включає в себе таку функціональність, як «маршрути» руху документів, списки завдань, нагадування, онлайн-дискусії. Раніше Microsoft SharePoint Foundation був відомий як Windows SharePoint Services (WSS).[5].

Microsoft SharePoint Server – платний компонент для інтеграції функціональності SharePoint в роботу застосунків MS Office. Він є надбудовою над Microsoft SharePoint Foundation і розширює його можливості.

Microsoft SharePoint Foundation пропонує базові засоби для створення web-застосунків. До таких засобів належать web-частини, списки даних, бібліотеки документів, середовища виконання робочих потоків і шаблони web-сайтів. Microsoft SharePoint Server має додаткові важливі прикладні функції, а саме:

- систему створення сайтів по запитам користувачів,
- функції бізнес-аналізу,
- технологію Forms Services,
- вбудовані функції пошуку та засоби побудови соціальних мереж.

Всі зазначені функції можуть бути доопрацьовані та доповнені розробниками з метою створення простих у використанні web-панелей для моніторингу основних бізнес-процесів.

Microsoft SharePoint Online – послуга в „хмарі”, що дозволяє створювати сайти і робочі області для спільної роботи з колегами, партнерами та замовниками. Для зберігання документів на сайтах SharePoint служать бібліотеки. Документ до бібліотеки може бути внесений декількома способами: із спеціального меню настільного клієнта Word 2010, завантажений із жорсткого диска натисканням на посилання „Add document” сайту або переписуванням файлу в провіднику Windows, оскільки будь-яка бібліотека Share Point може бути відкрита в Провіднику Windows як мережева папка.

Бібліотеки сайту SharePoint Online сервісу Office 365 можуть бути налаштовані для підтримки механізмів повідомлень. Зокрема, за допомогою механізму повідомлень користувачі можуть електронною поштою отримувати листи у разі додавання або зміни документів у їх бібліотеці. При цьому співробітник, який вніс зміни або опублікував новий документ, звільняється від обов'язку повідомляти зацікавлених осіб – SharePoint Online зробить цю роботу за нього.

Виділяють такі особливості бібліотеки сайту SharePoint Online :

- можливість створення особистого сайту співробітника (mySites) для збереження документів, посилань, контактів, публікацій матеріалів та спілкування в соціальній мережі підприємства;
- можливість створення порталу (team Sites) для спільної роботи з бібліотеками документів, робочими планами і календарями;
- можливість створення внутрішні порталу (intranet Sites) компанії для публікації новин, планів заходів та бізнес-інформації;
- створення закритих порталів (extranet Sites) документів і бізнес-даних для партнерів і замовників;

- легка і швидка розробка професійних web-сайтів для широкої аудиторії;
- можливість створення документів Office і збереження їх безпосередньо на порталах SharePoint Online;
- можливість управління правами доступу до документів для захисту службової інформації;
- робота з важливими документами в автономному режимі;
- спілкування з колегами в реальному масштабі часу;
- можливість слідкувати за певними версіями документів й організовувати колективну роботу щодо підготовки документів;[3]

Важливим моментом під час вибору програмних засобів є зручність роботи з ними і «люб'язний» інтерфейс. Тому необхідно аналізувати і враховувати, якими вони мають бути, щоб користувачеві було максимально комфортно працювати. Для цього найкраще використовувати те програмне середовище, до якого він вже звик. В цьому плані доцільно обрати Windows SharePoint Services, адже роботу з ним можна організувати в середовищі Microsoft Office, яке є найпоширенішим.

На базі SharePoint можна створити корпоративний web-портал, де розміщуються документи, що використовуються одночасно, або спеціалізовані програми, такі як «вікі» або «блоги». Дані в SharePoint організовані у вигляді списків (наприклад, завдання, обговорення, календарі) і бібліотек документів. У SharePoint в списках даних зберігаються дані для кінцевих користувачів. Кінцеві користувачі можуть створювати таблиці за схемами за допомогою користувацького інтерфейсу SharePoint або безкоштовного програмного продукту SharePoint Designer, а потім створювати, редагувати і переглядати дані в цих таблицях. Розробники можуть звертатися до цих даних за допомогою відповідних програм.

Функціональність SharePoint передається користувачеві за допомогою web-частин – це створювані розробником елементи web-інтерфейсу, які згодом можуть застосовуватися на безлічі web-сторінок. Вперше подібний підхід був застосований в ASP.NET, а потім web-частини були використані і в SharePoint, де додавати їх на сторінки можуть самі користувачі, а за налаштування таких web-частин відповідають IT-фахівці. Такі web-частини розміщуються на сторінках, які, в свою чергу, розташовані на порталі та доступні користувачеві через браузер.

З боку користувача SharePoint не вимагається встановлення специфічного програмного забезпечення чи особливих знань в галузі інформаційних технологій. Для роботи з цією системою на комп'ютері необхідно мати встановленою систему Microsoft Windows, та, з точки зору діловодства і документообігу, Microsoft Office.

Lotus Domino та Notes — це інтегрована система для колективної роботи, призначена для збирання, організації й розподілення інформації та знань, багатобічний програмний продукт, до складу якого входять такі інформаційні технології:

- документоорієнтована база даних;
- засоби розробки додатків;
- система електронної пошти;
- система реплікування (тиражування) документів, інформації і додатків;
- засоби захисту інформації і розмежування доступу;
- засоби календарного планування і складання розкладів;
- web-технології і технології інтернету;
- засоби інтеграції з реляційними базами даних, системами керування ресурсами підприємств (ERP) і транзакційними системами.

Domino та Notes — це клієнт-серверна технологія, де сервером є Lotus Domino, а клієнтською частиною — Lotus Notes. Унікальність сервера Domino полягає в тому, що це ще і web-сервер, і поштовий сервер, який підтримує стандарти Інтернету, тому клієнтською частиною для роботи з додатками Domino й електронною поштою можуть бути web-браузери та інші поштові клієнти Інтернету.

Документоорієнтована база даних Domino/Notes. Сховищем об'єктів, в яких зберігаються дані, є NSF-файли (Notes Storage File). Бази даних Domino і Notes відрізняються від реляційних систем управління базами даних (СУБД), де дані описуються за допомогою таблиць. Реляційна база даних твердо структурована, а кожний запис у таблиці має однаковий набір полів, обсяг пам'яті під який визначений і виділений заздалегідь.[4]

Структура документа Notes визначається формою, що містить набір полів різних типів. База даних Notes може зберігати будь-які типи даних — від простого тексту, чисел, часу і дати до

форматованого тексту, графічних образів, звуку, відео і довільних даних, що можуть зберігатися у вигляді приєднаних об'єктів у своєму рідному форматі.

Документ може мати як структурований, так і неструктурований формат, тому Notes може зберігати й обробляти такі масиви даних, обробка яких важко піддається реляційним та іншим системам баз даних.

Завдяки використанню документоорієнтованої моделі обробки Notes надає користувачам такі можливості.

- робота з текстами та об'єктами як таблиці, відео та графічні дані, тощо;
- взаємозв'язки між документами;
- повнотекстовий пошук;
- керування версіями.

Висновки. В загальному випадку, одна з головних цілей впровадження СЕД полягає у створенні спільної платформи для інтеграції усієї різноманітності процесів, що автоматизуються, в єдину систему. Вибору того або іншого рішення по впровадженню СЕД повинне передувати інформаційне обстеження об'єктів автоматизації з подальшою експертною оцінкою можливих варіантів впровадження СЕД. Для вищих навчальних закладів найбільш загальним є підхід до впровадження СЕД на базі готового рішення, реалізованого на універсальній технологічній платформі, з подальшим її розвитком і урахуванням існуючих або знову виникаючих інформаційних запитів.

Результати порівняння СЕД на основі SharePoint від Microsoft і Lotus Notes/Domino від IBM дають підстави стверджувати, що продукти компанії Microsoft і IBM мають усі функціональні можливості для організації СЕД і порталу організації. Це інструменти для спільної роботи, зберігання контенту, роботи з поштою і офісними додатками, передачі миттєвих повідомлень, засоби автоматизації бізнес процесів і реалізації інших можливостей.

1. Електронний документообіг: сучасні тенденції та проблеми провадження [Електронний ресурс]. – Режим доступу: http://www.rusnauka.com/34_VPEK_2012/Philologia/7_121024.doc.htm
2. DOCFLOW Україна: все об електронном документообороте, бизнес-конференция [Електронний ресурс] – Режим доступу: <http://www.docflow.ua/conference>
3. Ткачук Г. І. Використання електронної системи документообігу у ВНЗ / Г. І. Ткачук, С. А. Постова // Магістратура в умовах євроінтеграційних процесів вищої школи. – Житомир: ЖДУ, 2014. – С. 254.
4. Стив Керн, Дебора Линд, Дебора Пенни и др. Lotus Notes и Domino 6. Руководство разработчика. Пер. с англ.- Киев, ООО "DiaSoft", 2005.- 880 с.
5. Ноэл, Майкл, Спенс, Колин. Microsoft SharePoint 2010. Полное руководство. : Пер. с англ. — М. : ООО —И.Д. Вильямс, 2011. — 880 с.

Panasyuk Nataliia
Lutsk National Technical University

CONCEPTUAL APPROACHES TO MANAG THE EDUCATION QUALITY IN HIGHER TECHNICAL EDUCATIONAL INSTITUTIONS

Панасюк Н.Л. Концептуальні підходи до управління якістю освіти у вищих технічних навчальних закладах. У даній публікації розглянуто концептуальні підходи до управління якістю освіти у вищих технічних навчальних закладах, які ґрунтуються на засадах гуманізму, демократії, національної свідомості, взаємоповаги між націями і народами, громадяни мають право безоплатно здобути вищу освіту в державних і комунальних навчальних закладах на конкурсній основі, незалежно від статі, раси, національності, соціального і майнового стану, роду та характеру занять, світоглядних переконань, належності до партій, ставлення до релігії, віросповідання, стану здоров'я, місця проживання та інших обставин.

Ключові слова: управління, якість освіти, концептуальні підходи, вищі технічні навчальні заклади.

Панасюк Н.Л. Концептуальные подходы к управлению качеством образования в высших технических учебных заведениях. В данной публикации рассмотрены концептуальные подходы к управлению качеством образования в высших технических учебных заведениях, основанных на принципах гуманизма, демократии, национального сознания, взаимоуважения между нациями и народами, граждане имеют право бесплатно получить высшее образование в государственных и коммунальных учебных заведениях на конкурсной основе, независимо от пола, расы, национальности, социального и имущественного положения, рода и характера занятий, мировоззренческих убеждений, принадлежности к партиям, отношения к религии, вероисповедания, состояния здоровья, места жительства и других обстоятельств.

Ключевые слова: управление, качество образования, концептуальные подходы, высшие технические учебные заведения.

Panasyuk Nataliia. Conceptual approaches to manag the education quality in higher technical educational institutions. Education in Ukraine based on the principles of humanism, democracy, national identity and mutual respect between nations and peoples. The general scientific approaches the applying of which will be required during scientific research are defined in the article.

Keywords: higher education, research, research approaches, educational training, teaching scientific approaches.

Formulation of the problem in general. The right to education in Ukraine is constitutional and fixed in Article 53 of the Constitution of Ukraine. According to it, complete general secondary education is compulsory. Citizens have the right to obtain free higher education in state and communal educational establishments on a competitive basis, regardless of gender, race, ethnicity, social or economic status, type and nature of occupation, philosophical beliefs, membership of political parties, religion, creed, state of health, residence and other circumstances [1].

The content of education - is due to society's needs and objectives of the system requirements of knowledge and skills, philosophy and social and professional skills of the future specialists formed in the learning process from the perspectives of science, engineering, technology and culture [1].

Arguably, the quality management - a coordinated activity, which consists in directing and controlling the institution concerning the quality of education. Quality Management - a new way of organizing efforts of many people. Its purpose is to provide focus all resources on consumer demand. This leads to significant changes in the relationship between those who govern and those who actually performs the work.

The aim of education is the full development of human personality and the highest values of society, the development of talents and mental and physical abilities, education of high moral character, formation of citizens capable of deliberate choice, enrichment of the basis of intellectual, artistic and cultural potential of the people, improve the educational standards of the people, to ensure national economy with qualified professionals.

Analysis of the modernization concept of modern education for the period until 2014, materials of Expert-analithic centre, International Strategic Group of Experts of the Council of Europe materials discussion of the problems of modernization of education possible to determine the basic characteristics of quality to be met by education [3]. These characteristics include:

"the objectives and results of the general education requirements of the modern social system of education, due to the transition to an open and democratic society with a market economy, the transition to the overwhelming focus on students mastering a given amount of educational material to focus on developing the ability of students to independent problem solving in different areas on the basis of earned social experience;

"appropriateness of its educational goals and cognitive abilities of all students, improving the accessibility of educational content, creating additional conditions for the expansion and deepening students' knowledge in their field of education;

"raising the demand for education outcomes in the activity;

"compliance of educational activities with the preservation of health and providing psychological comfort of all participants in the educational process".

The wording of Article purposes. Achieve of a new quality of education due to the revision of the concept of "education", which is interpreted within the concept of modernization as a way to solve significant problems through Earned social experience.

The task of defining the purpose of education is to select goals in line with the nature of education and orientation of participants in the educational process, that is, those that reflect the most important problem for them.

The main material research. The first problem facing with all students is the need of mastering the ways of training activities. Therefore, the first purpose of education is to teach so that students learn. Learn to learn - it means to form students' learning motivation values, develop the ability to use different sources of information, to teach them to use effective methods of cognitive activities help students understand the content of education.

The second problem, more precisely, a group of problems that occur all the students, but rather, all people - a specific life problems that relate to different areas - family and home, entertainment, social, political, labor and related with the implementation of certain human social roles - citizen, family member, customer, client, resident, voter and so on.

This is a problem of orientation in habitat and ensure their own safety orientation in the law and administrative structures in the norms of behavior and forms of communication in the world of consumer values and aesthetic values, the phenomena of nature and social life.

This list can be extended (important to more rigorously define the range of life's problems, independent decisions are necessary to prepare graduates). The purpose of education is to teach students in order to solve standard life situations, that is, to understand the nature and significance of the issues, learn the existing rules and regulations of their decisions, to be able to justify focus on sources of information that can help in solving emerging everyday tasks.

The third group of problems concerns the orientation of the world's values. This is the attitude to nature, cultural phenomena (and spiritual and material), historical events and historical characters to the learning process, to man. The problem for everyone is to determine their attitude to these phenomena.

The purpose of education is to teach to navigate the world of values - to help students understand the essence of value attitude to the world, they create knowledge about basic values, develop the ability to determine criteria for assessment of reality.

The fourth group of problems is related to continuing education, preparation for the transition to the system of professional education. The purpose of education is to prepare the professional choice. This training during the transition to a market economy means the following: orientation in the labor market; orientation system of vocational education; orientation in their own interests and abilities; availability of skills, with reference to the value of vocational education; readiness for learning environments in vocational education.

The fifth group of problems relates primarily to foster community development. There is a need to prepare students for life in a society that is changing. On the one hand, such training can be provided through the development of basic students, primarily responsible, values; on the other hand, such training involves the development of students such as how navchenist. The essence of education is to teach so that the solution of non-standard tasks, among which include research, creative, ideological, etc. [5].

Analysis of the above goals of education suggests the existence of common components goals: motivation, value orientation, cognitive, operational and activity. These components can be put as relatively independent goals, specifying the general objectives of education and make the second level in the hierarchical system of educational programs.

The first objective of this level is to increase range of important issues for students. The emphasis on personally meaningful problems does not mean Aesop focus of the educational process. The pedagogical challenge is to socially significant problems make problems meaningful to students.

The award underlines the emphasis that the inclusion of the individual in the solution to a social problem, awareness of the importance of this personality, communication personal and social interests. At the same time the proposed statement of purpose emphasizes that personal development is reflected in the change of the scale of the world is perceived as part of the self [5].

The second goal is to expand cultural and educational environment of students. This means a change in the sources of information used in the educational activity. In the most general form of change is to move from sources adapted to the "primary sources".

Adapted sources - these are different kinds of textbooks used in schools [4]. The circle of "primary sources" is rather broad: achievements artistic, scientific and popular, political, scientific literature; original works of art (visual, musical, theatrical, etc.); historical monuments, attractions, media, etc.

The content of education lies in the fact, in order to enable students to real culture and teach them "not learn from textbooks." It should be noted that very important factors in the expansion of cultural and educational environment of students in modern conditions is knowledge of foreign languages and possession of information technology [2].

The third objective - the development of ways of activities. Modern education requires students to learn the ways of many small decisions subject of many small tasks. The problem is to build students' universal ways of life. The relative success in this reach (and then not always) in the formation obscheuchebnyh skills and general intellectual skills (compare, analyze, organize, synthesize, etc.). Mastering these skills will certainly extends cognitive capabilities of students, but did not provide solutions to complex problems, as these skills are just some of the techniques. It would be important to identify opportunities each school subject in the formation of "above subjects" ways of life, among which include system approach, modeling, forecasting, planning, heuristic methods, etc. [4].

The fourth goal is to build "of knowledge" base problem solving - the assimilation of facts, information, rules, scientific concepts, laws, principles, theories, etc.

General objectives of school education can be achieved only with achieving the goals of the second level - expanding the range of important issues for students, their cultural and educational space, forming a universal way of activities and knowledge necessary to solve problems [2].

From the point of view of the above general objectives of education is to achieve its result the students level of education that meets their personal potential. When erudition understood individual personal educational outcomes, resulting in the change of the scale of the world is perceived as part of the self [5].

The level of education is determined by the achieved level of implementation of the general objectives of education. There are levels of education that can be achieved in education: elementary literacy, functional literacy, general cultural competence, pre-professional competence, methodological competence.

Defining the purpose of education determines access to new educational outcomes. Access to new educational outcomes, in turn, involves the assessment of the results already achieved, identifying issues of achieving new results [4].

The educational program developed by higher education institutions in order to satisfy the need for education, training, training or retraining - that reaching a certain social effect (change in educational or professional level).

With the help of this product the higher education institution goes to the market. University offers market specific educational services in the form of lectures, seminars, etc., it offers services, united by a common goal and provided with adequate resources.

Educational institutions that are not professionals and equipped with computer labs can not offer educational programs to this area. However, even having given resources, higher education institution offers its customers the disparate lectures or workshops, as coming out of the educational program in that field, which includes some content and organization of the educational process, the process control system and its system methods, material and staffing. Therefore, the product of higher education institution defined as an educational program.

Education Program - a set of educational services aimed at changing the educational level or training consumers and provided with adequate resources educational organization [4].

Depending on its capabilities and customer needs, schools offer different range of applications that can be classified in a number of features.

The level of the proposed education program may be pre-university, undergraduate, master's, graduate, professional training programs, etc. [4].

By targeting specific specialty applications may be in finance, marketing, merchandizing, personnel management and other occupations that are in demand in the market.

The form of training programs vary daily, evening, part-time, remote, external, etc.

By teaching methods used programs can be traditional, problem-based learning programs, programs based on the analysis of business situations and more.

If additional components when the goal is not only sufficient resources of one of the university, educational programs may be so-called "sandwich courses" that include mandatory practice period between two periods of theoretical training programs "internship" as a theoretical study running parallel to the practical work in this direction, international, program of study when foreign university partner is an integral part of the learning process, etc. [3].

In the case of "sandwich course" and "internship" resources combined with the resources of the company in the implementation of international programs combined resources of two or more schools, located in different countries.

The new look of educational programs appears in response to market demand and technical capabilities (resources) University changing.

The rapid development of information technology, for example, brought to market distance learning programs that combine independent distance education possibilities of the individual (in the case of distance learning through computer) consultation with teachers [1].

Scope of market relations in higher education represented relationships between universities that train and retrain young professionals, businesses, consumers young professionals by young professionals and as a specific commodity.

We agree with the view of graduates as a specific product for the company, which employs graduates to meet their staffing requirements necessary qualifications [3].

But we can not take a definite view of the high school as a producer, and the goods in the form of young professionals. After all, on the same grounds and the manufacturer of this particular product can be called parents, environment and surroundings of the graduate.

Even in the formation of significant professional knowledge and skills used by employers in the labor market involved not only high school. Self while studying in high school, parallel training in various courses, the influence of family environment and so brings graduates with the knowledge and skills that are the questions on the labor market [2].

In addition, professional quality can not be seen in isolation from the other characteristics of persons as significant as in employment and in the profession.

Thus, the model of education quality management system can be represented as tasks, which provides qualitative changes in education to achieve new educational results that meet contemporary social needs.

Analysis of the modernization program of modern education can provide a number of problems of quality management education, which should be solved. These tasks include upgrading the educational process, educational way of life, educational environment: training teachers to improve the quality of decision tasks.

In the middle of each task can be divided into separate areas of the teaching staff, which determine the conditions for the transition to the new educational outcomes, specifying the nature of the changes that can be provided.

Modernization of the educational process includes:

Organization of the transition to the new educational standards:

"Selecting the curriculum.

"Preparation of work programs focused on substantive and metapredmetni results.

"Planning the learning process.

"The choice of the training regime.

"Scheduling training sessions.

"Changes in educational technologies.

"The choice of subject lines and methodological manuals.

"The organization of specialized education.

"The organization of network interaction.

"The selection and deployment of elective and optional courses.

The educational process:

"Changes in the final and intermediate certification students.

"Evaluation of Object and metapredmetnyh results.

"Evaluation of functional literacy and key competencies of students.

"Determining quality of the educational process.

The way educational life:

"The rules of interaction of the educational process (students, teachers, parents). Protection of children's rights in the educational institution.

"A combination of different forms of educational activities (training, self-education, additional education, experience social and creative activities).

"Self-government".

"Reconciliation of rights and duties of the educational process.

"Teaching and social initiatives. Social design.

"The public examination and certification of public education outcomes.

Educational environment:

"Interaction with parents as teaching partners.

"Cooperation with social organizations.

"Interaction with government agencies and institutions, enterprises of different ownership.

Teacher training to the task to achieve a new quality of education:

"The system of management of teaching goal-setting.

"The forms and methods of training.

"The program growth of professional skills of teachers.

"Measures to promote professional development of teachers.

"Methods of interaction between members of the teaching staff.

Model to improve the quality of education is cyclical, evolutionary and aims to support the process of education. Its implementation involves variation in quality management programs of education.

Selecting depends on the results achieved, the characteristics of the educational process, educational environment, and potential for professional growth of teachers.

At the same time, program options can be generalized into three main types, defining the nature of process control to improve the quality of education:

"Programs to meet new requirements and procedures;

"Programs aimed at fostering innovation teachers;

"Program aimed at creating conditions for waiver of the usual stereotypes.

At the same time it is possible to assume the existence of actual practice mixed options that combine all three approaches to quality management.

Conclusions and recommendations for further research. The priority of social development in the twentieth century, the international community recognized as undoubtedly is the quality of education, which is subject to all other indicators of human life.

We are the witness of qualitative changes in the development of civilization and conditions of life, and thus the functioning of education and understanding of quality. In their reports, scientists noted that quality education is the cornerstone of modern education paradigm, absolute priority educational policy of most countries, and Ukraine is no exception in this process.

The world realized that in high-tech information society, the quality of education is the main argument of human development to provide a level of professional competence and life of a man who would satisfy her desire for self-improvement and self-development and, consequently, the public's demand for educated and highly cultured citizens.

Problem institutional approach to quality management education is relevant for a number of factors:

- flexibility and dynamism of modern life that requires constant learning new individual competencies;
- transition to electronic information and communication technologies and development of e-governance of educational institutions;

- awareness of the value of human capital;

- Ukraine's accession to the common European education area (Bologn' Process);

- unequal starting opportunities for teaching children in families with different social status and living in different levels of socio-economic development of the territories that determines the need for alignment starting opportunities through presentation of common requirements for quality education.

Quality education is now regarded as one of the indicators of quality of life, an instrument of social and cultural cohesion, economic growth and declared both the international and domestic levels. And every state (and Ukraine as well) that takes care of your rating civilized world and his future, should develop a strategy and tactics of quality management education.

1. The Bologna Process: Documents / way.: Z.I. Timoshenko, A.M. Grehov, Y.A. Gapon, Y.I. Palena. – K.: Type of Evrup. Univ, 2004. – 169 p.
2. Breddyk W. Management in the organization / W Breddyk. – M: INFA-M, 1997. – 344 p.
3. Goncharenko S.U. Teaching research: methodological advices to young scientists: teach. Handbook. / S.U. Goncharenko. – K.: Vinnitsa, 2008 – 278 p.
4. Karamushka L.M. Psychology of the educational management: teach. Handbook./ L.M. Karamushka. – K. Lybid, 2004. – 424 p.
5. Peter F. Drucker, Management: Tasks, Responsibilities, Practices (New York: Harper Row, 1973), - 715 p.

УДК 004.413 (045)

Сидоров М.О.¹, Жигаревич О.К.²

¹Національний авіаційний університет,

²Луцький національний технічний університет

СИСТЕМАТИЧНЕ ДОСЛІДЖЕННЯ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЩОДО ЕКОСИСТЕМ НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Сидоров М.О., Жигаревич О.К. Систематичне дослідження літературних джерел щодо екосистем навчального програмного забезпечення. Розробка програмного забезпечення у нашій державі є глобальним явищем, яке включає в себе соціально – технічні системи. Сучасне програмне забезпечення характеризується масивністю, складністю, швидкою інтеграцією у соціальні відносини, застосуванням нових принципів при формулюванні задачі, проектуванні програмного забезпечення, розробці, супроводі, обміні та використанні. Саме взаємодія різних сфер впливу на розробку програмного забезпечення спонукала до застосування нових принципів при визначенні життєвого циклу та середовища розробки в цілому.

Ключові слова: екосистема, програмне забезпечення, дослідження, літературні джерела.

Сидоров Н.О., Жигаревич О.К. Систематическое исследование литературных источников относительно экосистем учебного программного обеспечения. Разработка программного обеспечения в нашем государстве является глобальным явлением, которое включает в себя социально – технические системы. Современное программное обеспечение характеризуется массивностью, сложностью, быстрой интеграцией в социальные отношения, применением новых принципов при формулировке задачи, проектировании программного обеспечения, разработке, сопровождении, обмене и использовании. Именно взаимодействие различных сфер влияния на разработку программного обеспечения побудило к применению новых принципов при определении жизненного цикла и среды разработки в целом.

Ключевые слова: экосистема, программное обеспечение, исследования, литературные источники.

Sidorov N.O., Zhyharevych O.K. Systematic study of the literature on ecosystems of educational software. Software development in our state is a global phenomenon that includes socio – technical systems. Modern software is characterized by the massiveness, complexity, fast integration into social relations, the application of new principles in the formulation of the task, the software design, development, maintenance, exchange and use. It is the interaction of the various spheres of influence on software development has led to the introduction of new principles in determining the life cycle and development environments in general.

Keywords: ecosystem, software, research, literary sources.

Вступ. Розробка програмного забезпечення у нашій державі є глобальним явищем, яке включає в себе соціально – технічні системи. Сучасне програмне забезпечення характеризується масивністю, складністю, швидкою інтеграцією у соціальні відносини, застосуванням нових принципів при формулюванні задачі, проектуванні програмного забезпечення, розробці, супроводі, обміні та використанні. Саме взаємодія різних сфер впливу на розробку програмного забезпечення спонукала до застосування нових термінів при визначенні життєвого циклу та середовища розробки. Програмне забезпечення швидко розвивається і поширюється у різних сферах життя сучасного українця. Кожна друга людина не може себе уявити без технічного пристрою, який допоможе знайти потрібну інформацію на пошукових сайтах, переглянути новини, набути знань у тій чи іншій сфері життя. Технічний прогрес включає в себе і прогрес програмного забезпечення, який народжується, розвивається, перебігає, живе в певному середовищі. Процес поєднання програмного забезпечення і суспільства викликає потребу у використанні додаткових понять і концепцій для вивчення і опису програмного забезпечення. Одним із методів вирішення даної проблеми є використання концепції екосистем програмного забезпечення. Саме тому екосистеми програмного програмного забезпечення можуть стати додатковим інструментом для дослідження навчального програмного забезпечення.

Аналіз останніх досліджень. За останні кілька років поняття «екосистема програмного забезпечення» і супутні терміни активно використовуються розробниками і дослідниками програмного забезпечення. Огляд веб-сайтів провідних розробників програмного забезпечення показує, що більшість з них застосовують поняття «екосистема програмного забезпечення» (наприклад [2-4]), позначаючи ним системи, що включають підприємство розробника, його програмне забезпечення і партнерів. Розробники застосовують цей термін «як є», без теоретичних бази або посилань на відповідні праці. Деякий виняток становить корпорація «Майкрософт», яка дає визначення екосистеми програмного забезпечення як сукупності взаємодій і взаємовпливів організацій (державних, навчальних і комерційних) і індивідумів, що працюють із програмним забезпеченням [4].

Наукові дослідження, які використовують поняття екосистем, на сьогоднішній день представлені кількома працями [6-8]. В роботі [6] «Екосистеми програмного забезпечення» автори описують типові елементи екосистем та їх контекст, роблячи спробу на якісному рівні спрогнозувати характеристики розвитку глобальної екосистеми програмного забезпечення з точки зору подальшого підвищення ефективності технологій розробки, появи і розвитку нових областей застосування.

Дослідження екосистеми навчального програмного забезпечення, на даний час є дуже актуальним. Використання та дослідження навчального програмного забезпечення у вищих навчальних закладах України дає багато переваг для оволодіння тією чи іншою професією, а пізнання, засвоєння інформації в певному середовищі і породжує екосистему навчального програмного забезпечення.

Метою дослідження є метод дослідження літературних джерел, щодо екосистем навчального програмного забезпечення.

Об'єкт дослідження - процес пошуку літературних джерел щодо екосистем навчального програмного забезпечення.

Предмет дослідження - методи пошуку літературних джерел.

Метод дослідження полягає в тому, щоб забезпечити найкращий підбір основної літератури щодо екосистем навчального програмного забезпечення.

Основним завданням є завдання здійснити систематичний аналіз та дослідження навчальної, монографічної та наукової літератури, законодавчої та нормативної бази відповідно до теми; зробити порівняльний аналіз стандартів пошуку інформації на різних пошукових сайтах Європи, США, та України.

Систематичний огляд літератури – це один з методів дослідження, що останнім часом набуває все більшого поширення у сфері дослідження навчального програмного забезпечення [6]. Systematic mapping study – це метод, що спочатку був дуже поширеним у сфері медичних досліджень, однак, на жаль, не використовується значною мірою при проведенні досліджень програмного забезпечення. На думку вченого Бейлі існує лише один чіткий приклад systematic mapping study в межах програмного забезпечення [1, 1]. Systematic mapping study являє собою структурний тип дослідження надрукованих публікацій та наукових робіт, а також належним чином їх класифікувати. Таким чином, можна побудувати свого роду візуальний підсумок-огляд, так звану карту результатів. Використання методу systematic mapping study можна провести набагато точніше та детальніше дослідження за короткий час.

Systematic mapping study – це більш «розгорнута» форма систематичного огляду літератури спрямована на повне планове дослідження, не на пошук конкретних відповідей на детальні питання та завдання початкового дослідження [6]. Таке дослідження спрямоване на заповнення свого роду «нестач» у ряді початкових досліджень, де необхідним є проведення новітньої науково-дослідної роботи, а також на пошук спірних питань, що потребують додаткових уточнювальних досліджень. А systematic mapping study дозволяє проводити дуже точні та детальні дослідження у необхідній галузі.

Переваги використання методу Systematic mapping study

На даний час даний метод є основним для дослідження, і забезпечує наступні переваги:

- Підсумовує існуючі результати досліджень, а також емпіричний досвід у галузі, означити переваги та недоліки певних методів.
- Визначає недоліки та «недостачі» у досліджуваній галузі, підготувати базу для проведення майбутніх науково-дослідних робіт.
- Будує структуру та порядок проведення нових досліджень.

Ознаки systematic mapping studies

Ознаки systematic mapping study від традиційного систематичного огляду літератури, що відрізняють його є наступні:

- Традиційно починається з побудови оглядового протоколу, що визначає завдання та питання дослідження, виходячи з яких підбираються відповідні методики проведення науково-дослідної роботи.
- Засновується на певній визначеній стратегії, що спрямована на пошук якомога більшої кількості значимої літератури.

- Читачам надається повний та цілісний опис всього процесу дослідження (слід враховувати те, що процес пошуку літератури в електронних бібліотеках майже неможливо відтворити).
- Потребує використання повних і чітких якісних критеріїв включення і виключення первинно знайдених наукових робіт.
- Конкретизують інформацію, отриману в результаті кожного первинного дослідження, у тому числі якісні критерії, за якими оцінюється кожне початкове дослідження.
- Передумова для кількісного мета-аналізу.

Форма systematic mapping study

Вчені Petticrew та Roberts пропонують використання даного методу систематичного аналізу літератури з метою з'ясування, "які саме науково-дослідницькі роботи відповідають критеріям пошуку, коли вони були надруковані, в яких базах даних (бібліотеках) зберігаються, яких результатів було досягнуто, а також дослідники яких країн займаються дослідженням поставлених питань із статті" [7].

Перші етапи *systematic mapping study* є дуже схожими з етапами традиційних систематичних оглядів літератури:

- **пошук.** Визначення низки науково-дослідницьких робіт, в яких можуть розкриватися поставлені питання/ завдання дослідження;

- **включення/виключення науково-дослідницьких робіт із дослідницького процесу.**

Відбір літератури для проведення подальшого дослідження;

- **відповідність питанням/завданням дослідження.** Якщо необхідно, проведення якісного оцінювання віднайдених науково-дослідницьких робіт.

Визначення питань/завдань дослідження

Основною метою *systematic mapping studies* є надання широкого та повного огляду рівня досліджуваності заданої області, а також визначення кількості та якості надрукованих науково-дослідницьких робіт із даної теми. Вторинною метою є надання чіткого графічного звіту про проведений *systematic mapping studies* [5]. У нашому дослідженні було визначено п'ять питань (П) для проведення систематичного огляду літературних джерел щодо екосистем навчального програмного забезпечення, формулювання питань проводилось двома мовами українською та англійською:

(П1) Які екосистеми навчального програмного забезпечення існують та успішно функціонують на даний час? /What ecosystems of educational software exist and are successfully functioning at the present?

(П2) Якими є основні складові екосистеми навчального програмного забезпечення?/ What are the main elements of educational software ecosystem?

(П3) Які існують зауваження щодо структури екосистем навчального програмного забезпечення? / What remarks are there regarding the structure of educational software ecosystems?

(П4) Які недоліками виявлені у екосистемах навчального програмного забезпечення?/What shortcomings have been exposed in educational software ecosystems?

(П5) Які можливі вирішення існуючих проблем та шляхи покращення екосистем навчального програмного забезпечення?/How can the current defects be eliminated and what are the ways to improve educational software ecosystems?

Критерії включення/виключення науково-дослідних робіт у дослідження

Одним із найважливіших етапів у систематичному огляді літератури є визначення якісних критеріїв включення/виключення робіт у дослідження.

Наступними є критерії включення робіт до нашого дослідження:

- дослідження, надруковані англійською та українською мовами;
- дослідження, які містять у назві або тексті науково-дослідницької роботи терміни «екосистема програмного забезпечення», «навчальне програмне забезпечення»;
- кандидатські та докторські роботи з екосистем програмного забезпечення;
- навчання з нефіксованим часовим періодом;
- «сіра література», включаючи доповіді, надруковані незалежно від освітніх та професійних організацій.

Наступними є критерії виключення робіт з нашого дослідження:

- науково-дослідницькі роботи, що повторюються у декількох електронних бібліотеках (базах даних). У таких випадках, тільки одна робота включається до наукового дослідження;
- науково-дослідницькі роботи, що мають однакові результати дослідження. У такому випадку, до уваги береться лише одна найбільш повна робота;
- науково-дослідницькі роботи та книги, що є недоступними до завантаження;
- роботи, основною темою дослідження яких не є екосистеми програмного забезпечення, або задана тематика лише згадується оглядово.

Далі ми надаємо результати проведеного *systematic mapping studies*. Відповідно до Kitchenham [8], оптимальним є використання у процесі пошуку як електронної (автоматичної) методики пошуку, так і неавтоматичної. У нашому дослідженні ми використовували обидві. В процесі неавтоматичного пошуку ми перевірили роботи, представлені у ряді наукових конференцій, на сайтах певних вчених-дослідників, а також на форумах досвідчених спеціалістів у галузі освіти створення програмного забезпечення у спеціалістів які керують науковими проектами. Рішення не використовувати у процесі дослідження лише автоматичну методику пошуку літератури було прийняте згідно із дослідженнями вчених, які стверджують що максимальної об'єктивності та повноти дослідження можна досягти лише використовуючи обидва методи – як автоматичний, так і неавтоматичний. Неавтоматичний пошук літератури також гарантує більшу адекватність знайдених науково-дослідницьких робіт та їх відповідність поставлених дослідницьких питань. Електронний автоматичний пошук науково-дослідницьких робіт було здійснено у наступних електронних базах даних (бібліотеках): IEEEExplore Digital Library, ACM Digital Library, Elsevier ScienceDirect, EI Compendex, Scopus and Web of Science. Ці електронні бібліотеки є вивереними та надійними джерелами літератури в галузі програмної інженерії і комп'ютерних наук в цілому. Нашим завданням першочергово було визначення ключових термінів та їх синонімів з поставлених питань / завдань дослідження. Також було побудовано наступну стратегію пошуку:

- визначити основний час та період для здійснення пошуку;
- перевірити низку підібраних ключових слів у науково-дослідницьких роботах, що вже були знайдені нами;
- пошук та визначення альтернативних та додаткових синонімів та термінів для здійснення дослідження.

Результати дослідження. Під час дослідженні були визначені наступні етапи підбору літературних джерел щодо екосистем навчального програмного забезпечення. Спочатку ми визначили 3 електронні бази даних (бібліотек), зазначених вище, вказали період часу для пошуку 2001 – 2014 рік. Результатом пошуку в електронних базах даних стали 40,3 українськомовних та 72,60 англomовних науково-дослідницьких робіт, однак лише 25 з них були доступні для завантаження українськомовних джерел та 75 англomовних. Після завантаження лише 15 українськомовних робіт були включені у подальше дослідження після оцінки їх відповідно до визначених якісних критеріїв включення/виключення, та 25 літературних джерел англійською мовою. Оскільки ми використовували декілька електронних баз даних, багато робіт мали дублікати. Остаточна кількість наукових робіт було скорочено до 23. У табл. 1 надано підсумки процесу пошуку літературних джерел на кожному етапі дослідження.

Таблиця 1. Процес формування списку літературних джерел, щодо екосистеми навчального програмного забезпечення

Види електронного пошуку	Етап I		Етап II		Етап III
Автоматичний (електронний) пошук за українськими ключами	40,30	Крок 1	25	Крок 2	7
Автоматичний (електронний) пошук за англійськими ключами	72,60		75		16
Загальна кількість робіт	112,9		100		23

У процесі відбору та аналізу літературних джерел ми користувалися методиками, запропонованими вченими Т.Дуба та Т.Dingsoyr [5]. Використовувався метод відкриття та порівняння інформаційних джерел із якісним аналізом (тобто ті що підпадають по темі).

На рис. 1 показано класифікацію знайдених літературних джерел по екосистемах навчального програмного забезпечення відповідно до року її публікації.

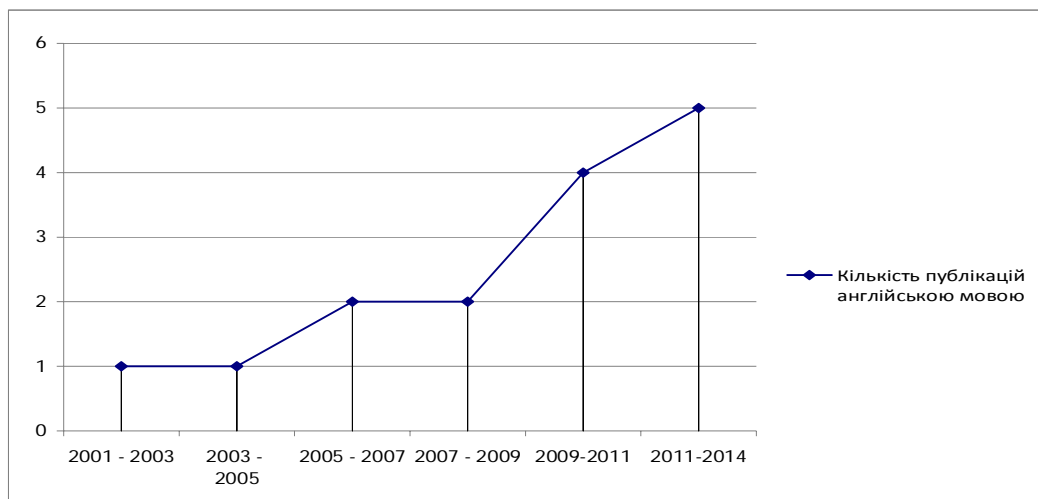


Рис. 1. Класифікація літературних джерел англійською мовою відповідно до року їх видання

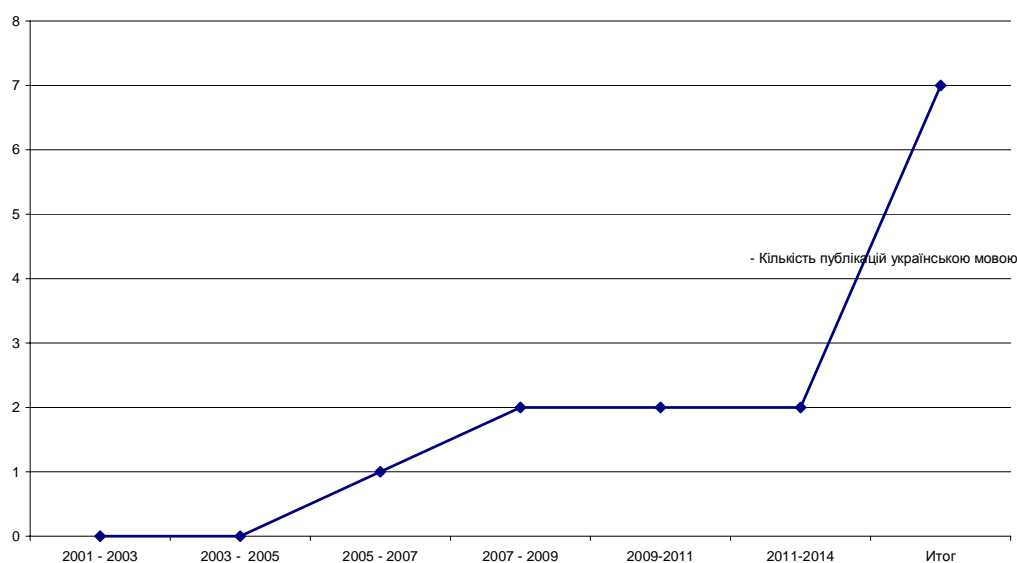


Рис. 2. Класифікація літературних джерел українською мовою відповідно до року їх видання

Згідно з графічними діаграмами період з 2007 по 2014 роки був найпродуктивнішим з погляду дослідження літературних джерел щодо екосистем навчального програмного забезпечення. У попередні роки тільки починали вивчати екосистеми програмного забезпечення, в Україні відомо праці М.О.Сидорова, доктора технічних наук, завідувача кафедри інженерії програмного забезпечення Національного авіаційного університету «Екологія програмного забезпечення», де представлено застосування екологічного підходу до дослідження програмного забезпечення; розглядаються основні положення екології програмного забезпечення, як частини інженерії програмного забезпечення; наводяться три напрямки екологічних досліджень програмного забезпечення («зелене» програмне забезпечення, сталий розвиток, цифрові екосистеми); результати демонструються на прикладі авіаційного тренажера[2].

Досліджував екосистеми програмного забезпечення О.О. Грінченко, у статті «Моніторинг екосистем програмного забезпечення», де розглядалося застосування екологічного підходу до досліджень програмного забезпечення та формуються основні поняття розділу інженерії програмного забезпечення – екології програмного забезпечення. ISSN 2306-4455. Вісник ЧДТУ, 2014, №2

Розглянув використання концепцій екосистем виробниками та дослідниками програмного забезпечення та запропонував моделі екосистем В.А. Хоменко у статті «Екосистеми програмного забезпечення» Вісник Національного технічного університету «ХПІ». Серія: Нові рішення у сучасних технологіях 2011рік.

Оскільки інформації для проведення дослідження вкрай обмаль сформулюємо основні дослідницькі питання двома мовами англійською та українською і продемонструємо результати проведеного дослідження.

(П1) Які екосистеми навчального програмного забезпечення існують та успішно функціонують на даний час? What ecosystems of educational software exist and are successfully functioning at the present?

Основними складовими екосистеми навчального програмного забезпечення, є саме програмне забезпечення, середовище його розробки та навчальна установа де проходить реалізація, або процес втілення його на практиці. На даний час інформації у українських електронних наукових бібліотеках про екосистеми навчального програмного забезпечення дуже мало. Під час підбору літературних джерел було знайдено матеріали за ключовими словами близько 40 300 українських джерел та 72 600 англійських джерел, що виявилось не зовсім те чого очікували. Література висвітлювалась абсолютно вся, відфільтровувати довелося довго : «What ecosystems of educational software».

Таблиця 2. Висвітлення літератури за першим запитанням

by teaching: A new agent paradigm for educational ... <i>Biswas</i>	Представлено у 274 електронних джерелах
Agile software development ecosystems - <i>Highsmith</i>	Представлено у 888 електронних джерелах
An e-learning ecosystem based on cloud computing	Представлено у 127 електронних джерелах

(П2) Якими є основні складові екосистеми навчального програмного забезпечення? / What are the main elements of educational software ecosystem?

Перші матеріали були опубліковані в матеріалах конференції «Програмне забезпечення та інженерні програми (SEA '12): Три точки моделі програмного забезпечення екосистем. Джон Макгрегор, і Йейтс Monteith представив Три точки зору моделі програмного забезпечення екосистем до IASTED в програмне забезпечення інженерних додатках конференції. Він був представлений Yates Монтейт 14 листопада 2012. Ключовими словами є - розробка програмного забезпечення, програмне забезпечення екосистеми, мета-моделювання, лінії програмного продукту. Компанія, яка розробляє навчальне програмне забезпечення бере участь в мережі зв'язків з іншими організаціями, які представляють необхідну інформацію, щодо поставлених завдань. Керівники груп, які приймають основні рішення повинні чітко розуміти ці відносини, щоб приймати оптимальні рішення. Моделювання підхід, представлений, створює три точки зору на мережі для розділення проблем і зменшують складність для осіб, що до прийняття рішення. Приклад ілюструє ефективність трьох точок зору і показує результати застосування аналізу в контексті цих поглядів. Навесні 2013 року, Джон Макгрегор і Йейтс Monteith співпрацював з Сімоні Аморі і Едуардо Алмейда Університету (UFBA) в Бразилії. Разом представили презентацію моделювання в архітектурі програмного забезпечення екосистеми в SEI Архітектура технології мережевого користувача конференції. Презентував матеріал Джоном Мак-Грегор під час конференції, 29 квітня - 3 травня. САТУРН 2013: Моделювання архітектури програмного забезпечення екосистеми навчального програмного забезпечення.

(П3) Які існують зауваження щодо структури екосистем навчального програмного забезпечення?/ What remarks are there regarding the structure of educational software ecosystems?

Програмне забезпечення навчальних екосистем програмного забезпечення має бути доступним і зрозумілим у використанні. Користувачі бажають та заслуговують на гарантії якості представлені програмного забезпечення. Використання програмного забезпечення несе в собі низку зауважень, та пропозицій, щодо тих чи інших питань, які виникають у процесі впровадження супроводу та всього життєвого циклу. Середовище існування програмного забезпечення повинно гармонійно співіснувати із соціумом.

Розробка програмного забезпечення і його підтримка вимагає багато навиків, у тому числі дизайн, управління, програмування, аналіз, оцінювання, документація, системна інтеграція, а також дизайн для безпеки і надійності. Можливість внесення змін у існуючу структуру екосистеми навчального програмного забезпечення, має велике значення.

Таблиця 3. Основні літературні джерела щодо екосистем навчального програмного забезпечення

	Ключові запитання на англійській мові для пошуку літературних джерел	Матеріали пошуку
1	What ecosystems of educational software exist and are successfully functioning at the present?	1,4,7,8,10,12
2	What are the main elements of educational software ecosystem?	2,3,5,8,14
3	What remarks are there regarding the structure of educational software ecosystems?	1,4,7,9,11,13
4	What shortcomings have been exposed in educational software ecosystems?	-----
5	How can the current defects be eliminated and what are the ways to improve educational software ecosystems?	1,3,5,6,8,15,16

Проведене нами дослідження є першим в галузі екосистем навчального програмного забезпечення та практичне впровадження. Проведення подальших вторинних досліджень та літературних оглядів є необхідними для майбутнього розвитку екосистем в цілому.

Основними кроками та процедурами для вдосконалення екосистеми навчального програмного забезпечення можуть стати наступні кроки:

- Правильний підбір літературних джерел в тій чи іншій галузі освіти.
- Модернізація навчального процесу та плану з використанням новітніх методик навчання і додаванням більшого об'єму практичних проєктів.
- Систематична інтеграція прикладних та експериментальних досліджень в навчальний процес.
- Укріплення та посилення взаємодії освітніх закладів та професійних організацій, які є розробниками у галузі програмного забезпечення.
- Постійне оновлення матеріальної бази, програмного забезпечення.

(П4) Які недоліками виявлені у екосистемах навчального програмного забезпечення?/ What shortcomings have been exposed in educational software ecosystems?

Дане питання виявилось надскладним для пошукових систем, чи через нестачу інформації, чи через складний підбір ключових слів.

(П5) Які можливі вирішення існуючих проблем та шляхи покращення екосистем навчального програмного забезпечення?

Приблизна кількість інформації українською мовою яка була представлена 46 000 за 0,57 секунд. Лише 2 електронних джерела відповідали темі і містили потрібну інформацію: Н.М.Сидорова., «Навчання інженерії програмного забезпечення – систематичний огляд літератури», №2(6)2011.

Н.М. Сидорова «Формування готовності майбутніх бакалаврів з інженерії програмного забезпечення до професійної комунікації» 2012 рік.

Результат проведеного дослідження англійських літературних джерел за (П5) такий: 24 000 000 за 0,7 секунд. Кількість видань 127, цитовано у 226та доступно для скачування лише 5.

Результат проведеного дослідження за 5 ключовими запитаннями англійською мовою такий:

1. Andreas Meiszner Open Education Ecosystems, learning analytics and supportive software system framework (67 SlideShares), Founder & Managing Partner at SCIO – Sociedade do Conhecimento, Inovação, e Organização LDA, Published on May 30, 2012
2. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: A research agenda for software ecosystems. In: *International Conference on Software Engineering. ICSE-Companion 2009*, Vancouver, British Columbia, Canada (2009)
3. Manikas, K., Hansen, K.M.: Software ecosystems—a systematic literature review. *Journal of Systems and Software* 86, 71–80 (2012)
4. Weiblen, T., Giessmann, A., Bonakdar, A., Eisert, U.: Leveraging the software ecosystem: Towards a business model framework for marketplaces. In: *3rd International Conference on Data Communication Networking, DCNET 2012, 7th International Conference on e-Business, ICE-B 2012 and 3rd International Conference* .
5. Dhungana, Deepak; Groher, Iris; Schludermann, Elisabeth; Biffel, Stefan. Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature., 2010 URI:<http://hdl.handle.net/10344/2359>, Publisher:Association for Computing Machinery,Publication type:info:eu-repo/semantics/conferenceObject.
6. A Survey of Associate Models used within Large Software Ecosystems, Joey van Angeren, Jaap Kabbedijk, Slinger Jansen, and Karl Michael Popp, Department of Information and Computing Sciences, Utrecht University Princetonplein 5, 3508 TB Utrecht, the Netherlands. vanangeren.j.kabbedijk.s.jansen@cs.uu.nl SAP AG, Corporate Development Dietmar-Hop-Allee 16, 69190 Walldorf, Germany, karl.michael.popp@sap.com
7. Waters, Sandie H.; Gibbons, Andrew S. (2004). *Design Languages, Notation Systems, and Instructional Technology: A Case Study* Educational Technology Research and Development, 52, 2.
8. Watkins, Debbie; Kritsonis, William Allan (2008). *Aristotle, Philosophy, and the "Ways of Knowing Through the Realms of Meaning": A National Study on Integrating a Postmodernist Approach to Education and Student Achievement* [Online Submission]
9. M. Lehman. Software Evolution – Background, Theory, Practice. *Integrated Design and Process Technology*. Society for Design and Process Science. – 2003 – 11p.
10. Aid to recovery: the economic impact of IT, software, and the Microsoft ecosystem on the global economy. – IDC White Paper, – 2009, 9p.
11. Cynthia Keeshan. The Software Ecosystem. - <http://www.microsoft.com/canada/media/ecosystem.msp> ,
12. David G. Messerschmitt and Clemens Szyperski. *Software Ecosystem: Understanding an Indispensable Technology and Industry*. Cambridge, MA, USA: MIT Press. ISBN 0262134322 – 2003.
13. GEORGE G. MITCHELL and JAMES DECLAN DELANEY. An Assessment Strategy to Determine Learning Outcomes in a Software Engineering Problem-based Learning Course. *Int. J. Engng Ed.* Vol. 20, No. 3, pp. 494-502, 2004 Printed in Great Britain.
14. Ghazi Alkhatib, Ghassan Issa, Aiman Turani, M. Ibrahim Zaroor. *Incorporating Innovative Practices in Software Engineering Education*. April 04 - 06, 2011, IEEE DUCON Education Engineering 2011 - Learning Environments and Ecosystems in Engineering Education. Amman, Jordan.
15. Olavo Barbosa, Carina Alves. A systematic mapping study on software ecosystems.
16. *Proceedings of the Third International Workshop on Software Ecosystems – IWSECO-2011*, Brussels, Belgium. – June 7th, 2011. – Vol.746. – pp. 15 – 26.

Висновок. В результаті проведених досліджень було створено підбір літературних джерел щодо екосистем навчального програмного забезпечення, за певними критеріями. Питання за якими проводився підбір містили загальні терміни. Проведений аналіз навчальної, монографічної та наукової літератури показав, що літературних джерел англійською мовою значно більше ніж українською. Потрібно вносити значні зміни, у використання навчального програмного забезпечення. Досліджувати використання, засвоєння, програмного забезпечення у вищих навчальних закладах. Аналізувати отримані знання та навички на практиці.

1. Сидоров. М.О. Экология программного обеспечения. – Материалы Всеукраинской конференции аспирантов и студентов «Инженерия программного обеспечения 2006» – К.: НАУ, 2006.
2. Сидоров М.О. Программное обеспечение – экологический подход к исследованиям. // *Инженерия программного обеспечения*. – 2010. – №1. – С. 5-13.
3. Сидорова Н.Н. Навчання інженерії програмного забезпечення – системний огляд літератури.- *Инженерия программного обеспечения*, №2(6), 2011, Київ: Національний авіаційний університет, с. 56-67.
4. Dyba, T. Dingsoyr, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. – 2008, vol. 50, pp. 833–859.
5. Petersen K., Feldt R., Maltsson M. Systematic Mapping Studies in Software Engineering: In *12 th International Conference on Evaluation and Assessment in Software Engineering*, 2008. – P. 378-390.

6. SE2004. Curriculum Guidelines for Undergraduate Degree programs in SE. A volume of the Computing Curricula Series. August 23, 2004.- p.
7. Kitchenham, B., Brereton P., Budgen, D., Turner M., Bailey J., Linkman, S. Systematic literature reviews in software engineering - A systematic literature review, Information and Software Technology, vol. 51, Issue 1. (2009)
8. Kitchenham, B. Guidelines for performing Systematic Literature Reviews in Software Engineering. vol 2. 3 EBSE Technical Report. (2007)