

Н. Б. Шаховська, В. А. Висоцька, Л. В. Чирун  
Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## МЕТОДИ ТА ЗАСОБИ ДИСТАНЦІЙНОЇ ОСВІТИ ДЛЯ ЗАОХОЧЕННЯ І ЗАЛУЧЕННЯ СУЧАСНОЇ МОЛОДІ ДО САМОСТІЙНИХ НАУКОВИХ ДОСЛІДЖЕНЬ

© Шаховська Н. Б., Висоцька В. А., Чирун Л. В., 2015

Нові інформаційні телекомунікаційні технології сприяють оптимізації управління навчальним процесом. Запропоновано інноваційні підходи до поліпшення навчальної програми вищої освіти у сфері ІТ. Розглянуто методи пошуку і залучення студентів до наукової та практичної роботи завдяки їх участі в командних змаганнях і спільних міжвузівських науково-практичних проектах. Узагальнено і подано методологічні наукові результати з проблематики впровадження технологій та методів дистанційного навчання. Запропоновано метод покращення наукової грамотності студентів, знаходячи інноваційні рішення для вирішення проблем проектів, що сприятиме науковій кар'єрі для учасників міжвузівських проектів. Розглянуто питання побудови комплексу математичних моделей процесів дистанційного навчання, на яких ґрунтується створення інтегрованого інформаційно-навчального середовища. Зазначені питання є актуальними у зв'язку з упровадженням технологій дистанційного навчання і нерозривно пов'язано з дидактичними та методологічними аспектами навчального процесу.

**Ключові слова:** інформаційні технології, інтелектуальний аналіз даних, міжвузівські науково-практичні проекти, дистанційне навчання, дистанційна освіта, електронне навчання, інформаційно-комунікаційні технології, електронний підручник, віртуальне навчальне середовище, інтегроване інформаційно-навчальне середовище, наукова кар'єра, учасники міжвузівських проектів.

New information, telecommunication technologies contribute to the optimization in the management of studies. This paper is devoted to the implementation of innovative approaches to improving the curriculum of higher education. The method of finding and attracting students including girls for scientific and practical work through and their participation at team competitions and joint Interuniversity scientific-practical projects are proposed. This work has considered a problem of distance education and involvement in her adaptive learning system. Improving scientific literacy of students by finding innovative solutions to address the problems of projects are proposed. Simplify the process of obtaining a scientific career for interuniversity projects participants. The questions of mathematical models of processes of distance education (remote training) are highlighted in this article. Creation of the integrated net oriented informational-educational environment is based on them. The indicated questions are actual in connection with implantation of technologies of distance learning and inextricably related with didactic and methodological aspects of the educational process.

**Key words:** information technology, data mining, Interuniversity scientific-practical projects, distance learning, distance education, e-learning, information and communications technology, electronic textbook, virtual learning environment, integrated net oriented informational-educational environment, scientific career, interuniversity projects participants.

### Вступ. Загальна постановка проблеми

Відомими актуальними проблемами сьогодення є заохочення і залучення сучасної молоді до проведення самостійних наукових досліджень, виховання серед них висококваліфікованих

спеціалістів та лідерів професійної діяльності [1–10], а також використання інноваційних інформаційних технологій для дистанційного навчання перспективних спеціальностей, наприклад, систем штучного інтелекту [23]. Поширеною проблемою є небажання молодих людей підвищувати рівень своєї освіти та займатися науково-дослідною діяльністю в аспірантурі [23]. Освіта – це цілеспрямована пізнавальна діяльність людей з отримання знань, умінь та навичок або щодо їх вдосконалення [1, 23–26]. Аспірантура є формою підготовки науково-педагогічних і наукових кадрів для вищих навчальних закладів і науково-дослідних установ. Процес і результат засвоєння особистістю певної системи наукових знань, практичних умінь та навичок і пов'язаного з ними того чи іншого рівня розвитку її розумово-пізнавальної і творчої діяльності, а також морально-естетичної культури визначають соціальне обличчя та індивідуальну своєрідність цієї особистості. Відсутність освіти та зайнятості серед молодих людей призводить до повільної деградації суспільства.

### **Зв'язок висвітленої проблеми із важливими науковими та практичними завданнями**

Призначення освіти – дати новому поколінню початкові знання культури, формуючи поведінку у дорослому житті і допомагаючи у виборі можливої ролі в суспільстві [23]. У широкому сенсі слова, освіта – процес або продукт формування розуму, характеру або фізичних здібностей особистості. У технічному сенсі утворення – це процес, за допомогою якого суспільство через школи, коледжі, університети та інші інститути цілеспрямовано передає свою культурну спадщину – накопичене знання, цінності та навички – від одного покоління іншому. В останні десятиліття набула поширення наукова експертиза у сфері освіти, яка здійснюється у формах педагогічного аудиту, психолого-педагогічної експертизи, комплексної гуманітарної експертизи тощо. Але відсутні дослідження мотивування молоді до навчання. У повсякденному розумінні освіта, крім усього іншого, передбачає навчання учнів учителем. Це є основним недоліком навчання. Викладачі з вузьких сфер, таких як системи штучного інтелекту, можуть навчати тільки цьому предмету, зазвичай в університетах та інших вишах. І це значно обмежує доступ ширшої аудиторії до навчання. Існує також викладання фахових навичок, наприклад, програмування та проектування відповідних систем. Крім освіти, в спеціальних установах існує також самоосвіта, наприклад, через Інтернет, читання, відвідування музеїв або особистий досвід. Але воно є неконтрольованим або регіонально спрямованим.

### **Аналіз останніх досліджень та публікацій**

Сучасний процес переходу до інформаційного суспільства, а також соціально-економічні зміни, що відбуваються в Європі, вимагають суттєвих змін у багатьох сферах діяльності держави [23]. Насамперед це стосується реформування галузі освіти та інновацій у забезпеченні навчального процесу. Тенденція використання інформаційних та телекомунікаційних технологій полягає в створенні систем, які здатні не тільки виконувати інформаційно-аналітичні функції, але й створювати умови для оперативного керівництва розподіленим дистанційним навчальним процесом, бути ефективним середовищем організації й управління процесом навчання та укладання ефективного забезпечення навчального процесу. Це дає можливість зробити універсальним процес планування навчального процесу та підготовки навчально-методичних матеріалів для різних форм навчання та різних структур навчальних програм. Використання інформаційних комунікаційних технологій повинно оптимізувати процеси управління навчальним процесом і створити ефективну організаційну структуру. Забезпечення навчального процесу за допомогою технічних, інформаційних, інтелектуальних та програмних інновацій надзвичайно важливо в контексті управління сучасним розподіленим навчальним процесом [1, 23].

Переваги дистанційного навчання (ДН) з переходом на нові інформаційні технології очевидні [1]. Крім того, в публікаціях щодо проблем ДН значну увагу звертають на розроблення електронних посібників, вибір платформ, дидактику нових освітніх середовищ, методологію проведення навчання та ін. Проте питання, пов'язані із математичним моделюванням процесів ДН, залишаються нерозглянутими. Тенденція використання інформаційних і телекомунікаційних

технологій сьогодні полягає в створенні систем, які ґрунтуються на ефективних математичних моделях, здатних не тільки забезпечувати виконання інформаційно-аналітичних функцій, але і створювати умови для оперативного керування розподіленим дистанційним навчальним процесом, бути ефективним середовищем організації й управління процесом навчання та укладання ефективного особистісно-орієнтованого навчального плану та розкладу [18]. Це дає змогу досягти певної універсальності процесів планування навчального процесу та підготовки навчально-методичних матеріалів для різних форм навчання і різних структур навчальних програм.

Отже, проблема проектування інтегрованого інформаційно-навчального середовища систем ДН, що спирається на сучасні досягнення в області дидактики, ергономіки і математичного моделювання, набуває ще більшої актуальності. Запропонований у статті комплекс математичних моделей містить структурну модель дистанційних курсів, модель знань дистантного слухача та процесу навчання. Слід також зазначити, що крім комплексу математичних моделей, інтегроване середовище містить засоби добору контрольних питань, визначення дидактичних характеристик завдань, створення сценарію навчання [19].

Проектування та моделювання систем дистанційного навчання і діагностики якості знань для мережі Internet дає змогу автоматизувати не тільки процес подання навчального матеріалу, але й індивідуальну навчальну роботу студентів, оскільки в основу таких систем покладено розподілену модель збереження інформації. Своєю чергою, використання єдиних форматів передавання інформації в мережі Internet дасть змогу розширити сферу застосування автоматизованих навчальних і діагностуючих систем у різних навчальних закладах.

Однак слід зазначити, що наявні сьогодні інструментальні засоби (PHOENIX Web, Quest Net+, ToolBook тощо) дають змогу лише розробляти навчальний матеріал для автоматизованих систем дистанційного навчання і діагностики, а при проектуванні компонентів підсистем діагностики необхідний зв'язок прямого програмування спеціалізованими мовами. Такі системи не ґрунтуються на накопичених у результаті багаторічних досліджень у галузі штучного інтелекту знаннях про методи проектування, реалізації, оцінювання якості й експлуатації інтелектуальних комп'ютерних навчальних систем [21]. Розроблювачі цих систем набагато більше займалися питаннями побудови інтерфейсу дистантного слухача, ніж вибором адекватної стратегії навчання та його індивідуалізації [15]. Отже, у сформованих умовах актуальним є розроблення засобів, що дають змогу автоматизувати проектування основних компонентів процесу дистанційного навчання і діагностики якості знань. Проектовані компоненти повинні вирішувати не тільки завдання підготовки та подання навчальної інформації, але й проблеми організації дистанційних адаптивних процедур навчання, діагностики знань, збирання й опрацювання результатів контролю та формування статистичних даних про дистантного слухача. Зазначені проблеми частково розглядаються в [13, 14, 19].

### **Виділення проблем**

Складність потоків даних, обсяги інформації, що передається, в інформаційній інтелектуальній системі ДН вимагає введення в її структуру допоміжних засобів інтелектуальної навігації, опрацювання інформації, каталогізації та планування. Більшість задач адміністрування і управління структурною одиницею покладаються на агента-секретаря, якого можна вважати інтелектуальним ядром структурної одиниці навчального закладу. Це не просто утиліта, що спрощує процес інтерактивного діалогу із системою, але й активний учасник її діяльності: інтерфейсний агент, що має індивідуальність і виконує вказівки користувача з питань планування, комунікації і управління.

Виникають завдання, які необхідно першочергово вирішити:

1. Зміцнення міжнародної міждисциплінарної та міжгалузевої мобільності дослідників:
  - підвищення можливості для кар'єрного росту й умов праці молодих науковців та досвідчених дослідників;
  - розвиток наукової кар'єри;
  - сприяння співробітництву між науковими установами та промисловістю.

## 2. Розвиток дослідницьких інфраструктур.

Необхідно забезпечити:

1. Підтримку найталановитіших та креативних молодих людей та їх команд щодо проведення передових досліджень у сфері інформаційних технологій та способів надання дистанційної освіти;
2. Наукову міжнародну співпрацю з метою пошуку нових та перспективних областей досліджень через підтримку сучасних інноваційних інформаційних технологій;
3. Надання молоді можливостей для навчання та розвитку професійної кар'єри;
4. Доступ всіх молодих учасників проекту до висококласних дослідницьких інфраструктур.

### Формулювання мети

Актуальним є завдання для всіх верств населення та країн світу заохочувати та мотивувати студентів до навчання в аспірантурі та проведення науково-дослідної діяльності. Якщо кваліфікація потенційних працівників знижується, то це призводить до зниження матеріального виробництва та деградації соціальних інститутів. Відсутність молодих людей, які хочуть брати участь у науково-дослідній діяльності, гальмує науково-технічний прогрес. Саме тому є необхідність у розробленні інноваційних засобів, наприклад, через такий поширений серед молоді процес, як гра. Використовуючи різні навчальні ігри як заохочення студентів до навчання та проходження певних етапів або рівнів, спонукати їх до самовдосконалення через активну участь у вдосконаленні цих ігор, спрямовану на науково-дослідну діяльність. Для підвищення професійного міждисциплінарного знання та набуття навичок у галузі інформаційних технологій необхідно розробляти та впроваджувати інформаційні програмні засоби, які безпосередньо зацікавлять учасників навчання та сприятимуть зростанню їх мотивації для активізації власних досліджень. Метою є створити багатомірний освітній інструмент, який стимулюватиме студентів для пізнання і дослідження.

Завдання:

1. Впровадження інноваційних підходів до поліпшення навчальної програми вищої освіти через гру з проходженням трьох рівнів складності з безпосередньою участю самих студентів у формуванні процесу навчання.
2. Пошук та залучення до наукової та практичної роботи студентів через командні змагання та їх участь у спільних міжвузівських науково-практичних проектах.
3. Підвищення наукової грамотності студентів безпосереднім залученням їх до удосконалення процесу викладання перспективних дисциплін, знаходячи інноваційні рішення для вирішення проблем проектів.
4. Спрощення процесу отримання наукової освіти та спеціальності для учасників міжвузівських та міжнародних проектів.

### Аналіз отриманих наукових результатів

Досягнення цілей:

1. *Розробити інноваційну програму, впроваджуючи нові інтегровані курси, які будуть викладатися студентам у різних університетах консорціуму.* Так викладачі можуть поділитися дослідженнями і професійним досвідом. Студенти  $S = \{s_1, s_2, \dots, s_{n_s}\}$  виконуватимуть спільні проекти і отримуватимуть досвід справпраці в міжвузівській та міжнародній командах (відсутність кордонів, проблем мови та раси, статі та віку). Програма навчання має містити предмети з класичної школи (I рівень навчання). Це дасть змогу підвищити рівень професійних та практичних навичок студентів і скоротити кількість випробувань протягом семестру. Після стадії моніторингу студента залучають до вивчення низки предметів залежно від складності сцени та специфіки компетентності самого студента (II–III рівень навчання). Курс організовується у вигляді гри, участь в якій сприяє отриманню знань з предметної області, із задоволенням підтримувати спілкування в команді, проявляти організаторські та лідерські здібності, підвищувати комунікабельність, навчатися працювати в команді, підтримувати азартність в отриманні знань, вдосконалюватися в професійній технічній англійській мові та підвищити рівень конкурентоспроможності в професійній сфері.

Навчальна програма повинна містити предмети з класичної ІТ школи [1]. Це дасть змогу поліпшити професійні та практичні навички студентів та зменшити кількість предметів в семестрі. Після стадії моніторингу студенти поступають на рівень залежно від складності курсу та його конкретної компетенції (рівень II-III). Курс організований у вигляді порталу, де навчання студентів реалізоване у формі ігор, що допомагає учасникам:

- отримати знання з предметної області;
- приємно провести час;
- підтримувати зв'язок у команді;
- продемонструвати організаційні та лідерські навички;
- набувати навичок;
- працювати в команді;
- отримувати нові знання;
- спілкуватися діловою англійською;
- підвищувати конкурентоспроможність у своїй галузі знань.

На рис. 1 наведено структуру вивчення навчальних курсів у вигляді мережі Петрі, де вершини мережі визначають курси, а структура мережі – послідовність їх вивчення. Згідно з її структурою курси, що знаходяться у вершинах з номерами 2, 3 та 4, студент може вивчати паралельно після вивчення курсу 1. Предмет 5 можна вивчати лише після успішного складання іспиту з предметів 2 та 3, а 6-й – 4-го та 5-го предметів.

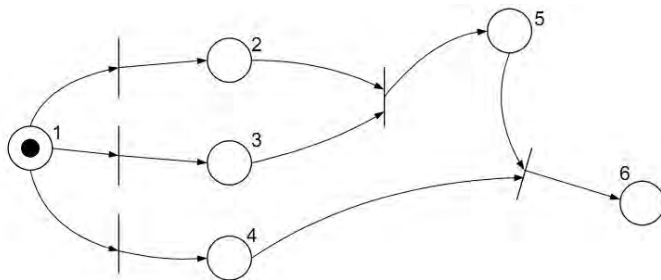


Рис. 1. Приклад подання структури дистанційних навчальних курсів за допомогою мережі Петрі

Місцезнаходження фішки в мережі показує множину курсів, які вивчає студент у поточний момент. Детальніше розглянемо процес дистанційного навчання окремого курсу, який задається вершиною мережі Петрі. Структура навчальних курсів спирається не лише на лекційний матеріал і практичні заняття. Однією зі значимих складових структури навчального курсу є індивідуальна робота студентів (ІРС). За положеннями Міністерства освіти та науки України, на індивідуальну роботу студентів відведено від 17 до 34 годин навчального часу, а на лекційний матеріал – від 51 до 68 таких годин. Отже, використання мережі Internet як основи для проектування автоматизованих систем навчання і діагностики дає змогу побудувати нову модель надання навчальних знань для автоматизованих освітніх систем.

Наприклад, якщо структуру лекційного матеріалу називати ядром знань (основні чи базові знання), то навчальний матеріал, що не ввійшов у курс лекцій, але такий, що стосується тем, досліджуваних у межах певної дисципліни, можна назвати додатковими знаннями. Уся сукупність додаткових знань утворить оточення ядра. Оточення ядра подає той матеріал, який необхідно вивчити студенту під час виконання ІРС. Ядро знань разом з оточенням ядра формують макромодель усього навчального матеріалу у межах розглянутої дисципліни.

Кожен компонент оточення ядра можна розглядати як зв'язний граф. Однак з огляду на те, що для викладача – розроблювача навчального курсу не важлива структура “чужого навчального матеріалу”, а важливі зв'язки, встановлювані між ядром знань та його оточенням, а також те, що для полегшення автоматизованого аналізу й опрацювання структури макромоделі пропонується

останню розглядати у вигляді семантичної мережі  $V = (Z, F)$ , де  $Z = \{z_1, z_2, \dots, z_n\}$  – множина об'єктів семантичної мережі;  $F$  – відображення  $Z \rightarrow Z'$ , зважене відношеннями між об'єктами семантичної мережі. У цій семантичній мережі кожний компонент оточення ядра подано у вигляді однієї вершини, що з'єднана з ядром деякою кількістю встановлених викладачем зв'язків.

Процес оцінювання знань дистантного слухача (студента)  $s_i$  розглянемо як процес встановлення зв'язків між студентом і викладачем. Під процесом оцінювання знань розуміють взаємодію викладача і студента, що містить дві фази:

- подання моделі знань студентом для оцінювання їх викладачем у вигляді виконаного тесту (мета – побудова моделі знань студента на основі результатів виконання запропонованого тесту);
- сприйняття викладачем цієї моделі і порівняння її з моделлю, складеною на підставі вимог до знань студента і власних уявлень викладача про якість моделі (мета – порівняння моделі знань студента з моделлю, складеною викладачем на основі своїх уявлень про знання студента, а також на результатах тесту, мети тестування і вимогах до засвоєння вивченої області знань, оснований на його особистому досвіді і нормах оцінювання).

Систему оцінювання знань студентів як систему керування подамо у вигляді кортежу

$$O = \langle V_p, V_s, I, f, j \rangle,$$

де  $V_p$  – модель необхідних знань, подана у вигляді семантичної мережі;  $V_s$  – модель знань студента, інтерпретована викладачем;  $I$  – оцінювання знань;  $f$  – алгоритм керування;  $j$  – мета тестування.

Керування оцінюванням знань як сприйняття моделі знань, поданої студентами для оцінювання, і порівняння її з деякою моделлю знань, побудованою на основі власних уявлень викладача про те, якими знаннями має володіти студент, реалізується алгоритмом 1.

**Алгоритм 1.** Керування оцінюванням знань студента-дистанта.

**Етап 1.** Будується модель  $f(s_i)$  знань студента як результати виконання завдань тесту.

**Етап 2.** Будується модель знань студента  $V_s$  на основі сприйняття викладачем моделі  $f(s_i)$ .

**Етап 3.** Визначається модель  $V_p$ .

**Етап 4.** Порівнюються моделі  $V_s$  та  $V_p$ .

Задача оцінювання знань (порівняння семантичних мереж) формулюється так. Створюється тест, запитання якого належать до одного з обраних розділів. Кожне з цих запитань припускає вибір одного з декількох варіантів відповідей і кодується 0, якщо відповідь обрано неправильно, чи 1, якщо правильно. Зауважимо, що будь-яке складне запитання можна подати у вигляді кінцевої послідовності з  $M$  елементарних висловлень, відповідями на які є бінарні змінні  $x_1, x_2, \dots, x_{n_i}$ . При цьому значимість коефіцієнтів  $I_1, I_2, \dots, I_M$  змінних  $x_1, x_2, \dots, x_{n_i}$  може бути різною.

Інформація про величину коефіцієнтів  $I_1, I_2, \dots, I_M$  важлива для вибору методу ухвалення рішення. Якщо усі відповіді мали б однакову значимість, тобто  $I_1 = I_2 = \dots = I_M$ , або ж значимість кожного з них була б відомою, можна було б застосувати широковідомий метод підсумовування балів. Однак такий підхід вимагає попереднього вивчення коефіцієнтів  $I_1, I_2, \dots, I_M$  і відповідних порогів  $c_1, c_2, \dots, c_e$  прийняття  $e=4$  рішень (балів в межах [2,5]) на доволі представницькому класифікованому матеріалі. Збирати і об'єктивно аналізувати дані складно.

Об'єктивність прийнятого рішення в межах цього методу залежить від якості оцінок коефіцієнтів  $I_1, I_2, \dots, I_M$ . Незважаючи на ці обмеження, завдяки своїй простій реалізації цей метод є найпоширенішим для оцінювання знань.

Використання штучних нейронних мереж дає змогу розв'язати задачу без явного визначення оцінок коефіцієнтів  $I_1, I_2, \dots, I_M$  і порогів  $c_1, c_2, \dots, c_e$ . У відомих підходах попередньо задається архітектура нейронної мережі (як правило, вибирається багат шаровий перцептрон із заданою кількістю нейронів і шарів). Потім формується навчальна вибірка  $X$  та  $Y$

$$X = (x^{(1)}, x^{(2)}, \dots, x^{(n)})^T, \quad Y = (y_1, y_2, \dots, y_n)^T,$$

складена з  $n$  прикладів рішень  $y_i = \{2,3,4,5\}$ ,  $i = \overline{1, n}$ , що належать до оцінки знань  $i$ -го студента за тестом  $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)})^T$ .

Кожний з цих прикладів ретельно аналізується для того, щоб надалі виключити необ'єктивні рішення. Для навчання нейронних мереж найчастіше використовуються методи зворотного поширення помилки. Для успішної реалізації цих методів необхідні сотні навчальних прикладів. Алгоритми цього типу не завжди приводять до бажаного результату, якщо навчальну вибірку складено усього з декількох десятків прикладів.

Перераховані недоліки у меншому ступені властиві нейронним мережам, структура і параметри яких визначаються в процесі навчання методом самоорганізації А.Г. Івахненко. Однак результати самоорганізації залишаються залежними від деяких умов, що задаються ззовні. Якщо семантична мережа задає лише відношення між розділами та темами курсу, то математичну модель такої мережі можна задати у вигляді матриці:  $A = \|a_{ij}\|$ . Матриця  $\|A\|$  симетрична щодо головної діагоналі і характеризує наявність зв'язку двох окремо взятих розділів, зокрема:

$$a_{ij} = \begin{cases} s_{ij}, & \text{якщо } i \text{ розділ пов'язаний з } j, \\ 0, & \text{інакше.} \end{cases}$$

Потім для розв'язання задачі побудови адаптивної системи навчання, основаної на формуванні системи в автоматичному режимі, а також враховуючи результати діагностики і нові структури вивчення матеріалу, розглянемо матрицю відносних оцінок рівня знань розділів  $C = \|c_{ij}\|$ :

$$c_{ij} = \begin{cases} a_{ij} - q_i, & \text{якщо } a_{ij} \neq 0 \wedge a_{ij} - q_i > 0, \\ 0, & \text{якщо } a_{ij} = 0 \wedge a_{ij} - q_i \leq 0. \end{cases}$$

Вага зв'язку між двома розділами визначається за  $p$ -бальною шкалою, де  $p$  задає викладач. Разом з цим заповнювати матрицю  $\|A\|$  необхідно, дотримуючись такої умови:

$$\sum_{j=1}^{i-1} a_{ij} = p.$$

Ця умова показує, що сумарна залежність будь-якого розділу від інших, пов'язаних з ним, не може перевищувати максимальної бальної оцінки самого розділу. Очевидно, що максимальна бальна оцінка для будь-якого розділу ядра знань та його оточення однакова, тобто залежить від прийнятої шкали і дорівнює  $p$ . Крім того, необхідно зазначити, що рівень вивчення розділу ( $R$ ) визначається так само, як і значимість зв'язку, за  $p$ -бальною шкалою. Ця умова необхідна для того, щоб система могла самостійно формувати плани вивчення матеріалу. Задача формування плану вивчення матеріалу належить до класу задач лінійного цілочислового (булевого) програмування.

Результатом розв'язання цієї задачі є складання оптимального плану вивчення матеріалу, що враховує не тільки рівень вивчення розділів  $R_i$ , але і відносні оцінки.

При цьому  $x_{ij} = 1$ , якщо перед  $i$ -м розділом включається  $j$ -й розділ, що має нижчий порядковий номер;  $0$  – у протилежному випадку. Відносні оцінки  $c_{ij}$  дають змогу для кожного розділу виявити зв'язані попередні розділи навчального матеріалу і вибрати з них ті, незадовільні знання з яких могли б вплинути на засвоєння матеріалу розглянутого розділу. Сучасні автоматизовані системи навчання аналізують і опрацьовують тільки масив  $q_i$ . Підхід, оснований на застосуванні як абсолютних, так і відносних оцінок, допомагає студенту розібратися в першопричині його незадовільних знань з конкретного розділу.

Кожному запитанню в підсистемі діагностики якості знань ставиться у відповідність вага і форма відповіді. Якщо кількість запитань для підсумкового контролю задано як  $N$ , а розділів у темі –  $n$ , то необхідно зробити вибірку так, щоб враховувався рівень вивчення розділу. Інакше кажучи, у вибірці повинні бути присутні переважно запитання з тих розділів, за якими студент показав незадовільні знання. Для цього зважується задача цілочислового програмування, де  $N > n$ ;

$n$  – кількість розділів навчального матеріалу (вершин графу макромоделі);  $X_{\min}$  – мінімальна кількість питань (визначається викладачем), яких необхідно вибрати для одного розділу. За замовчуванням  $X_{\min} = \lceil N/n \rceil$ . Результатом розв’язання цієї задачі є значення  $X = \{x_i\}$ , що вказує, яку кількість питань для підсумкового контролю необхідно взяти з того чи іншого розділу, з огляду при цьому на рівень вивчення розділу. Крім того, вибираючи питання всередині розділу, необхідно враховувати і ту обставину, що питанню ставиться у відповідність не тільки вага, але і три форми відповіді, тобто кожна форма відповіді має свій коефіцієнт складності. Усе це необхідно передбачити, тому що в одну вибірку не повинні потрапити ті самі питання з різними формами відповіді. Для цього розглядається задача:

$$\sum_{i=1}^n R_i x_i = \min, \quad \sum_{i=1}^n x_i = N, \quad \left\lceil \frac{N}{n} \right\rceil + 1 \geq x_i \geq X_{\min}, \quad i = \overline{1, n},$$

де  $x_i$  – кількість питань, отримана на основі розв’язування задач, які необхідно взяти з 1-го розділу.

Матриця  $B = \|b_{ij}\|$  є накопичувальною матрицею, що враховує ті питання, на які було дано неправильну відповідь. Спочатку всі елементи цієї матриці дорівнюють одиниці. У випадку, якщо на питання  $b_{ij}$  дано неправильну відповідь, то відповідний елемент матриці  $\|B\|$  обнуляється, якщо ж на питання дано правильну відповідь, то відповідний стовпець у матриці  $\|B\|$  викреслюється. За бажанням викладача (для точнішого оцінювання знань студента) можна вибирати не тільки “складні питання”, але й частину “складних” і частину “простих” питань. Отже, застосування викладених вище підходів допомагає краще адаптувати систему діагностики щодо кожного студента індивідуально й автоматизувати операції, пов’язані з проведенням підсумкового контролю.

2. При вивченні інтегрованих курсів доцільно ввести підходи інноваційного навчання до формування практичних завдань і методів оцінювання, а саме:

- проведення міжкомандних змагань на першому рівні проходження курсу для формування командного духу, підвищення рівня конкурентоспроможності в професійній сфері та підвищення комунікабельності;
  - практичні форми управління, коли переможці отримують вищі бали;
  - більше залучення дівчат через обов’язковість наявності їх в команді та бонусну систему оцінювання результатів змагань (як всієї команди, так і окремих учасників через їх особистий внесок у результат) під час виконання науково-практичних проектів;
- продовжити наукові дослідження на наступних рівнях для студентів з високими балами за результатами оцінки проектів;
- використання принципу конкурентоспроможності та змагань між командами студентів для збільшення їх інтересу та мотивації до продовження до навчання;
- використання студентами або командами на другому рівні навчання засобів розроблення для самостійного створення елементів ігор та конкурсів;
- використання студентами або командами на третьому рівні навчання засобів розроблення для самостійного створення ігор та конкурсів нових курсів або вдосконалення старих;
- участь студентів старших курсів у викладанні та поясненні матеріалу студентам молодших курсів.

3. Студенти, які отримують хороші результати під час навчання, проходять підсумкове тестування, за результатами якого та результатами історії навчання формуються рекомендації щодо подальшого навчання в аспірантурі та поради обрання відповідного університетського консорціуму.

Ці цілі сприяють:

1. Залученню все більшого числа молодих людей до науки, особливо дівчат, щоб поліпшити гендерний баланс серед учених.



2. Отриманню студентами практичних навичок у процесі вивчення різних предметних областей, для підвищення рівня взаємодії між науково-дослідними інститутами та іншими промисловими установами та організаціями громадянського суспільства.

3. Кар'єрному зростанню в галузі науки завдяки міжнародній та міжвузівській співпраці між різними рівнями системи освіти.

4. Підвищенню мотивації у сфері досліджень серед молоді та покращенню рівня викладання через заохочення самих студентів взяти участь у процесі навчання та викладання молодшим учасникам курсів.

Метою є створення багатовимірного освітнього інструменту, який стимулюватиме студентів отримати знання і провести дослідження. Розроблення інноваційної програми із впровадженням нових комплексних курсів, які викладатимуть студентам в різних університетах-партнерах, тобто викладачі поділяться професійним досвідом, студенти створюватимуть спільні проекти і отримують досвід роботи з різними командами: міжнародними та міжвузівськими.

Проект реалізовано на базі інтернет-ресурсу у вигляді інформаційного порталу, що інтегрує наукові школи університетів альянсу. Цей факт підтверджує наміри розвитку наукових досліджень для студентів. Студенти ІТ-шкіл університетів матимуть можливість застосовувати цей портал для навчання та оцінювання своїх знань.

Інноваційні підходи до практичних завдань і методів оцінювання, а саме: організація командних змагань між студентами ІТ-шкіл університетів альянсу; практичні форми управління, які будуть організовані в декількох університетах, які введуть нову стратегію оцінювання результатів навчання; залучення аспірантів до пояснення навчального матеріалу в молодших студентів. Результати досліджень при проходженні II та III рівнів повинні публікуватися як матеріали міжнародних конференцій, що сприятимуть росту наукових амбіцій студентів.

Студенти можуть отримати хороші результати, пройти остаточне тестування, результат якого допомагає сформулювати пропозиції щодо навчання в аспірантурі. На відміну від таких освітніх порталів, як UDACITY, Coursera, EDX, що поєднують відеопояснення викладача, домашню роботу та тестові завдання для перевірки знань студентів, дає можливість отримати досвід наукової роботи, у пошуку нових шляхів розв'язку завдань та інтерактивної роботи між членами команди та їхніми наставниками, оскільки робота проходить у формі дискусії.

Обов'язкове залучення дівчат до команд сприятиме активізації роботи команд. На підставі гендерних досліджень, спільна робота чоловіків та жінок характеризується більшою результативністю. Діяльність жінок характеризується детальним і виваженим підходом, підвищеною увагою до дрібниць: вони значно краще сприймають і аналізують деталі, подробиці, події, вміють скрупульозно ділити ціле на частини і логічно аналізувати елементи цілого. Повнота і точність такого диференціального аналізу доповнює чоловічі риси та підходи у роботі, а саме: чоловіки краще можуть уявити подію загалом, оцінити стратегічні тенденції явища, встановити інтегрований зв'язок між частинами цілого. Ці фактори є особливо позитивними при реалізації командних рішень, а бонусна система оцінювання як всієї команди, так і окремих учасників через їх особистий внесок у результат якісно впливатиме на результати роботи, бо на психологічному рівні це створюватиме позитивну конкуренцію між ними. Головною рушійною силою в жінок, за Дж. Б. Міллер, К. Гілліган і Джордан, є снага до спілкування, взаємності, чуйності, що сприятиме зняттю напруги при конфліктних ситуаціях. Жінки мають більш рухливу мотиваційну структуру, вони можуть реагувати з великими відмінностями, і їхня реакція є яскравішою.

Інноваційні підходи до практичних завдань і методів оцінювання створення мають передбачати:

- командні змагання на першому рівні курсу для посилення командного духу, підвищення конкурентоспроможності та комунікативних навичок;
- практичні форми управління, які будуть організовані в декількох університетах, які введуть нову стратегію оцінювання результатів навчання, коли переможці отримають вищі бали;
- більше залучення жінок і бонусна система оцінювання конкурсу (як всієї команди, так і окремих учасників через їх особистий внесок у результат) під час наукових проектів.

- продовження досліджень на наступному рівні для студентів з високими балами;
- використання принципу конкуренції між командами, щоб збільшити їх інтерес і мотивацію для продовження навчання;
- використання студентами або командами розробників інструментів для самостійного створення ігор та конкурсів елементів на другому рівні;
- використання студентами або командами розробників інструментів для самостійного створення повноцінних ігор та конкурсів або для поліпшення старих версій на другому рівні;
- пояснення аспірантами навчального матеріалу молодшим студентам.

Студенти, які навчаються, мають отримати хороші оцінки, пройти остаточне тестування, результат якого допомагає сформувавши пропозиції щодо подальшої його компетентності для вибору спеціальності та рекомендацій щодо аспірантури, а також поради для обрання відповідного навчального закладу, участь у міжнародних конференціях, які сприятимуть активній науковій роботі студентів.

Для проведення міжкомандних змагань, тестування та олімпіад тощо важливо скласти розклад змагань з врахуванням особливостей курсу, часових поясів проживання учасників, пройдених ними курсів тощо.

Розклад змагань надзвичайно важливий в контексті управління розподіленим дистанційним навчальним процесом. Він містить не тільки звичний для нас перелік дисциплін із зазначенням часу проведення змагання й список учасників, але і забезпечує систематизоване об'єднання усіх видів навчальної роботи у межах дисципліни й засобів їх реалізації у Web-системі [17].

Процес укладання розкладу в оптимальному випадку має ітеративний характер, тобто відбувається у декілька етапів під керівництвом людини-фахівця. Послідовність та кількість цих етапів, тобто методика укладання розкладу, залежать від специфіки закладу дистанційної форми навчання. Віртуальні кафедри, в яких ресурс часу не є критичним, чи, наприклад, не діє часова система навчання, цілком задовольнить варіант розкладу, автоматично згенерований на повній множині навчальних планів. Інші віртуальні кафедри можуть починати складати розклад з розподілу найбільш критичних (з погляду використання кафедральних ресурсів) семінарів [11]. Розклад на будь-якому етапі гарантовано залишається коректним і узгодженим з усіма вимогами; крім того, зручний редактор розкладу дає змогу вносити зміни, не порушуючи його цілісності і коректності [22, 14]. Проблема визначення оптимальних умов проведення занять для дистантних слухачів в реальному режимі часу (семінарів) є доволі важливою, враховуючи різномірне географічне розташування, а отже, й різні часові характеристики учасників дистанційного навчання для проведення занять у реальному часі. Для синхронізації (оптимального часового розрахунку) розглянемо проблему знаходження часового проміжку одночасного відвідування семінарів студентами дистанційної форми навчання в режимі on-line.

Нехай маємо часовий проміжок  $T = [0, 24]$  – час за Грінвічем. Також маємо скінченну кількість студентів дистанційної форми навчання  $S = \{s_1, s_2, \dots, s_{n_s}\}$ . Тоді для студента  $s_i$  можливість відвідувати заняття в реальному режимі часу можна подати деякою функцією  $m_i(t)$ , область визначення якої  $t \in T = [0, 24]$ , а множина значень є відрізком  $[0, 1]$ . Те, що функція  $m_i(t) = 0$  в деякий проміжок часу, означає, що студент  $s_i$  у цей момент часу не може відвідувати заняття (або ігру, змагання, олімпіаду тощо, бо вже бере участь в іншій) і  $m_i(t) > 0$  в іншому випадку. При  $m_i(t) = 1$  студент  $s_i$  з 100 % впевненістю в цей момент часу може брати участь у певному процесі:

$$a_1 = m_1(t), a_2 = m_2(t), \dots, a_n = m_n(t).$$

Кожен студент може навчатися в певний проміжок часу, і йому відповідатиме функція  $m_i(t)$ . На рис. 2 наведено функції можливого відвідування електронних семінарів студентами дистанційної форми навчання. Функція  $m_1(t)$  демонструє циклічність часу. Функції  $m_2(t)$  та  $m_3(t)$  – типові графіки відвідування електронних семінарів студентами дистанційної форми навчання.

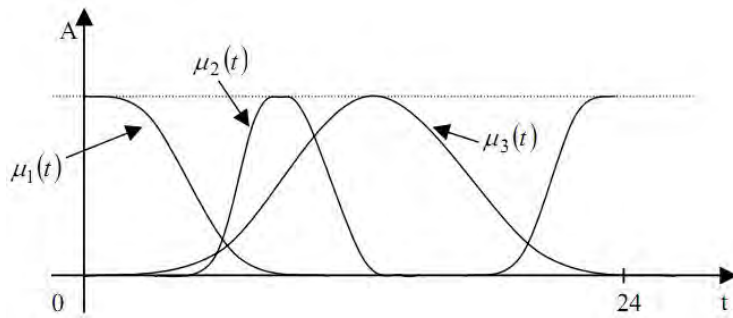


Рис. 2. Часові проміжки можливого відвідування семінарів студентами дистанційної форми навчання  $s_1$ ,  $s_2$  та  $s_3$  відповідно

Зауважимо, що функції  $\mu_i(t)$  необов'язково є неперервними.

Задача побудови розкладу занять для груп студентів полягає в знаходженні такого часового проміжку  $[t - \Delta t, t + \Delta t]$ , що  $A = a_1 + a_2 + \dots + a_n \rightarrow \text{Max}$ ,  $a_i > 0$ ,  $i = \overline{1, n}$ . Для знаходження максимуму функції мети запропоновано алгоритм 2 (ітераційний процес).

**Алгоритм 2.** Ітераційний процес знаходження максимуму функції мети для формування розкладу

**Етап 1.** Часовий проміжок  $T = [0, 24]$  розбиваємо на інтервали  $\Delta t$  (одна академічна година).

**Етап 2.**  $\forall t_i$  знаходимо значення функції мети  $A = a_1 + a_2 + \dots + a_n \rightarrow \text{Max}$ ,  $a_i > 0$ ,  $i = \overline{1, n}$  (рис. 3).

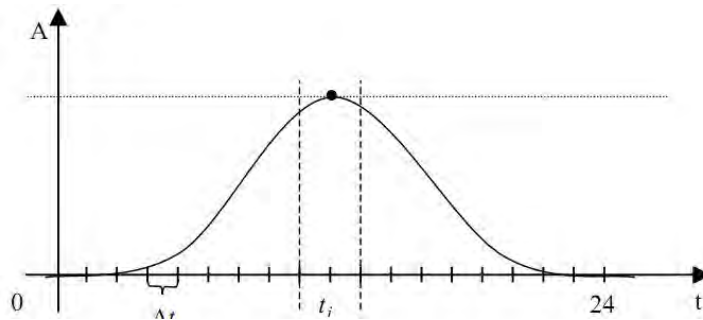


Рис. 3. Знаходження максимуму функції мети

**Етап 3.** Впорядковуємо масив значень функції мети за спаданням. Призначаємо  $i = 1$ .

**Етап 4.** Беремо  $i$ -й елемент масиву значень функції мети та перевіряємо, чи можливо в відповідний часовий проміжок  $[t - \Delta t, t + \Delta t]$  зробити призначення заняття.

**Етап 5.** Якщо для призначеного заняття виконуються умови  $a_i > 0$ ,  $i = \overline{1, n}$ , то вважаємо, що розв'язок поставленої задачі знайдено. Призначаємо  $i = i + 1$ . Якщо  $i \leq n$ , перехід на етап 4, інакше на етап 6.

**Етап 6.** Якщо, перебравши всі елементи масиву значень функції мети, не вдалося зробити призначення, тобто не існує  $[t - \Delta t, t + \Delta t]$  такого, що  $A = a_1 + a_2 + \dots + a_n \rightarrow \text{Max}$ ,  $a_i > 0$ ,  $i = \overline{1, n}$ , то необхідно поділити групу на підгрупи і повторити процедуру призначення заняття.

Еволюція інформаційного суспільства, а також соціально-економічних змін в Європі вимагає серйозних змін у багатьох сферах. Насамперед, це стосується реформи освіти та інновацій в навчальному процесі. Тенденція використання інформаційних та комунікаційних технологій означає побудову системи, що може не тільки опрацьовувати та аналізувати інформацію, але й створити умови для ефективного управління розподіленим процесом електронного навчання, і бути ефективним середовищем для навчання процесу формування, управління та обслуговування. Цей факт відкриває можливість створення універсального шаблону навчального процесу для

планування та підготовки навчальних матеріалів для різних форм освітніх і навчальних програм і різних установ, – так званих інтегрованих курсів (рис. 4).

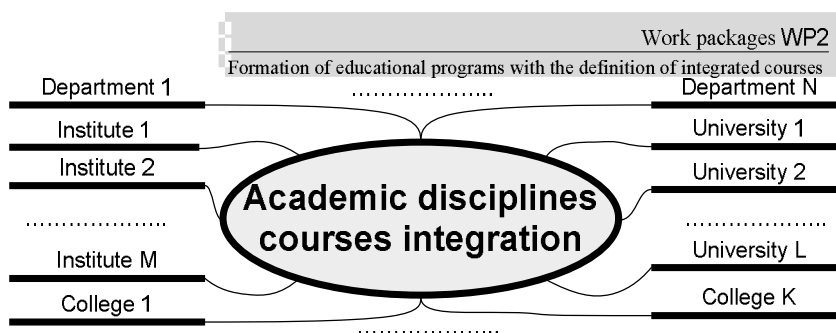


Рис. 4. Інтеграція курсів у міжвузівських консорціумах

Використання інформаційних та комунікаційних технологій має оптимізувати навчальне управління і створити ефективну організаційну структуру. Забезпечення навчального процесу технічними, інформаційними, інтелектуальними та програмними інноваціями вкрай важливо для управління сучасною розподіленою освітнього процесу, наприклад, для отримання спеціальності із штучного інтелекту. Складність потоку даних і обсяг інформації, яка передається в інтелектуальній інформаційній системі логістики аналізу інституційних структурних підрозділів, вимагають додаткових посібників з інтелектуальної навігації, обробки інформації, каталогізації і планування. Більшість адміністративних і управлінських завдань цих підрозділів призначає секретар-агент, якою можна вважати інтелектуальним ядром інституційних структурних підрозділів. Це не просто інструмент, який спрощує процес інтерактивного діалогу із системою (алг. 3, рис. 5), але й активний учасник його діяльності, унікальний інтерфейс агента, який слідує інструкціям користувачів про ручне планування, зв'язок і управління.

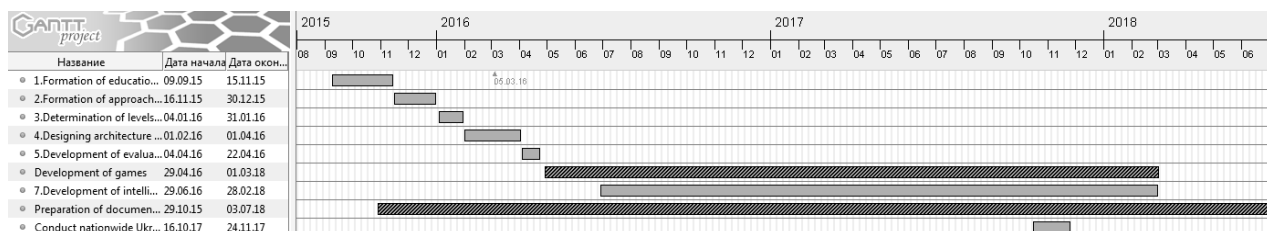


Рис. 5. Діаграма Ганта для запуску дистанційної освіти

**Алгоритм 3.** Ітераційний процес запуску дистанційної освіти за визначеною спеціальністю за інтегрованими навчальними програмами в міжвузівському консорціумі (*Work plan — Work packages, deliverables and milestones*).

**Етап 1.** Формування освітніх програм з визначенням інтегрованих курсів (Formation of educational programs with the definition of integrated courses).

**Етап 2.** Формування підходів до практичних завдань та оцінювання створення методів (Formation of approaches to the practical tasks and assessment methods creation).

**Етап 3.** Оцінювання знань студентів з навчальних предметів освітніх програм (Determination of levels of evaluation the students' knowledge in the study subjects of educational program).

**Етап 4.** Проектування архітектури системи оцінки знань (Designing architecture of the system evaluation of knowledge).

**Етап 5.** Розроблення системи оцінювання (Development of evaluation system).

**Етап 6.** Розроблення ігор (Development of game).

**Етап 7.** Розвиток інтелекту помічника (Development of intelligence assistant).

**Етап 8.** Підготовка документації для кожного етапу дослідницького проекту дистанційного навчання (Preparation of documentation for each stage of research project of e-learning).

**Етап 9.** Проведення загальнонаціональних українських конкурсів для рекламного проекту впровадження та супроводу системи (Conduct nationwide Ukrainian competitions for advertising project).

Основні вимоги до спільного науково-дослідницького проекту дистанційного навчання:

1. Різноманітність стосовно вмісту (процеси, засоби масової інформації, дані тощо)
2. Різноманітність відповідних людино-комп'ютерних інтерфейсів.
3. Взаємодія різних підсистем.
4. Взаємодія між висновками доменів.
5. Розширюваність і масштабованість запропонованого багатовимірного освітнього інструменту.

Запропоновані курси поділено на три рівні. Проходження всіх курсів для студента або команди є обов'язковим! Перехід між курсами дозволений за проходження всіх курсів попереднього рівня. Поки не буде пройдено всіх курсів одного рівня, перехід на наступний рівень неможливий. Приклад курсу "Системи штучного інтелекту"

### **I рівень**

1. Алгоритми та структури даних (Algorithms and Data Structures).
2. Вища математика (Математичний аналіз та лінійна алгебра, Mathematical Analysis and Linear Algebra).
3. Дискретна математика (Discrete Mathematics)
4. Основи програмування (Fundamentals of Programming).
5. Операційні системи (Operating Systems).
6. Іноземна мова (бажано Англійська мова, Foreign Language).
7. Теорія ймовірності та статистичні методи (Probabilistic Methods and Statistics).

### **II рівень**

1. Основи комп'ютерної графіки (Fundamentals of Computer Graphics).
2. Основи комп'ютерних мереж (Fundamentals of Computer Networks).
3. Об'єктно-орієнтоване програмування (Object Oriented Programming).
4. Числові методи та оптимізація (Numerical Methods and Optimization).
5. Основи баз даних (Database Fundamentals).
6. Компонентне програмування (Component Programming).
7. Телекомунікація та передавання даних (Telecommunication and Data Transmission).

### **III рівень**

1. Програмна інженерія (Software Engineering).
2. Вбудовані системи (Embedded Systems).
3. Штучний інтелект та експертні системи (Artificial Intelligence and Expert Systems).
4. Програмування комп'ютерної графіки (Computer Graphics Programming).
5. Графічний дизайн (Graphic Design).
6. Основи криптографії (Foundations of Cryptography).
7. Введення в виробництво кіно і телебачення (Introduction to Film and Television Production).

Кожний курс має три ступеня складності. Достатньо пройти перший, щоб курс вважався пройденим, але це значно зменшує сумарну кількість балів, які можливо отримати в кінці цього рівня. Це набагато зменшує шанси для команди виграти змагання і дійти до фінішу. Також це використовується при оцінюванні знань, навичок, компетентності як команд, так і окремих учасників. Результат змагання не лише для учасників гри, але і для розробників курсів оцінюється за декількома критеріями: не лише за часом, а ще і за обчислювальною складністю та схожимістю таких алгоритмів, якістю та складністю дизайну, реалізацією коду, наявністю інтелектуальної компоненти, за віком, статтю тощо).

Багатовимірний освітній інструмент кожного курсу складається з трьох ступенів:

1. Ступінь виконавця (гравець, учасник гри). Використовується для різних методів навчання для отримання основних знань з курсу та фіксує час виконання під час змагань між командами.

2. Ступінь адміна (розвідника, модератора, програміста). Використовується для вдосконалення ігрових елементів або написання свого гравця-бота, а також для гри з іншими командами. Учасники прописують нові правила гри для збільшення шансів виграшу та ускладнення гри для суперників.

3. Ступінь розробника (проектувальника, тим-лідера, системного аналітика). Використовується для створення нових курсів учасниками або вдосконалення старих курсів через зміну правил гри з врахуванням основних засад курсу та сучасних інноваційних тенденцій у цьому напрямі досліджень.

### **Приклади гри для курсу Algorithms and Data Structures.**

#### **I ступень** “Поняття алгоритму. Лінійні структури даних”

Основні визначення, властивості та класи алгоритмів. Рекурсія. Структури даних: масив, дескриптор, множина, запис, таблиця. Моделювання представлення в пам'яті векторів та таблиць. Символьні типи даних. Операції над рядками. Інтегровані структури даних. Стек і черга. Динамічні структури даних. Списки. Хеш-таблиця. Методи вирішення колізій. Переповнення таблиці та рехешування.

Типові задачі – ігри із змаганнями між учасниками та командами для I ступеня:

#### **I. Лабіринт**

1. Вийдіть з лабіринту так швидко, як це можливо
2. Знайти щось в лабіринті, так швидко, як це можливо, а потім вийти
3. Знайти  $n$  ключів у лабіринті. Кожен ключ має свій унікальний номер. Кожен ключ має бути використаний на вказаних місцях з тим самим номером. Коли було використано всі ключі – вихід з лабіринту буде роздруковано. (Отже, гравець повинен шукати в лабіринті, коли знайшли ключ; прийняти його, коли знайшов певне місце. Використовувати ключ або його повернути, коли він знайде ключ).

#### **II. Перегони**

1. Дійти до фінішу, уникаючи різних пасток.
2. Перейдіть до фінішу, уникаючи пасток, і ловити бонуси, які прискорюють автомобіль гравця, коли це корисно.
3. 1 і 2 плюс намагаються знищити інші автомобілі, штовхаючи їх до пасток. Гравці можуть перемогти ворога.

#### **III. Хитрість**

1. Гравець повинен знайти і вкрасти щось з певної кімнати, уникаючи камер;
2. 1 плюс рухомі опікуни, рухомі камери і пастки.
3. 2 плюс за допомогою спеціальних стелс-навички: невидимість, відключення камери і т.д.

#### **II ступень** “Нелінійні структури даних”

Дерева. Бінарні дерева. Графи. Обхід графу. Пошук. Алгоритми проходження дерев. Подання бінарних дерев у зв'язній пам'яті. Формування бінарного дерева. Застосування бінарних дерев в алгоритмах пошуку. Види бінарних дерев: збалансоване дерево, червоно-чорне дерево, avl-дерево. Подання графа за допомогою структур суміжності. Алгоритми проходження графу.

Типові задачі – ігри із змаганням між учасниками та командами для II ступеня.

#### **IV. Поліцей**

1. Гравець має наздогнати та схопити шпигуна.
2. Гравець має знайти та наздогнати шпигуна
3. Гравець повинен включити і переміщати камери, щоб виявити шпигуна.

#### **V. Захоплення бази**

1. Гравці повинні захопити всі бази на мапі швидше, ніж ворога. Для захоплення гравець повинен залишитися на базі в певний момент часу.
2. Додатково до попереднього гравці можуть відключити противника, дивлячись на нього певну кількість часу.
3. Додатково до попереднього гравці можуть зламати ворога.

## **VI. BrainRacing**

1. Автомобіль на дорозі. Два гравці повинні вирішувати різні завдання із просування автомобіля. Коли автомобіль потрапляє в базу одного гравця, він перемагає.

2. Гравці Програма автомобілів, щоб рухатися в їхньому напрямку, і злам його, якщо машина вже запрограмовано.

3. Додатково до попередніх присутні перегони автомобілів.

## **III ступінь “Алгоритми”**

Рекурсивні алгоритми обробки структур даних. Пошук і сортування в одновимірних масивах. Файли. Операції над файлами. Алгоритми пошуку. Лінійний пошук. Двійковий (бінарний) пошук. Інтерполяційний пошук в масиві. Бінарний пошук з визначенням найближчих вузлів. Пошук в таблиці. Пошук рядка: прямий; алгоритм Батога, Моріса і Пратта; алгоритм Боуера і Мура. Алгоритми сортування. Методи внутрішнього сортування: сортування включенням, обмінне сортування, сортування вибором, сортування поділом, сортування за допомогою дерева, пірамідальне сортування, побудова піраміди методом Флойда, сортування злиттям. Зовнішнє сортування: пряме злиття, природне злиття, збалансоване багатошляхове злиття, багатофазне злиття. Типові задачі – ігри із змаганням між учасниками та командами для III ступеня.

## **VII. Марсохід**

1. рух з перешкодами

2. 1 + розпізнавання об’єктів з навчанням щодо перешкод та ідентифікації відомих вже об’єктів

3. 1 і 2 + знайти та доставити необхідний об’єкт, наприклад, ґрунт з певної координати.

## **VIII. Літак**

1. навігація літака

2. 1+ автопілот

3. 1 і 2+ розвідувач безпілотник з розпізнаванням місцевості... та уникання обстрілу

## **VIII. Зіркові війни (стратегія)**

1. знаходження та ідентифікація об’єктів (ворожого флоту, власного флоту, планет, галактик, сузір’я, туманностей тощо).

2. 1+колонізація та дослідження невідомої території галактики, з врахуванням зайнятих позицій ворожими силами.

3. 1 і 2 + захоплення ворожої території з вираховування власних сил порівняно з суперником.

Перед початком навчання дистанту присвоюється рівень знань 0. Успішне проходження хоча б I ступеня всіх дисциплін  $i$ -го курсу (I рівень компетентності) та здача тестових завдань з того курсу автоматично дозволяє студенту перейти на I рівень  $i+1$  курсу. Але це суттєво зменшує його загальний рейтинг і формується його індивідуальний портрет як сумлінного виконавця. Участь в курсі II і III рівнів піднімає компетентність студента та збільшує його загальний рейтинг, що сприяє уточненню його інформаційного портрету як програміста-розробника та проектувальника відповідно (табл. 1) [1–10, 23, 24].

Згідно із статистикою проходження студентом курсів (час проходження, командна робота, кількість проходження ступенів курсу) формується його інформаційний портрет, який в кінці закінчення навчання дозволить сформулювати пропозиції та рекомендації щодо профільної спеціалізації, вибору предметної області для подальших досліджень згідно з кваліфікацією та особистими навичками, щодо подальшого навчання в аспірантурі та поради обрання відповідного університетського консорціуму.

Ітераційний процес запуску дистанційної освіти за визначеною спеціальністю за інтегрованими навчальними програмами в міжвузівському консорціумі (Work plan — Work packages, deliverables and milestones).

Матриця компетентності програміста за знанням теорії

Теорія:	2 <sup>n</sup> (Рівень 0)	n <sup>2</sup> (Рівень I)	n (Рівень II)	log(n) (Рівень III)
Структури даних	Не розуміє різниці між масивом і зв'язковим списком.	Може пояснити і використовувати на практиці масиви, зв'язні списки, словники і т.д.	Розуміє плюси і мінуси використання тих чи інших базових структур даних (розмір пам'яті, час виконання операцій з даними, в чому різниця між масивами і зв'язковими списками в цьому плані). Може пояснити, як реалізувати хеш-таблиці і як обробити колізії. Пріоритетні черги і способи їх реалізації і т.д.	Знання складних структур даних, таких як B-дерево, Біноміальна купа і фібоначевська купа, АВЛ-дерево, Червоно-чорне дерево, Косе дерево, Список з пропусками, TRIE-структури і т.д.
Алгоритми	Не може знайти середнє значення масиву чисел.	Базові методи сортування та пошуку. Обхід і пошук у структурах даних.	Дерева, Графи, “простий шлях” і “розділяй-і-володаруй” алгоритми, розуміє значимість рівнів наведеної тут матриці.	Розпізнає і пише рішення динамічним програмуванням, добре знає алгоритми на графах, чисельні методи, ідентифікує проблеми класу NP.
Системне програмування	Не знає, що таке компілятор, лінковщик або інтерпретатор.	Базове розуміння компіляторів, лінковщик і інтерпретаторів. Розуміє, що таке асемблерний код і як працюють програми на рівні заліза. Невелике розуміння віртуальної пам'яті і пейджінга.	Розуміє, чим відрізняється kernel mode від user mode, що таке мульти-трединг, способи синхронізації і як реалізовані примітиви синхронізації, може читати асемблерний код. Розуміє, як працюють мережі, мережеві протоколи і може реалізувати передачу даних через сокети.	Розуміє, як працює весь “програмний стек”: залізо (CPU + Пам'ять + Кеш + Переривання + мікрокоди), двійковий код, асемблер, статична і динамічна лінковка, компіляція, інтерпретація, JIT-компіляція, збірка сміття, купа, стек, адресація пам'яті ...

Крім тестування за теорією дистанта перевіряють через результати ігри та самого перебігу гри рівень навичок, досвід, програмування та знання інструментів та засобів (табл. 2, 3) [1–10, 23, 24].

Таблиця 2

Матриця компетентності програміста за набутими навичками

Навички:	2 <sup>n</sup> (Рівень 0)	n <sup>2</sup> (Рівень I)	n (Рівень II)	log(n) (Рівень III)
Контроль версій вихідних кодів	Бекап вихідних кодів в папку з датою бекапа.	VSS та основи CVS / SVN в якості користувача	Спеціаліст по CVS and SVN. Знає, як розгалузити і злити, налаштувати репозиторій і т.д.	Знає розподілені системи VCS. Пробував Bzr / Mercurial / Darcs / Git.
Авто-матизація build'ів	Знає, як запустити Build з середовища програмування.	Вміє білдити з командного рядка.	Може налаштувати скрипт для збірки основної системи.	Може налаштувати скрипт для складання системи та документації, інсталяторів, генераторів “release notes” і додати скрипт в систему контролю версій початкових кодів.
Автоматизоване тестування	Думає, що тестування – це робота тестерів.	Створює автоматизовані юніт-тести і свої хороші юніт-тести для коду, який пише в даний час.	Пише код в стилі Test-driven Development (TDD).	Розуміє і може налаштувати автоматичні тести на функціонал, користувацький інтерфейс і завантаження / продуктивність ...



Матриця компетентності програміста за рівнем програмування

:	2 <sup>n</sup> (Рівень 0)	n <sup>2</sup> (Рівень I)	n (Рівень II)	log(n) (Рівень III)
1	2	3	4	5
<b>Декомпозиція задачі</b>	Послідовні рядки коду, а copy / paste - для повторного використання коду.	Може розбивати рішення задачі на декілька функцій.	Здатний створювати багаторазово використовувані функції / об'єкти, які вирішують спільні завдання.	Використовує відповідні структури даних і алгоритми. Створює загальний / об'єктно-орієнтована код, який інкапсулює ті умови задачі, які можуть бути змінені.
<b>Декомпозиція системи</b>	Система не складніше одного класу або файлу.	Декомпозує задачі і проектує систему в межах однієї платформи або технології.	Може спроектувати систему, яка охоплює декілька технологій / платформ.	Може візуалізувати і проектувати складні системи з декількома лінійками продуктів та інтеграцією із зовнішніми системами. Також повинен уміти проектувати системи підтримки роботи: моніторинг, генерація звітів, аварійні переходи на використання запасних ресурсів.
<b>Спілкування</b>	Не може висловити своїх думок/ідей. Погано з правописом і граматику.	Його розуміють. Хороші правопис і грамика.	Може ефективно спілкуватися.	Розуміє і пояснює думки / дизайн / ідеї / специфіку в точно вираженій формі, у спілкуванні відповідає ситуації. Зі збільшенням аутсорсингу розробки ПЗ в ті країни, де англійська не є рідною мовою, це питання стало актуальнішим.
<b>Організація коду в файлі</b>	Немає чіткої організації у файлі.	Методи згруповані логічно і за викликами.	Код розділений на регіони, має хороші коментарі, в т.ч. з посиланнями на інші файли початкових кодів.	Файл має розділи "license header", "summary", хороші коментарі, несуперечливу розстановку прогалін і табуляції. Файл повинен виглядати красиво.
<b>Організація коду між файлами</b>	Не спадає на думку чітко організувати код за допомогою поділу на файли.	Подібні файли групуються в папку.	Кожен фізичний файл призначений для чогось одного, наприклад, для оголошення одного класу або для реалізації одного функціоналу тощо	Організація коду на фізичному рівні точно відповідає проекту, і за іменами файлів і структурою папок можна зрозуміти, як спроектована ця реалізація.

1	2	3	4	5
Організація дерева вихідних кодів	Все в одній папці.	Простий розподіл коду в логічні підкаталоги.	Немає “кодових” залежностей. Бінарники, ліби, документація, білди, сторонній код – все розкладено у відповідні папки.	Структура дерева вихідного коду відповідає логічній ієрархії та організації коду в проекті. Дивлячись за іменами файлів і структурою папок можна зрозуміти, як спроектована дана система. Різниця між цим пунктом і попереднім полягає в масштабі організації. Організація дерева вихідних кодів належить до всього комплексу продуктів, які визначають систему.
Читабельність коду	Односкладові імена.	Хороші імена файлів, змінних, класів, методів і т.д.	Немає довгих функцій, код нестандартний. Виправлення і допущення в коді пояснюються коментарями.	Допущення в коді супроводжуються assert'ами, потік операцій в коді природний - немає глибокої вкладеності умов або методів.
Безпечне програмування (defensive coding)	Не розуміє цієї концепції.	Перевіряє всі аргументи і ставить assert'и на критичні допущення в коді.	Переконується, що повертається значення і що обробляє виключення в потенційно бажному коді.	Має свою власну бібліотеку, що допомагає в безпечному програмуванні, пише юніт-тести, які емулюють збої.
Обробка помилок	Пише код для “ідеального” випадку, коли все працює і немає збоїв.	Обробка помилок в коді, який або кидає виняток, або генерує помилку.	Переконується, що після того, як сталася помилка / виключення, програма продовжує працювати, а непотрібні більш ресурси, конекшені і пам'ять були коректно звільнені обробником помилки.	Пише код так, щоб визначати можливі помилки на ранньому етапі, дотримується послідовної стратегії обробки виключень у всіх шарах коду, розробляє загальні принципи обробки виключень у всій системі.
IDE	В основному використовує IDE для редагування тексту.	Здатний ефективно користуватися меню в IDE. Знає деякі тонкощі середовища.	Для самих використовуваних функцій середовища знає гарячі клавіші.	Написав власні макроси.

1	2	3	4	5
API	Часто потребує звернення до документації.	Пам'ятає самі часто використовувані API.	Великі і глибокі знання API.	Написав бібліотеки, які обертають API, для спрощення завдань, які найбільш часто зустрічаються. Ці бібліотеки також часто заповнюють прогалини в API. Прикладом API може бути Java-бібліотека, .NET фреймворк або API якого-небудь додатку.
Фреймворки	Не використовує ніяких фреймворків за рамками основної платформи.	Знає, але не використовує популярні фреймворки, доступні для його платформи.	Професійно користувався більш ніж одним фреймворком і добре розбирається в ідіомах фреймворків.	Є автором фреймворка.
Вимоги	Розуміє виставлення вимоги і пише код відповідно до специфікації.	Задає питання, які стосуються не розглянутих у специфікації випадків.	Розуміє картину загалом і пропонує додаткові аспекти, які повинні бути описані в специфікації.	Може запропонувати кращі альтернативи і слідувати виставленим вимогам, ґрунтуючись на власному досвіді.
Скрипти	Відсутнє знання скриптових інструментів.	Batch-файли/shell.	Perl/Python/Ruby/VBScript/Powershell.	Писав і публікував повторно використовувані скрипти.
Бази даних	Думає, що Excel – це база даних.	Знає основи баз даних, нормалізацію, ACID, транзакції і може написати прості select'и.	Може спроектувати хороші нормалізовані схеми БД, з урахуванням запитів, які будуть виконуватися; вміло використовує уявлення, збережені процедури, тригери і власні типи даних. Розуміє різницю між кластеризованими і не-кластеризованими індексами. Спеціаліст у використанні ORM-Тулза.	Може здійснювати адміністрування БД, оптимізацію продуктивності БД, індексу оптимізацію, писати складні select'и, може замінити використання курсору викликами функцій SQL, розуміє, як дані зберігаються всередині, як зберігаються індекси, має уявлення про те, як дзеркалом і реплікують БД і т. д. Розуміє як працює двофазний commit.

### Етап 1. Формування освітніх програм з визначенням інтегрованих курсів (Formation of educational programs with the definition of integrated courses).

Мета – Узгодження та синхронізація програм навчання між трьома ВНЗ: Львівської політехніки, польським та сербським ВНЗ. Формування освітніх програм з визначенням інтегрованих курсів. Адаптація програми під навчання через гру, тобто впровадження інноваційних підходів до поліпшення навчальної програми вищої освіти у сфері ІТ.

Узгодження робочих програм з курсів “Дискретна математика”, “Теорія алгоритмів” та “Штучний інтелект”. Навчальна програма повинна містити предмети з класичної ІТ школи. Це дасть змогу поліпшити професійні та практичні навички студентів та зменшити кількість предметів на семестр. Після стадії моніторингу студент поступив вчитися, залежно від складності курсу і його/її конкретних компетенцій (рівень II–III).

Розробити інноваційну програму шляхом впровадження нових інтегрованих курсів, які будуть викладатися студентам в різних університетах консорціуму. Викладачі поділяться професійним досвідом, студенти спільно виконуватимуть проекти, співпрацюючи в міжвузівській та міжнародній командах (відсутність кордонів, проблем мови та раси, статі та віку). Програма навчання в навчальному плані має містити предмети з класичної школи (I рівень навчання). Це дасть змогу підвищити рівень професійних та практичних навичок студентів і скоротити число випробуваних протягом семестру. Після стадії моніторингу, студент зараховується до вивчення ряду предметів залежно від складності сцени та специфіки компетентності самого студента (II–III рівень навчання). Курс організовується у вигляді гри, участь в якій сприятиме отриманню знання з предметної області, працювати в команді.

### Етап 2. Формування підходів до практичних завдань та оцінювання створення методів (Formation of approaches to the practical tasks and assessment methods creation).

Мета: створення та генерування нових підходів до формування практичного завдання і методів оцінювання знань абітурієнтів (рис. 6).

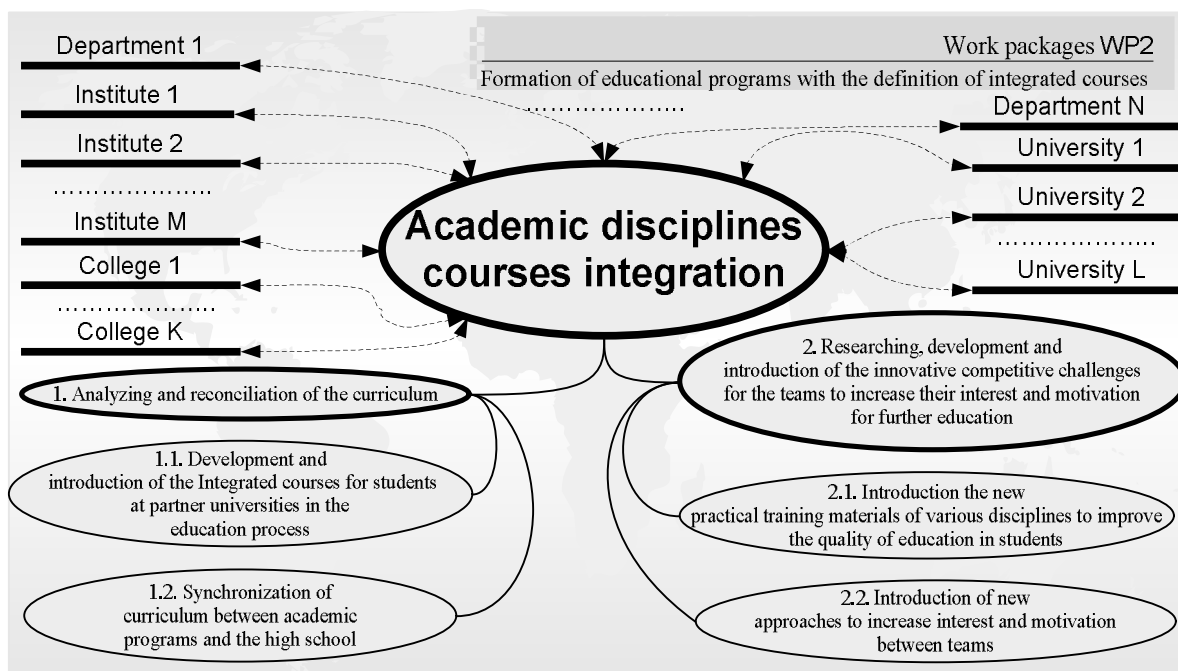


Рис. 6. Формування підходів до практичних завдань та оцінці створення методів

1. Створення нових підходів до формування практичного завдання.
2. Створення нових методів оцінювання знань абітурієнтів.
3. При вивченні інтегрованих курсів доцільно ввести підходи інноваційного навчання до формування практичних завдань і методів оцінки, а саме:

- формування міжкомандних змагань на першому рівні проходження курсу для формування командного духу, підвищення рівня конкурентоспроможності в професійній сфері та підвищення комунікальності.

- практичні форми управління, щоб організувати у декількох університетах нову стратегію результатів студентських знань студентів, де переможці отримують більш високі бали.

- залучення більше дівчат через обов'язковість наявності в команді їх представників та бонусну систему оцінювання результатів змагань (як цілої команди так і окремих учасників через їх особистий вклад в результат) під час виконання науково-практичних проектів.
- продовження наукових досліджень на наступних рівнях для студентів з високими балами за результатами оцінки проектів.
- використання принципу конкурентоспроможності та змагань між командами студентів для посилення їх інтересу та мотивації до продовження навчання.
- використання студентами або командами на другому рівні навчання засобів розроблення для самостійного створення елементів ігор та конкурсів.
- використання студентами або командами на третьому рівні навчання засобів розроблення для самостійного створення ігор та конкурсів нових курсів або вдосконалення старих.
- участь студентів старших курсів у викладанні та поясненні матеріалу студентам молодших курсів.

На рис. 7 подано схему ієрархії навчання через ігри.

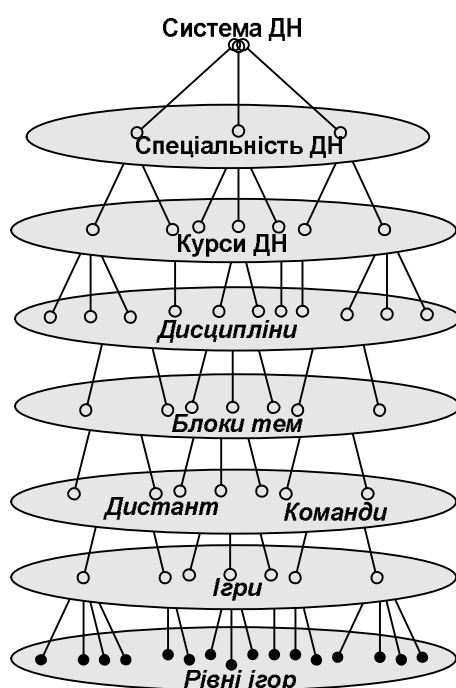


Рис. 7. Ієрархія навчання через ігри

Введемо позначення для курсів (3 рівня) та робочих програм дисципліни (ІРП):

- |       |                       |    |  |
|-------|-----------------------|----|--|
| I.    | Дисципліна 1 – ІРП1   | 1. | {а, б, в} – блок тем 1 (алгоритми ⇒ гра 1) |
|       | Дисципліна 2 – ІРП2   | 2. | {а, б, в} – блок тем 2 (алгоритми ⇒ гра 2) |
| ..... |                       | 3. | {а, б, в} – блок тем 3 (алгоритми ⇒ гра 3) |
|       | Дисципліна 7 – ІРП7   |    |  |
| II.   | Дисципліна 1 – ІІРП1  |    |  |
|       | Дисципліна 2 – ІІРП2  |    |  |
| ..... |                       |    |  |
|       | Дисципліна 7 – ІІРП7  |    |  |
| III.  | Дисципліна 1 – ІІІРП1 |    |  |
|       | Дисципліна 2 – ІІІРП2 |    |  |
| ..... |                       |    |  |
|       | Дисципліна 7 – ІІІРП7 |    |  |

Рівні ігор в дисципліні:

- А.** Грати в гру (одному або командою змагатися на час) як дозвілля та навчання алгоритмам теми  $i$ .
- В.** Модифікувати гру  $i$  (запрограмувати свій алгоритм з теми  $i$  або вдосконалити гру  $i$  псевдокодом, доробити щось)
- С.** Нова гра псевдокодом для теми  $i$  або нової підтеми з дисципліни  $a$ , що не включена в загальний список тем.

Модель дистанційної освіти через ігри:

$$O' = \langle O, P, C, D, L, R, G, K, U, a, b, g, d \rangle,$$

де  $P = \{p_1, p_2, \dots, p_{N_p}\}$  – спеціальності;  $C$  – курси при  $C = \{c_1, c_2, \dots, c_{N_c}\}$  або  $C_i^P = \{C_{i1}^P, C_{i2}^P, C_{i3}^P\}$ ,  $C_{ij}^P \subset C_i^P \subset C$ ;  $D = \{d_1, d_2, \dots, d_{N_D}\}$  – дисципліни;  $L$  – блоки тем при  $L = \{l_1, l_2, \dots, l_{N_L}\}$ , або  $L_i^D = \{L_{i1}^D, L_{i2}^D, L_{i3}^D\}$ ,  $L_{ij}^D \subset L_i^D \subset L$ ;  $R$  – теми при  $R = \{r_1, r_2, \dots, r_{N_R}\}$ , або  $R_i^L = \{r_{i1}^L, r_{i2}^L, \dots, r_{iN_{LR}}^L\}$ ,  $R_{ij}^L \subset R_i^L \subset R$ ;  $G$  – ігри при  $G = \{G_A, G_B, G_C\}$ ,  $G = \{g_1, g_2, \dots, g_{N_G}\}$ ,  $G_i^D = \{G_A^D, G_B^D, G_C^D\}$ ,

$$G_A = \{g_1, g_2, \dots, g_m\}, G_B = \{g_{m+1}, g_{m+2}, \dots, g_{m+n}\}, G_C = \{g_{m+n+1}, g_{m+n+2}, \dots, g_{N_G}\},$$

$$G_A^D \subset G_i^D \subset G \text{ і } G_A^D \subset G_A \subset G,$$

$$G_B^D \subset G_i^D \subset G \text{ і } G_B^D \subset G_B \subset G,$$

$$G_C^D \subset G_i^D \subset G \text{ і } G_C^D \subset G_C \subset G,$$

тому  $G_A^D = \{g_1^D, g_2^D, \dots, g_m^D\}$ ,  $G_B^D = \{g_{m_D+1}^D, g_{m_D+2}^D, \dots, g_{m_D+n_D}^D\}$ ,  $G_C^D = \{g_{m_D+n_D+1}^D, g_{m_D+n_D+2}^D, \dots, g_{N_{GD}}^D\}$ ,

$G_C = \{g_{m+n+1}, g_{m+n+2}, \dots, g_{N_G}\}$ ;  $K$  – команд,  $K = \{k_1, k_2, \dots, k_{N_K}\}$ , або  $K_i^D = \{K_{i1}^D, K_{i2}^D, \dots, K_{iN_{KD}}^D\}$ ,

$K_{ij}^D \subset K_i^D \subset K$  або  $K_i^C = \{K_{i1}^C, K_{i2}^C, \dots, K_{iN_{KC}}^C\}$ ,  $K_{ij}^C \subset K_i^C \subset K$ ;  $U$  – дистант,  $U = \{u_1, u_2, \dots, u_{N_U}\}$  або

$U_i^C = \{U_{i1}^C, U_{i2}^C, \dots, U_{iN_{KC}}^C\}$ ,  $U_{ij}^C \subset U_i^C \subset U$  або  $U_i^D = \{U_{i1}^D, U_{i2}^D, \dots, U_{iN_{KD}}^D\}$ ,  $U_{ij}^D \subset U_i^D \subset U$ ;  $a$  –

управління переходом дистантів або команд між рівнями гри;  $b$  – оцінювання знань дистанта з дисципліни та формування рекомендацій дистанту щодо зміни напрямку навчання для успішного складання іспитів з дисципліни і/або курсу;  $g$  – управління переходом дистантами між курсами (оцінювання знань дистанта із закінченням курсу та формування рекомендацій дистанту щодо зміни напрямку навчання для успішності здачі дисциплін цього курсу);  $d$  – формування рекомендацій дистанту щодо його професії згідно із статистикою навчання.

На рис. 8 подано дерево переходу між курсами, дисциплінами та темами.

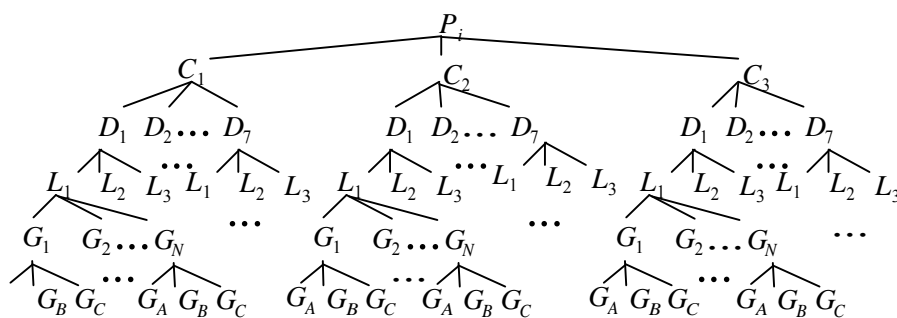


Рис. 8. Ієрархія навчання за допомогою ігор

На рис. 9, 10 подано мережі Петрі переходу між дисциплінами та курсами, де  $a$  – управління переходом дистантів або команд між рівнями гри;  $b_1$  – оцінювання знань дистанту з дисципліни та формування рекомендацій дистанту щодо зміни напрямку навчання для успішності складання іспиту з дисципліни і/або курсу;  $g_1$  – управління переходом дистантами між курсами (оцінювання знань дистанта із закінченням курсу та формування рекомендацій дистанту щодо зміни напрямку навчання для успішності здачі дисциплін цього курсу);  $d_1$  – формування рекомендацій дистанту щодо його професії згідно із статистикою навчання.

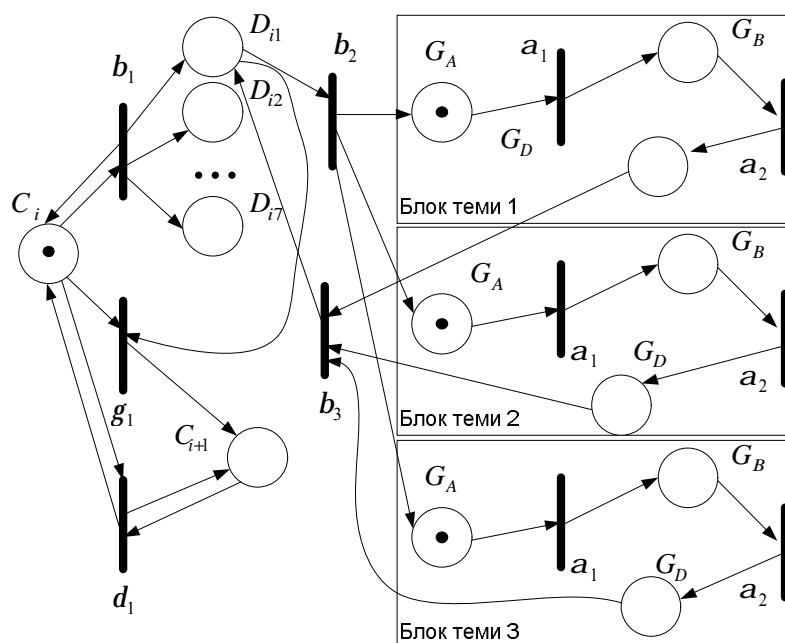


Рис. 9. Мережа Петрі переходу між дисциплінами та курсами

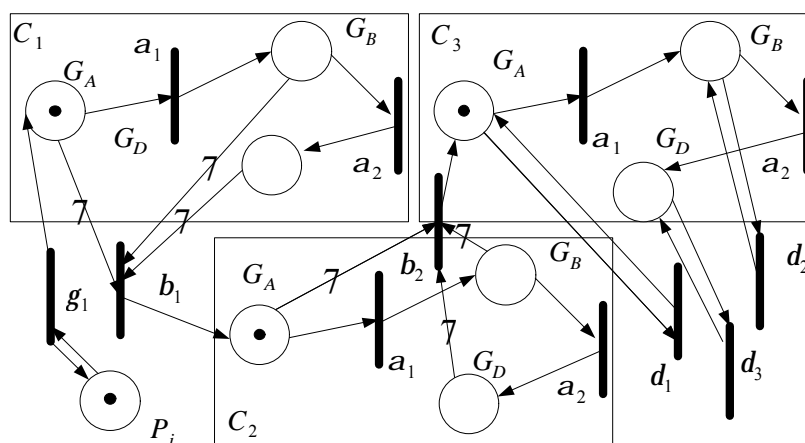


Рис. 10. Мережа Петрі переходу між дисциплінами з формуванням рекомендацій

**Етап 3. Визначення рівнів оцінювання знань студентів у навчальних предметах освітніх програм (Determination of levels of evaluation the students' knowledge in the study subjects of educational program).**

Формування критеріїв оцінювання знань студентів за їх компетентністю. Визначення рівнів оцінки знань студентів у навчальних предметів освітніх програм:

1. Формування критеріїв оцінювання знань студентів у команді та в одиночній грі.
2. Формування критеріїв оцінювання знань студентів згідно з різними рівнями складності участі в грі (навчання на курсі першого, другого чи третього рівня складності).
3. Визначення рівнів оцінювання знань студентів.

4. Студенти, які отримують хороші результати під час навчання, проходять підсумкове тестування за результатами якого та результатами історії навчання формуються рекомендації щодо подальшого навчання в аспірантурі та поради обрання відповідного університетського консорціуму.

Студенти з комп'ютерних наук отримують знання, вміння та навички з проектування і впровадження сучасних інформаційних систем та технологій, систем підтримки прийняття рішень в економіко-виробничих та бізнесових системах, математичних та інтелектуальних методів аналізу бізнес-процесів, методів та комп'ютерних засобів прогнозування в бізнесі, інвестуванні, страхуванні та кредитуванні. Фахівці з системного аналізу проектують, створюють й експлуатують

комп'ютерні системи аналізу, прогнозування, управління та проектування динамічних процесів в макроекономічних, технічних, технологічних, екологічних і фінансових об'єктах, аналізують бізнес-процеси з погляду їхньої подальшої автоматизації, розробляють технічні завдання та специфікації, тестують програмне забезпечення, формують аналітичні звіти. Магістри "Системи штучного інтелекту" вивчають застосування систематизованого і упорядкованого підходу до аналізу, проектування, програмування, експлуатації і супроводу інтелектуальних систем як інструменту наукових досліджень та розв'язання складних науково-технічних задач. Магістри "Системи і методи прийняття рішень" вивчають методологію системного, структурного та об'єктно-орієнтованого аналізу складних об'єктів і систем, проектування й використання баз і сховищ даних, розроблення та застосування інформаційних комп'ютерних систем і технологій аналітичного та управлінського характеру, принципи побудови та особливості функціонування сучасних інформаційних систем і комп'ютерних мереж. Магістри "Управління проектами" вивчають програмні засоби та технології для управління ІТ-проектами, методологію керівництва проведенням наукових досліджень і розроблень проектів та програм, методи виконання ІТ-проектів засобами CASE-технологій, оцінювання етапів і кінцевих результатів виконання робіт ІТ-проекту та коригування параметрів ІТ-проекту, принципи побудови та особливості функціонування сучасних обчислювальних систем, інформаційних і комп'ютерних мереж, нормативно-правові документи і методологію формування проектної документації з ІТ проектів.

Під час навчання формується профіль студента-дистанта  $O_{ijk}$  в межах  $[0;1]$  (табл. 4). Треба аналізувати детальніше історію користувача (дистанта ДН), його профіль та виводити з цього його компетентність згідно з історією навчання за визначеним набором характеристик (основні характеристики учасників команд програмістів, матриця компетентності програміста, тімлідера, адміністратора, аспіранта, архітектора, знання англійської (рівні), командна гра (рівні), гендерність (участь, керівництво тощо), час виконання завдання, творчість при виконанні завдання тощо) та формування пропозиції щодо вибору роботи.

Таблиця 4

**Правила формування профілю компетентності дистанта**

Курс	Дисципліна	Тема	$G_A$	$G_B$	$G_C$	Результат ДН
$C_i$	$D_j$	$L_k$	$O_{ijk}^A$	$O_{ijk}^B$	$O_{ijk}^C$	$O_{ijk}^A \cdot O_{ijk}^B \cdot O_{ijk}^C$
		$L_{k+1}$	$O_{ijk+1}^A$	$O_{ijk+1}^B$	$O_{ijk+1}^C$	$O_{ijk+1}^A \cdot O_{ijk+1}^B \cdot O_{ijk+1}^C$
		$L_{k+2}$	$O_{ijk+2}^A$	$O_{ijk+2}^B$	$O_{ijk+2}^C$	$O_{ijk+2}^A \cdot O_{ijk+2}^B \cdot O_{ijk+2}^C$
Результат ДН			$\sum_k O_{ijk}^A$	$\sum_k O_{ijk}^B$	$\sum_k O_{ijk}^C$	Аналіз результатів ДН

Ієрархія рекомендованих основних професій ІТ-профілю:

- 1) Аспірантура
- 2) Архітектор систем
- 3) Senior Executive – головний керівник ІТ-проекту
- 4) Senior Manager – старший менеджер
- 5) Manager – менеджер
- 6) Associate Manager – помічник менеджера
- 7) Team Leader – керівник групи розробників ІТ проектів
- 8) Senior Software Engineer – старший інженер-програміст
- 9) Software Engineer – інженер-програміст
- 10) Програміст
- 11) Адміністратор
- 12) Associate Software Engineer – помічник інженера-програміста
- 13) Рекрутер



14) Тестувальник

15) Асистент, лаборант

Дистанти (студенти) – 3 курси та всі дисципліни – тоді присудження кваліфікації за спеціальністю. Успіх – хоча би закінчити 1 рівень дисципліни за всіма блоками тем. Згідно з навичками рекомендація обрання професії згідно з табл. 5, 6 (за умови, що результати більші за 50 %, тобто більші за 30 балів).

Таблиця 5

**Правила визначення компетентності дистанта**

Правила	I рівень	II рівень	III рівень
Оцінки	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $\sum_i \sum_j \sum_k O_{ijk}^B = 0,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C > 30$
Компетентність	Тестер, рекрутер	Бізнес, програміст	Аспірантура, Архітектор систем
Професія та сфера діяльності випускників	Комерційні фірми, проектні організації, комерційні банки та біржі, інвестиційні компанії, фірми-розробники програмних продуктів для економіки та бізнесу, інформаційні підрозділи Державної податкової служби, органи державної статистики, інформаційні підрозділи структур державної адміністрації, інші установи та організації як провідні спеціалісти з розроблення і застосування інтелектуальних систем опрацювання інформації та комп'ютерних систем підтримки прийняття рішень в проектуванні, розробленні, налагодженні та використанні інформаційних технологій та автоматизованих інформаційних систем для корпорацій, фінансово-кредитних установ, банків та ін.	Банки, фондові, валютні та товарні біржі, аналітичні відділи міністерств, податкові інспекції, відділи валютних, товарних і фондових ринків, Міністерство фінансів України і Міністерство економіки України, аналітичні відділи спецслужб, пенсійні фонди, центри проектування баз даних, розроблення автоматизованих систем.	Наукові співробітники; аналітик комп'ютерного банку даних; адміністратор бази даних; адміністратор системи; аналітик з комп'ютерних комунікацій; конструктор комп'ютерних систем; інженер-дослідник з комп'ютеризованих систем та автоматизованих систем; інженер з програмного забезпечення комп'ютерів; інженер з комп'ютерних систем; інженер-програміст; програміст; програміст прикладний; програміст системний; інженер із застосування комп'ютерів. Адміністратор баз даних, аналітик з комп'ютерних комунікацій, аналітик комп'ютерного банку даних, аналітик систем, науковий співробітник, керівник комп'ютерного центру, конструктор комп'ютерних систем, менеджер проекту, провідний програміст, провідний фахівець з програмного забезпечення, програміст прикладний, інженер-програміст. Керівник проектів; архітектор проектів; журналіст проектів; керівник групи розробників ІТ проектів; проектний аналітик; старший розробник; керівник ІТ служб на підприємствах; начальник відділу інформаційних технологій; консультант директора з інформаційних технологій; адміністратор засобів інформаційних технологій підприємства; директор програми; директор проекту, сертифікований менеджер проекту; професійний менеджер проекту; практикуючий менеджер проекту.
Рекомендація компаній, корпорацій та ВНЗ	ELEKS Software, N-iX, Google, GlobalLogic, Amazon, EPAM Systems, HelloWebApps.com, Persha Studio, Nikitova LLC, Finport Technologies Inc, Datamart, LLC, Limelight Networks, Lohika systems Inc., SoftServe, CBS Local, EAE-Consult, LUKOIL-Inform, Hearst, LanguageMate, Code and Theory, Software Mind, CPP Investment Board, OpenText, Citibank, Romexsoft, Spline Software, World Link Solutions, Consensia, Kartinka, Lviv Academic Gymnasia, WebLancer.net, "Mikro-Kod" Ltd, Harrah's Entertainment, TRK "NTK", 1C-Tellur.Modul, Skelia Ukraine, Edvantis, PrimoCollect, Remit, DataArt, HelloWebApps, IBM Canada, Toroki UA, ShadowMasters, Profit*Studio, Tekomp, Hit24, Infopulse, 9YARDS, Symphony Solutions, SoftBistro, Ericpol, Vakoms, Solano Labs Inc.		

## Правила визначення рекомендації щодо професій дистанта

I випадок	II випадок	III випадок
$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $\sum_i \sum_j \sum_k O_{ijk}^B = 0,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$	$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $0 < \sum_i \sum_j \sum_k O_{ijk}^B \leq 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $0 < \sum_i \sum_j \sum_k O_{ijk}^B \leq 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$
Асистент, лаборант	Тестувальник	Рекрутер
IV випадок	V випадок	VI випадок
$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C = 0$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $0 < \sum_i \sum_j \sum_k O_{ijk}^B \leq 30,$ $0 < \sum_i \sum_j \sum_k O_{ijk}^C \leq 30$	$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $0 < \sum_i \sum_j \sum_k O_{ijk}^C \leq 30$
Associate Software Engineer	Адміністратор	Програміст
VII випадок	VIII випадок	IX випадок
$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $0 < \sum_i \sum_j \sum_k O_{ijk}^B \leq 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C > 30$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $0 < \sum_i \sum_j \sum_k O_{ijk}^C \leq 30$	$\sum_i \sum_j \sum_k O_{ijk}^A > 30$ $0 < \sum_i \sum_j \sum_k O_{ijk}^B \leq 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C > 30$
Software Engineer – Інженер-програміст	Senior Software Engineer – Старший інженер-програміст	Team Leader – Керівник групи розробників ІТ проектів
X випадок		
$0 < \sum_i \sum_j \sum_k O_{ijk}^A \leq 30$ $\sum_i \sum_j \sum_k O_{ijk}^B > 30,$ $\sum_i \sum_j \sum_k O_{ijk}^C > 30$		
Associate Manager – Помічник менеджера		

*Переваги майбутньої професії в ІТ*

1. Гарантоване працевлаштування – 90 % випускників працевлаштовано.
2. Високий рівень заробітної плати – середньомісячний дохід ІТ-спеціаліста 1500 \$.
3. Кар'єра та комфортні умови праці – в 25 років спеціалісти стають керівниками проектів в таких компаніях, як Facebook, Google, Microsoft, Samsung, IBM, HP, Dell, Amazon.

**Етап 4. Проектування архітектури системи оцінювання знань (Designing architecture of the system evaluation of knowledge).**

Система організована у вигляді порталу (рис. 11), де навчання студентів реалізоване у формі ігор, що забезпечує учасникам (користувачам, акторам) системи (СТУДЕНТ, КОМАНДА (команда студентів), ЛЕКТОР, ПРОГРАМІСТ, АНАЛІТИК, МОДЕРАТОР, АДМІНІСТРАТОР СИСТЕМИ, КУРАТОР (куратор групи), КООРДИНАТОР (головний над проектом), КОНСОРЦІУМ (учасники проекту):

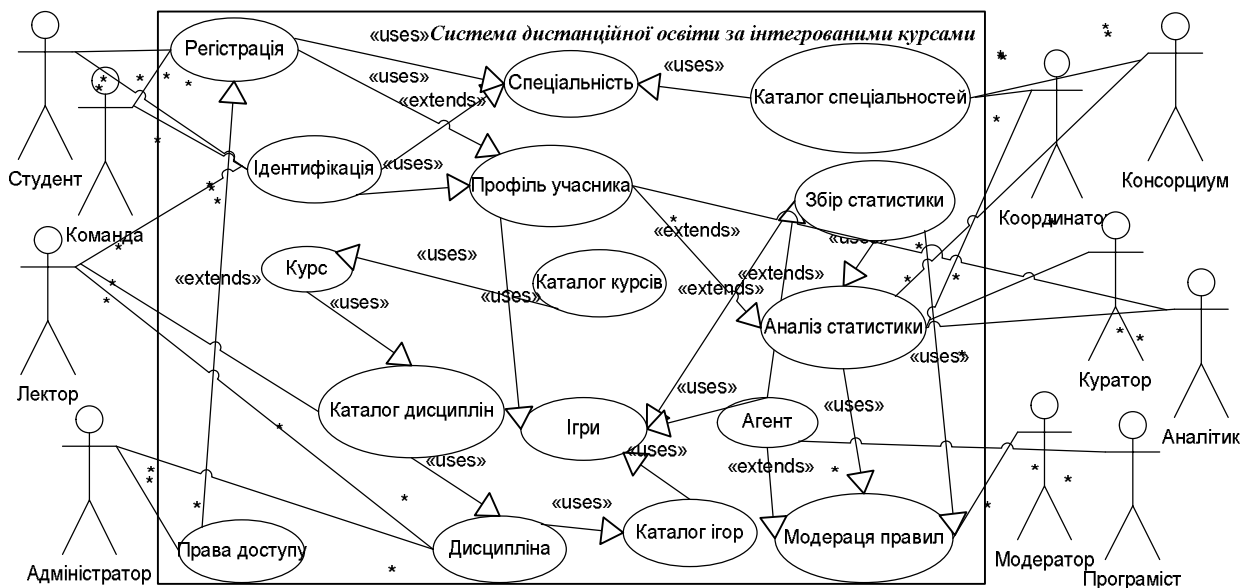


Рис. 11. Діаграма варіантів архітектури системи оцінки знань

1. Навігацію запропонованими розробленими курсами, де можна отримати знання з предметної області;
2. Скористатися дружнім інтерфейсом для приємного проведення часу та підтримання зв'язку в команді;
3. Фіксує статистику дій учасників для визначення їх рейтингу, формування списку учасників, продемонструвати організаційні та лідерські навички, ентузіазм з приводу отримання нових знань;
4. Фіксує статистику дій учасників як в одиничній грі, так і в команді для порівняння успішності та корегування поведінки у команді, що підвищує зв'язок навички, вчить роботі в команді, дає практичну підготовку в спілкуванні діловою англійською, підвищує конкурентоспроможність у своїй галузі знань.

Основні можливості актора СТУДЕНТ: реєстрація за спеціальністю, ідентифікація для навчання, гра (навчання дисципліні через гру), тести (проходження тестування для оцінки знань), перехід між рівнями, перехід між курсами, ознайомлення з оцінкою та своїм профілем, перехід між дисциплінами, ознайомлення з рівнем набутої компетентності під час навчання, ознайомлення з рекомендаціями для скеровування процесу навчання під обрану компетентність, програмування псевдокодом.

Основні можливості актора КОМАНДА: ідентифікація (команди + учасників), гра, тести (команди + учасників), перехід між рівнями, перехід між курсами, оцінки (команди + учасників), профіль (команди + учасників), рекомендації (команди + учасників), програмування псевдокодом, оцінка іншими командами, учасниками команд, лекторами, кураторами тощо.

Основні можливості актора ЛЕКТОР: ідентифікація, наповнення тестів, наповнення курсів, оцінювання: система оцінювання через тести, системи оцінювання рівня програмування (швидкість, Ю оптимальність коду тощо).

Основні можливості актора ПРОГРАМІСТ: ігри оновлення, нові ігри, ШІ – модифікація агенту, ідентифікація, оцінювати код учасників, нові правила гри і змагань - програмування.

Основні можливості актора АНАЛІТИК: ідентифікація, аналіз статистики відвідування, аналіз статистики роботи системи, аналіз статистики гри команд, аналіз статистики гри учасників, аналіз навчання на курсах та дисциплінах, аналіз гендерної політики, аналіз залежності рівня навчання від статі учасника, аналіз залежності рівня перемог команд з врахуванням наявності/відсутності дівчат у командах, формування пропозицій.

Основні можливості актора МОДЕРАТОР: ідентифікація, нові каталоги та рубрики, правила змагань, правила гри, нові опції статистики та правила, дайджести, фільтри некоректного псевдокоду, дозвіл на зміни в іграх з коректним псевдокодом та дозвіл на коректні ігри до курсів, блокування учасників або команд з порушенням правил гри та змагань, контроль дозволів на переходу учасників на інший рівень або курс.

Основні можливості актора АДМІНІСТРАТОР: ідентифікація, підтримка та супровід системи, контроль, управління, дизайн, модерація за побажаннями (координаторів, лекторів, модераторів, аналітика тощо) учасників проекту та користувачів системи.

Основні можливості актора КУРАТОР групи учасників (множина учасників або декілька груп, декілька учасників груп можуть мати декілька кураторів): консультування учасників (команд) як з гри, псевдокодування, так і з курсів, ідентифікація, оцінювання активності учасників, оцінювання командної гри, контроль за процесом навчання, контроль за процесом змагань.

Основні можливості актора КООРДИНАТОР (куратор проекту): контроль за роботою аналітика та програміста, управління (учасники та модератори), архітектура системи, контроль за іграми (запуск та статистика проходження), аналіз результатів роботи аналітика, формування пропозиції, адаптація та інтеграція курсів, адаптація та інтеграція дисциплін і робочих програм, аналіз і контроль профілів учасників проекту та формування інших пропозицій та рекомендацій, імідж ДО, супровід та просування системи, просування ДО, інтернет-маркетинг.

Основні можливості актора КОНСОРЦІУМ (інші учасники проекту): контроль певними курсами або спеціальностями, дисциплінами, управління певними курсами або спеціальностями, дисциплінами, нові курси, нові дисципліни, нові робочі програми, нові пропозиції, аналіз профілів як курсів, так і команд, аналіз профілів учасників та формування інших ділових пропозицій.

Згідно із статистикою проходження студентом курсів (час проходження, командна робота, кількість проходження ступенів курсу) формується його інформаційний портрет, який в кінці закінчення навчання дозволить сформулювати пропозиції та рекомендації щодо профільної спеціалізації, вибору предметної області для подальших досліджень згідно кваліфікації та особистих навичок, подальшого навчання в аспірантурі та поради обрання відповідного університетського консорціуму.

#### **Етап 5. Розроблення системи оцінювання (Development of evaluation system)**

Розроблення системи оцінювання студентів, які навчаються отримати хороші ціни, прийняти остаточне тестування, результат яких разом з історією навчання та цілей студента допомагає сформулювати пропозиції щодо подальшої аспірантури та поради, щоб вибрати підходящий університет. Участь у міжнародних конференціях, які сприятимуть активній науковій роботі студентів і розширити свої дослідження. Інноваційні підходи до практичних завдань і методів оцінювання мають бути введені, а саме:

- організувати командні змагання на першому рівні курсу для формування командного духу, підвищення конкурентоспроможності та комунікативних навичок;
- практичні форми управління, організовані в декількох університетах, які введуть нову стратегію оцінювання результатів навчання студента, коли переможці отримують вищі бали;
- залучати більше жінок та використовувати бонусну систему до оцінювання конкурсу (як всієї команди, так і окремих учасників через їх особистий внесок у результат) під час наукових проектів;
- продовжувати дослідження на наступному рівні для студентів з високими балами;
- використовувати принцип конкуренції між командами, щоб збільшити їх інтерес і мотивацію для продовження навчання;
- використовувати студентами або командами розробників інструментів для самостійного створення ігор та конкурсів елементів на другому рівні;
- використовувати студентами або командами розробників інструментів для самостійного створення повноцінних ігор та конкурсів або для старих версій поліпшення на другому рівні;

#### **Етап 6. Розроблення ігор (Development of game)**

Розроблення ігор згідно з узгодженими робочими програмами з курсів “Дискретна математика”, “Теорія аглоритмів” та “Штучний інтелект”.

1. Розроблення ігор для курсу “Теорія алгоритмів”
2. Розроблення ігор для курсу “Дискретна математика”,
3. Розроблення ігор для курсу “Штучний інтелект”.

Кожний курс має три ступеня складності. Достатньо пройти перший ступінь, щоб курс вважався пройденим, але значно зменшується сумарна кількість балів, яку можна отримати в кінці проходження цього рівня. Це набагато зменшує шанси для команди виграти змагання. Також це використовується для оцінювання знань, навичок, компетентності як команд, так і окремих учасників. Результат змагання не лише для учасників гри, але і для розробників курсів оцінюється за декількома критеріями – не лише за часом, а й за обчислювальною складністю та збіжністю таких алгоритмів, якістю та складністю дизайну, реалізацією коду, наявністю інтелектуальної компоненти, змаганням у віковій групі, статтю тощо).

Багатовимірній освітній інструмент кожного курсу складається з трьох ступенів:

1) Ступінь виконавця (гравець, учасник гри). Використовується для різних методів навчання для отримання основних знань з курсу. Лише фіксує час виконання під час змагань між командами.

2) Ступінь адміністратора (розвідника, модератора, програміста). Використовується для вдосконалення ігрових елементів або написання свого гравця-бота і використовується їх для гри з іншими командами. Учасники прописують нові правила гри для збільшення шансів виграшу та ускладнення гри для суперників.

3) Ступінь розробника (проектувальника, тим лідера, системного аналітика). Використовується для створення нових курсів учасниками або вдосконалення старих курсів через зміну правил гри з врахуванням основних засад курсу та сучасних інноваційних тенденцій у цьому напрямі досліджень.

#### **Етап 7. Розвиток інтелекту помічника (Development of intelligence assistant)**

Розвиток інтелекту помічника, що дає можливість покращити наукову грамотність студентів, знаходячи інноваційні рішення для вирішення проблем проектів.

1. Розроблення intelligence assistant для ігор з курсу “Теорія алгоритмів”
2. Розроблення intelligence assistant для ігор з курсу “Дискретна математика”,
3. Розроблення intelligence assistant для ігор з курсу “Штучний інтелект”.
4. ...

#### **Етап 8. Підготовка документації для кожного етапу дослідницького проекту дистанційного навчання (Preparation of documentation for each stage of research project of e-learning)**

Основні вимоги до спільного науково-дослідного проекту ДН:

1. Різноманітність за різного вмісту (процеси, засоби масової інформації, дані тощо)
2. Різноманітність відповідних людино-комп'ютерних інтерфейсів.
3. Взаємодія різних субсистем.
4. Взаємодія між висновками different доменів.
5. Розширюваність і масштабованість запропонованого багатовимірного освітнього інструменту.

#### **Етап 9. Проведення загальнонаціональних українських конкурсів для рекламного проекту впровадження та супроводу системи (Conduct nationwide Ukrainian competitions for advertising project)**

1. Аналіз статистики відвідування інформаційного ресурсу та користування ним через Google Analytics.
2. Застосування інтернет-маркетингових методів для просування інформаційного ресурсу.
3. Розміщення додаткової реклами у відповідних структурах, наприклад, Google AdWords.
4. Реклама в інших ВНЗ різних країн.
5. Проведення черех систему олімпіад та змагань із залученням не лише учасників (студентів) проекту. Проведення шкільних, студентських та аспірантських конференцій.
6. Періодичне проведення змагань-турнірів.

## Висновки та перспективи подальших наукових розвідок

Запропоновано інноваційні підходи до поліпшення навчальної програми вищої освіти у сфері ІТ. Розглянуто методи пошуку і залучення студентів до наукової та практичної роботи через участь у командних змаганнях і спільних міжвузівських науково-практичних проектах. Запропоновано метод покращення наукової грамотності студентів за допомогою інноваційних рішень для вирішення проблем проектів, що спростило наукове зростання для учасників міжвузівських проектів. Отже, запропонований комплекс математичних моделей, на якому ґрунтується подальша організація розподіленого процесу навчання, є основою для інтелектуальної надбудови інтегрованого середовища розроблювача дистанційних навчальних систем, що дає змогу вирішувати завдання кількісного подання параметрів дистантного слухача, динамічного контролю його стану й адаптивного керування навчальним процесом. Подальші дослідження плануються проводити за такими основними напрямками:

- Розроблення методики адаптивного тестування з використанням елементів нечіткої логіки для зменшення похибки оцінювання знань.
- Розроблення алгоритму динамічної модифікації моделі навчання у випадку неможливості досягнення за певний час мети навчання, використовуючи обрану індивідуальну стратегію навчання.
- Пошук методів оцінювання особистісних особливостей отримання знань дистантним слухачем для підвищення ступеня індивідуалізації навчального процесу та його якості.
- Дослідження технології інтелектуальних агентів для введення в структуру інтелектуальних мережевоцентричних систем дистанційного навчання допоміжних засобів інтелектуальної навігації, опрацювання інформації, каталогізації і планування.

1. Голощук Р. О. Математичне моделювання процесів дистанційного навчання / Р. О. Голощук, В. В. Литвин, Л. В. Чирун, В. А. Висоцька // Інформаційні системи та мережі: Вісник Національного університету "Львівська політехніка". – 2003. – № 489. – С. 100–109. Режим доступу: <http://ena.lp.edu.ua:8080/bitstream/ntb/8697/1/10.pdf>. 2. System administrator leveling matrix. Режим доступу: [https://docs.google.com/spreadsheets/d/1FBr20VIOePQH2aAH2a\\_birvdB1NOTHZaD8U5e2MOMiw/pub?output=html](https://docs.google.com/spreadsheets/d/1FBr20VIOePQH2aAH2a_birvdB1NOTHZaD8U5e2MOMiw/pub?output=html). 3. Таблица уровней для системного администратора. Режим доступу: <http://habrahabr.ru/post/145148/>. 4. Матрица компетентности программиста. Режим доступу: <http://habrahabr.ru/post/37707/>. 5. Матрица компетентности программиста / Programmer Competency Matrix. Режим доступу: [https://dev.by/pages/programming\\_matrix](https://dev.by/pages/programming_matrix). 6. Haugeland John. Programmer Competency Matrix. Perspectives on Software, Technology and Business / John Haugeland. Режим доступу: <http://sijinjoseph.com/programmer-competency-matrix/>. 7. Кравчук О. Матрица компетентности програміста / О. Кравчук. Режим доступу: <http://buffer-overflow.org.ua/wp-content/uploads/2009/03/Programmer%20competency%20matrix.htm>. 8. Формування і розвиток проектної команди. Підручник Управління проектами. Режим доступу: <http://www.bookz.com.ua/4/12.htm>. 9. Ролі розробників. Microsoft Solutions Framework. Моделі MSF. Екстремальне програмування. Режим доступу: <http://lib.mdpu.org.ua/e-book/vstup/L8.htm>. 10. Керівництво і лідерство. Режим доступу: <http://library.if.ua/book/36/2443.html>. 11. Abramson D., Krishnamoorthy M., Dang H. Simulated Annealing Cooling Schedules for the School Timetabling Problem (1997). <http://www.rdt.monash.edu.au/~davida/papers/cool.ps.Z> (5/10/2000). 12. Автоматизация проектирования компонентов дистанционного обучения и диагностика качества знаний специалистов для сети ИНТЭРНЭТ. Я. Э. Львович <http://www.fcde.ru/de/st109.html> – 11К – 19.08.1999. 13. Астанин С. В. Сопровождение процесса обучения на основе нечеткого моделирования // Дистанционное образование. – М., 2000. – № 5. – С. 27–32. 14. Белоус Н. В., Выродов А. П., Шубин И. Ю. Инструментальная система подготовки учебно-методических материалов для дистанционного образования // Образование и виртуальность. – Харьков-Севастополь: УАДО, 2000. – С. 209–214. 15. Белоус Н. В., Выродов А. П., Шубин И. Ю. Математические аспекты проектирования интегрированной среды разработчика

компьютерных обучающих систем // Сб. науч. тр. 5-й Междунар. конф. Украинской ассоциации дистанционного образования. – Харьков-Ялта, 2001. – С. 335–349. 16. Белоус Н. В., Выродов А. П., Шубин И. Ю. Математические модели в интегрированной среде разработчика компьютерных обучающих систем // Сб. науч. тр. 5-го форума “Радиоэлектроника и молодежь в XXI веке”. Ч. 1. – Харьков: ХТУРЭ, 2001. – С. 175–176. 17. Верес О. М. Алгоритм укладання розкладу навчальних занять у ВНЗ // Вісн. Нац. ун-ту “Львівська політехніка”. – 1998. – № 330. – С. 40–51. 18. Голощук Р. О. Системні принципи побудови інтелектуальної інформаційної системи “Віртуальна кафедра” // Вісн. Нац. ун-ту “Львівська політехніка”. – 2001. – № 438. – С. 42–45. 19. Голощук Р. О., Висоцька В. А. Інтерактивна взаємодія та зворотний зв'язок у системі дистанційного навчання // Вісн. Нац. ун-ту “Львівська Політехніка”. – 2002. – № 464. – С. 44–53. 20. Костюкова Н. И., Попков В. К. Математические модели, дидактические и эргономические аспекты разработки автоматизированных обучающих комплексов // Дистанционное образование. – 1999. – № 6. – С. 18–21. 21. Литвин В. В., Бакаїм Р. Б., Процовський О. Й., Садовий В. М., Шаховська Н. Б., Шаховський Р. В. Основні принципи та функціональне наповнення інформаційної системи “Школа” для автоматизації управління навчальним процесом // Вісн. Нац. ун-ту “Львівська політехніка”. – 1999. – № 383. – С. 145–149. 22. Литвин В. В., Назаров О. Ю., Чумаченко С. В. Інтелектуальна інформаційна система планування навчального процесу // Вісн. Нац. ун-ту “Львівська політехніка”. – 2000. – № 406. – С. 179–184. 23. Рашкевич Ю. М., Пелешко Д. Д., Пасека Н. С., Стецюк А. Б. Проектирование Web-ориентированных распределенных учебных систем // УСМ. – 2002. – № 3/4. – С. 72–79. 24. Яцишин Ю. В., Чип І.Є. Математична модель планування навчального процесу у вищому навчальному закладі: постановка задачі // Вісн. Нац. ун-ту “Львівська політехніка”. – 1998. – № 330. – С. 269–273. 25. Берко А. Ю. Intranet архітектура інтелектуальних систем електронного навчання / А. Ю. Берко, В. А. Висоцька // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – 2001. – № 438. – С. 3–10. 26. Висоцька В. А. Система опрацювання структури електронного підручника / В. А. Висоцька // Інформаційні системи та мережі: Вісник Національного університету “Львівська політехніка”. – 2003. – № 489. – С. 49–63.