

ПРО ОДИН З ПІДХОДІВ ВИКОРИСТАННЯ РОЗПОДІЛЕНИХ ЕКСПЕРТНИХ НАВЧАЛЬНИХ СИСТЕМ У ДИСТАНЦІЙНІЙ ОСВІТІ

У статті запропоновано принцип розподілення одного класу експертних систем та розглянуто його застосування в системах дистанційної освіти.

Вступ

Для дослідників у галузі дистанційної освіти неабиякий інтерес становлять комп'ютерні навчальні системи [2]. З огляду на специфіку систем дистанційної освіти, що надають одночасний доступ багатьом користувачам до значного за обсягом навчального матеріалу, який знаходиться в мережі, навчальні системи доцільно реалізовувати у розподіленому варіанті. Поміж існуючих видів комп'ютерних навчальних систем найпривабливішими для дистанційної освіти можна назвати інтелектуальні навчальні системи, які завдяки спеціальним методам пропо-

нують своїм користувачам інтелектуальну підтримку в навчанні [7, 8]. До таких інтелектуальних систем навчання належать експертні навчальні системи, які на основі знань експертів-педагогів реалізують функції керування навчанням, діагностики помилок та розв'язання задач у деякій предметній області, демонструючи результати на рівні експертних [3]. Для використання експертних навчальних систем у дистанційній освіті необхідно розробити їх варіант реалізації у вигляді традиційної розподіленої системи, що й буде зроблено в цій роботі.

Розробка архітектури розподіленої експертної навчальної системи (РЕНС) передбачає за-

стосування методів та моделей двох технологій: побудови експертних навчальних систем (ЕНС) та побудови розподілених систем (РС). Для реалізації будь-якої РС потрібно визначитися з типом розподіленої архітектури, способом передачі повідомлень між компонентами РС, типом СУБД для тривалого збереження даних. Для побудови ЕНС необхідно вибрати спосіб представлення даних та виведення на них. Одночасно з цим існує проблема визначення у структурі ЕС такого поділу на окремі компоненти, який дає змогу ефективно функціонувати ЕС у розподіленому варіанті. Визначальними для оцінки ефективності запропонованого варіанту розподілення мають бути як задоволення вимог до РС узагалі, так і гнучкість у представленні знань ЕНС.

До вимог, які має задовольняти РС, належать: масштабованість, відкритість, неоднорідність, розподілення ресурсів та стійкість до відмов [6]. З огляду на значні втрати часу при взаємодії розподілених компонентів, порівняно з викликами локальних методів у централізованій системі, потрібно насамперед визначити таку компонентну структуру розподіленої ЕНС, при якій передачу даних між окремими її компонентами буде зведено до мінімуму. Здійснити такий розподіл у нашому випадку можна, проаналізувавши специфіку предметної області та особливості як самої структури експертної навчальної системи (ЕНС), так і того класу ЕС, до якого вона належить.

1. Огляд принципів побудови та функціонування ЕНС

Розглянемо класичний опис ЕНС, наведений у [3]. Архітектура ЕНС відображає модель процесу навчання. Визначається певна мета навчання за допомогою поточних характеристик моделі учня. Поки мети не досягнуто, повторюється такий цикл: на основі поточного стану моделі учня і методики навчання генерується задача (тут задача трактується у широкому сенсі, як будь-яка інформація, що потребує від учня дій у відповідь); відповідь учня порівнюється з еталонним розв'язанням і на підставі розходжень виконується діагностика помилок учня; за результатами діагностики коригуються поточні характеристики учня.

Згідно з цією моделлю процесу навчання, ЕНС являє собою сукупність трьох взаємодіючих ЕС: розв'язок задач у ПО, що вивчається (ЕС РЗ); діагностика помилок учня (ЕС ДП); планування процесу управління навчанням (ЕС УН).

Взаємодія ЕНС з учнем відбувається таким чином. ЕС УН відповідно до поточної мети формує чергове завдання для учня, яке одночасно передається і ЕС РЗ. Далі ЕС ДП порівнює розв'язки учня з еталоном, отриманим ЕС РЗ, і на основі розходжень намагається з'ясувати, які неправильні уявлення учня про ПО могли призвести до цих розходжень. У результаті діагностики змінюється уявлення ЕНС про учня, від-

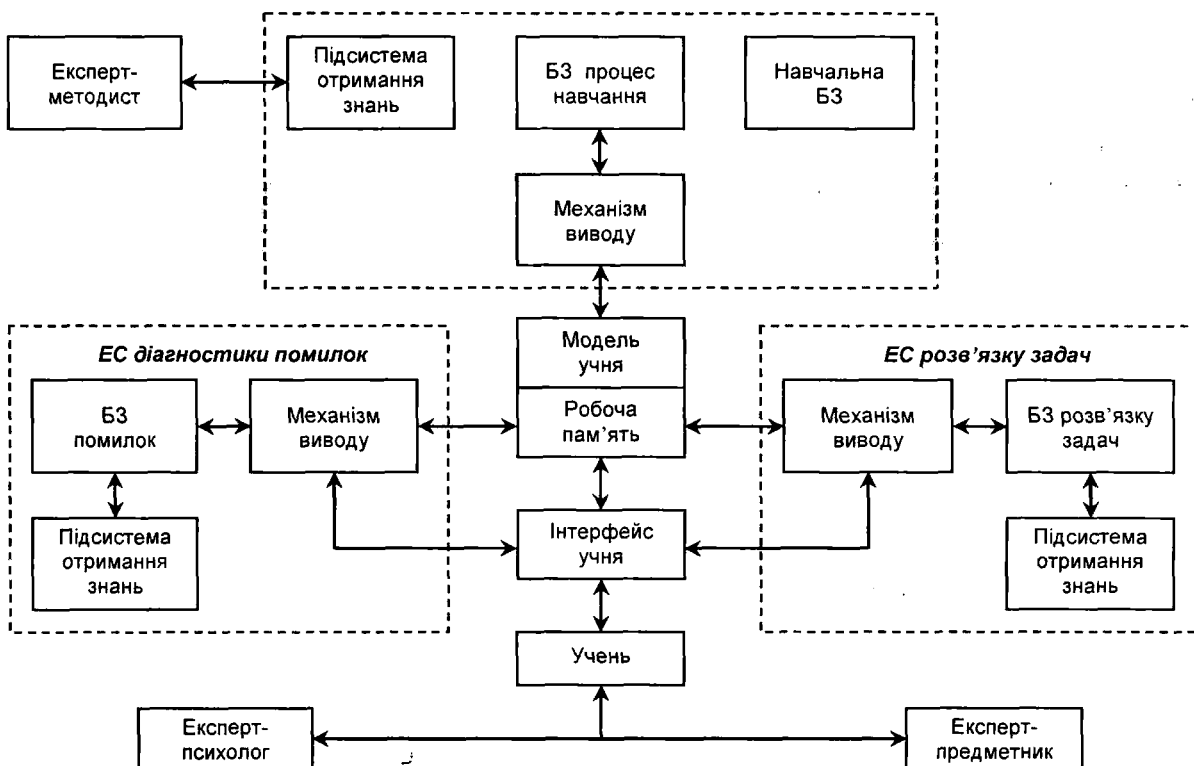


Рис. 1. Структура ЕНС

бражене у моделі учня (МУ), й управління знову отримує ЕС УН, яка уточнює поточну мету та формує нове завдання.

У структурі ЕНС можна виділити такі бази знань (БЗ): навчальна БЗ для даної ПО; модель учня; БЗ про можливі помилки; БЗ про процес навчання.

Навчальна база знань (НБЗ) описує не лише основні поняття і методи розв'язання задач у ПО, а й містить визначення понять, опис методів, приклади, вправи і задачі. На відміну від бази знань ЕС ВЗ, НБЗ повинна явно відображати структуру ПО і стратегічні знання про методи розв'язання задач. З іншого боку, НБЗ можна розглядати як представлення обмежень, у рамках яких ЕС УН планує навчання.

Модель учня (МУ) містить інформацію про стан знань учня: як загальні, інтегровані характеристики, так і ті, що відображають засвоєння ним поточного матеріалу. На початку МУ формується під час попереднього тестування учня. У термінах МУ виражається мета навчання.

База знань про помилки учня (БЗП) містить каталог можливих помилок учня і правила висування та перевірки гіпотез про неправильні уявлення учня, що призвели до даної помилки, на основі розходжень між розв'язаннями, запропонованими учнем, і ЕС ВЗ, а також поточного стану МУ.

База знань про процес навчання (БЗПН) містить знання про планування і організацію процесу навчання, загальні і спеціалізовані методики навчання. Для БЗПН запропоновано ієрархічну структуру, в якій виділено рівень педагогічних, стратегічних і тактичних знань. Знання представлено у вигляді правил, що визначають тип навчального впливу. Вибір наступного правила визначається спеціальними метаправилами, які залежать від поточної мети навчання і МУ. В цілому БЗПН можна розглядати як мета-знання стосовно решти БЗ.

2. Передумови для розподілу експертної системи

Відомо, що залежно від типів задач предметної області від ЕС може вимагатися вміння розв'язання проблем, які належать до різних типів діяльності. Тобто одну ЕС може бути призначено для виконання більше одного типу діяльності. Наприклад, поряд з діагностикою ЕС допомагає у відлагодженні, спостереженні поєднуються з керуванням, а планування з проектуванням [4].

Тому можна вважати, що існує такий клас ЕС, де наявні кілька відносно незалежних функціональних частин. Мається на увазі, що кож-

ну таку частину призначено для розв'язання окремої проблеми, причому в процесі розв'язання вона не потребує звернень до інших функціональних частин, а результат її роботи або видається користувачу, або використовується іншою функціональною частиною. Окрім цього, можна також виокремити ту частину БЗ, яка потрібна для роботи лише певній функціональній частині і не використовується іншими.

Отже, у складі певного класу ЕС можна визначити певні функціональні частини і фрагменти БЗ, якими користуються лише вони, а також вхідні й вихідні дані, потрібні для їх роботи, та джерела цих даних.

Також можна виділити певну координаційну частину системи, яка визначає послідовність передачі проміжних результатів від однієї функціональної частини до іншої. Для визначення такої послідовності координаційна частина має містити модель проблеми, що складається з набору параметрів, значення яких є результатами роботи певних функціональних частин. Також потрібна модель переходів, яка ставить у відповідність одному з можливих станів моделі проблеми одну або кілька функціональних частин, які можуть використовувати значення з поточних параметрів моделі проблеми як свої вхідні дані. У свою чергу, до складу кожної функціональної частини також може входити певна частина моделі проблеми, в тому разі, якщо при її вирішенні необхідно враховувати попередню історію розв'язання проблеми даною функціональною частиною.

Надалі в роботі буде запропоновано підхід до розподілення саме такого класу ЕС, для яких можливо визначити вищезгадані елементи.

2.1. Загальна концепція розподіленої архітектури ЕС

РЕС складається з таких частин: координаційний компонент (координатор); сервісні компоненти (функціональні частини); експертні системи, які використовуються координаційним компонентом; експертні системи, які використовуються сервісними компонентами; компонент клієнта (клієнт).

Користувач за допомогою компонента клієнта звертається до координатора для отримання інформації про запити (дії), які він може виконувати в системі. Ця інформація включає перелік можливих запитів клієнта, для кожного з яких вказується: тип сервісу; назва запиту, з яким потрібно звернутись до відповідного типу сервісного компонента; перелік значень параметрів, необхідних для виконання запиту сервісним компонентом; тип результату запиту, який

буде повернуто клієнту; перелік параметрів, які має повернути сервісний компонент клієнту для передачі координатору після виконання дії.

Після вибору одного із запитів клієнт звертається з ним до відповідного сервісного компонента й аналізує отриманий результат, який складається з двох частин: результату виконання запиту і значення параметрів для передачі координатору. Результат виконання клієнт може обробляти і видавати користувачеві, а значення параметрів передає координатору.

ЕС, яка використовується координаційним компонентом, містить такі знання: координаційна модель проблеми, модель переходів та конфігураційні дані. Модель проблеми містить дані, необхідні для вироблення рішення про можливі запити клієнта.

Модель переходів містить дані, необхідні для вибору можливих запитів, а також інформацію, потрібну для виконання запитів: назва запиту; тип значення, який він повертає, назви елементів моделі проблеми, значення яких потрібно передати як параметри запиту; тип сервісного компонента, до якого потрібно звертатися з даним запитом; назви параметрів, які потрібно повернути координатору після виконання запиту; дані про те, які частини моделі проблеми потрібно аналізувати для вибору можливих запитів.

Конфігураційні дані пов'язують сервісні частини з моделі переходів з конкретними адресами ЕС, до яких треба звертатись.

У ЕС, які використовуються сервісними компонентами, містяться сервісна модель проблеми; модель планування виконання запиту; дані, необхідні для виконання запиту. Сервісна модель проблеми містить дані про роботу з вирішення цієї проблеми у минулому. Модель планування виконання запиту містить дані про можливі плани виконання запиту залежно від показників сервісної моделі проблеми та вхідних параметрів виклику запиту. Дані, необхідні для виконання запиту, обробляються після його планування, і результат передається клієнту.

З попереднього опису зрозуміло, що модель проблеми розподіляється між координаційними та сервісними БЗ. Координаційна частина моделі містить основні результати виконання сервісних запитів і є, по суті, підсумком роботи користувача із сервісами системи.

У свою чергу, сервісна модель проблеми відображає історію роботи над вирішенням проблеми у даній сервісній ЕС. Можливо, виникає запитання про проблеми існування кількох моделей проблеми для одного й того ж сервісу у різних ЕС. Відповідь проста – у координаційній моделі проблеми прописується логічна назва тієї сервісної ЕС, до якої має звертатись відпо-

відний сервісний компонент при виконанні запиту користувача. Існування кількох сервісних ЕС для одного й того ж сервісу можливе, але з кожною з них працює чітко визначений набір користувачів, а значить, історія роботи з проблемою зберігається в одній сервісній ЕС. У разі, якщо сервісні ЕС не потребують для своєї роботи історії вирішення проблеми, вони не міститимуть модель проблеми.

3. Принципи розподілення ЕНС

З огляду на особливості структури ЕНС можна зробити висновок, що вона належить до класу ЕС, призначених для виконання кількох типів діяльності. Отже, описаний вище підхід до розподілення ЕС можна застосувати і для ЕНС. Запропоновані нижче принципи її побудови та підходи до організації навчального матеріалу не є єдино можливими. Вони відображають базові елементи ЕНС, потрібні для її функціонування, та ілюстрації запропонованої розподіленої архітектури. Принципи, що лежать в основі цієї архітектури, допускають свободу вибору як в організації структури навчального матеріалу, так і в контролі за його презентацією в подальших її вдосконаленнях.

3.1. Словник термінів ЕНС

Спочатку визначимо словник термінів системи. Курс – упорядкований набір уроків, попереднього та заключного тесту. Попередній тест проводиться перед початком викладання студенту даного курсу. Заключний тест – після викладання курсу. Урок – упорядкований набір з 1 до *n* викладень понять курсу. Поняття курсу – фрагмент знань, що становить узагальнену модель певного елемента предметної області, якій присвячено курс. Викладення поняття курсу – впорядкований набір інформаційних одиниць, кожна з яких відіграє певну роль в ознайомленні студента з поняттям. Основні види ролей інформаційних одиниць – вступ, визначення, приклад, пояснення, висновок. Інформаційна одиниця складається з таких елементів або їх поєднань: текст, зображення, відео, звук. Тест – упорядкований набір запитань. Запитання тесту складається з тексту запитання, варіантів відповіді на нього та переліку правильних варіантів. Надалі наведені терміни вживатимуться саме в такому визначенні.

3.2. Принципи планування процесу навчання в ЕНС

Розглянемо етапи проходження студентом курсу і дані, які використовуються системою для контролю за подачею навчального матеріа-

ду на кожному з етапів. Система надає студентові дві можливості роботи з навчальним матеріалом: урок і тест. Планування подачі матеріалу уроку здійснюється на підставі таких даних: план уроку; результат тесту попереднього уроку; кількість разів проходження студентом даного уроку. Планування подачі тесту здійснюється на підставі такої інформації: план тесту; інформація про варіанти тесту, які студент уже отримав. Складність запитань тесту залишається однаковою для всіх варіантів тесту незалежно від рівня студента.

3.3. Визначення сервісних та координаційних частин ЕНС

Проаналізувавши функції системи, можна виділити сервіси, які надаються студентові відносно незалежно один від одного: робота з уроком – навчання, а також робота з тестом уроку або курсу – тестування. Також можна виділити фрагменти знань, потрібні для роботи лише цим частинам: БЗ тестування та БЗ навчання.

Згідно з таким розподілом на сервісні частини в координаційній частині має знаходитись модель процесу навчання студента (як модель тієї проблеми, що вирішується системою) і модель переходів, яка визначатиме можливі дії студента під час роботи з окремим курсом (або проходження тесту, або проходження уроку). Також необхідно мати відповідний фрагмент моделі процесу навчання студента в кожній сервісній частині. Для тестування такою моделлю стане історія роботи студента з тестами, а для навчання – історія роботи з уроками та курсами.

4. Загальний принцип функціонування РЕНС

На початку роботи з системою студент за допомогою компонента клієнта може надсилати два види запитів до компонента координатора: запит на початок реєстрації студента на курси або запит на отримання списку курсів, на які він уже зареєстрований. Після отримання списку курсів, на які він зареєстрований, студент може надсилати запит на отримання переліку можливих дій, які він виконуватиме на даному етапі вивчення конкретного курсу. Після отримання такого переліку студент обирає одну із запропонованих дій.

Залежно від вибраної студентом дії компонент клієнта звертається із запитом на виконання дії до того сервісного компонента, який надає відповідний сервіс для її виконання, аналізує отриманий від сервісного компонента результат і надсилає координаторові значення параметрів моделі процесу навчання, які змінились унаслідок виконання запиту.

4.1. Огляд структури координаційної та сервісних частини РЕНС

Координаційна частина РЕНС складається з таких елементів: координаційний компонент (координатор) та ЕС, що використовується координаційним компонентом (координаційна ЕС). Координатор забезпечує прозорість доступу до координаційної ЕС, приховуючи від клієнта деталі її реалізації.

Сервісна частина РЕНС складається з сервісів тестування та навчання. Кожен із сервісів складається з таких частин: компонент тестування або навчання (сервісні компоненти) та сервісна ЕС, які використовуються сервісним компонентом. Сервісні компоненти забезпечують прозорість доступу до сервісних ЕС, приховуючи від клієнта деталі реалізації.

5. Опис реалізації РЕНС

5.1. Засоби реалізації

Для побудови РЕНС згідно з вищеописаним підходом були обрані такі засоби: технологія CORBA [5] і мова Java [9] для реалізації сервісних та координаційних компонентів, а також постреляційна СУБД Cache [1] для реалізації ЕС, які використовуються компонентами.

5.2. Загальний опис принципів реалізації ЕС у СУБД Cache

Нагадаємо, що СУБД Cache підтримує три моделі доступу до одних і тих самих даних – реляційний, об'єктний, а також прямий через багатовимірні структури. Завдяки цьому вдалося організувати БЗ таким чином. Знання представляються у фреймоподібних структурах, для зберігання яких використовуються багатовимірні масиви. Розглянемо правила переведення фреймів у багатовимірні масиви Cache.

У глобаліях – багатовимірних розріджених масивах – може бути будь-яка кількість індексів. Для значень цих індексів не передбачено визначення типів даних, і тому вони можуть бути будь-якими: рядками символів, числами з плаваючою крапкою, цілими числами і т. д. Це означає, що різномірні індекси, наприклад ціле число 5 і рядок «Курс Основи програмування», можуть перебувати на одному рівні. Наприклад, може існувати така структура даних:

\wedge COURSE(id курсу, назва курсу, час викладання) = кількість студентів

Приклад конкретних даних:

\wedge COURSE(5, "Основи програмування", "літній триместр") = 32

Пропонуються такі правила для відображення фреймової структури у структуру глобалі:

1. Назва фрейму переходить у назву глобалі.
2. Значення слота 1 – у 1-й рівень глобалі, значення слота *n* у значення рівня глобалі *n*. У разі потреби швидкого пошуку серед значень певного слота його розміщують якомога вище в ієрархії рівнів.
3. У разі, якщо значенням слота виступас набір слотів нижчого рівня, у відповідний рівень глобалі ставиться назва цього набору, а в подальших рівнях – значення слотів з цього набору.

Приклад переведення фреймової структури «Питання тесту» у структуру глобалі Cache.

Фрейм:

```
(Question
  Ідентифікатор питання
  Текст питання
  Варіанти відповідей (Порядковий номер варіанта.
                        Текст варіанта відповіді)
  Номери правильних варіантів відповідей
  Пояснення відповідей (Порядковий номер
                        варіанта. Пояснення, чому він
                        правильний чи неправильний))
```

Відповідні індекси глобалі з прикладами їх наповнення:

```
QUESTION(1)="text of the question 1"
QUESTION(1,"answers",1)= "1-st answer variant for
                          question 1"
QUESTION(1,"answers",2)= "2-nd answer variant for
                          question 1"
QUESTION(1,"correct",2)=""
QUESTION(1,"explain",1)= "explain for 1-st answer
                          variant for question 1"
QUESTION(1,"explain",2)= "explain for 2-nd answer
                          variant for question 1"
```

Відмінність цих структур від фреймів полягає у відсутності приєднаних процедур. Усі знання БЗ можна розбити на дві категорії: знання, що характеризують стан і властивості проблеми, для вирішення якої призначено дану ЕС; а також інший вид знань – метазнання, потрібні для представлення правил обробки знань першої категорії та планування виконання цієї обробки.

Методи обробки БЗ інкапсульовано в методах відповідних об'єктів СУБД Cache. Таким чином, знання з предметної області та правила обробки цих знань повністю відокремлено від

процедур обробки. Такий підхід дає змогу визначати для різних видів однієї і тієї ж проблеми різні структури представлення даних та методи її обробки. Для РЕНС такий підхід до організації БЗ означає можливість конструювання різних за змістом і видом курсів, моделей процесу навчання і для кожного з них визначати власні методи обробки. Тепер розглянемо одну з можливих організацій БЗ для кожного типу ЕС, які задіяні у роботі РЕНС.

5.3. Координаційна ЕС

БЗ координаційної ЕС містить чотири багатомірних масиви (глобалі), які призначені для зберігання таких моделей: моделі процесу навчання, моделі можливих дій (модель переходів), конфігураційної моделі курсу та реєстраційної моделі. Також вона містить об'єкти, методи яких інкапсують методи обробки даних моделей. Розглянемо кожну модель докладніше.

5.3.1. Моделі координаційної ЕС

Модель процесу навчання міститься в глобалі ^STATE. Для неї визначено такі елементи моделі для кожного із студентів по кожному з курсів, які вони слухають: ідентифікатор останнього уроку, який прослухав студент; ідентифікатор тесту для цього уроку; результат проходження тесту для цього уроку; ідентифікатор правил оцінювання для цього тесту, які застосовуються для оцінювання результату тесту; ідентифікатор області дій для зв'язку з моделлю можливих дій; логічна назва навчальної ЕС, яка використовується для роботи з уроками цього курсу; логічна назва тестувальної ЕС, яка використовується для роботи з тестами цього курсу. Структуру глобалі ^STATE представлено в табл. 1.

Наприклад, для студента з ідентифікатором «Piter», для курсу з ідентифікатором «course_1», який він прослуховує, в певний момент часу роботи з системою частина моделі процесу навчання студента для даного курсу може містити такі дані:

```
^STATE("Piter", "course_1", "Test_addr") = "TEST_4"
^STATE("Piter", "course_1", "Learn_addr") =
                                         "LEARN_1"
^STATE("Piter", "course_1", "Result") = 4
```

Таблиця 1. Структура глобалі ^STATE

^STATE						
User_id						
Course_id						
«Test_addr» = назва ЕС тестування	«Learn_addr» = назва ЕС навчання	«Result» = результат тесту	«Test» = id тесту	«Lessons» = id уроку	«Eval» = id правил оцінювання	«Activities» = id області дій

Таблиця 2. Структура глобалі ^ACTIVITIES

^ACTIVITIES					
Activities_id					
«State sequence»	«State_values»		«Operations»		
Порядковий номер = назва елемента моделі процесу навчання	Можливе значення елемента = символ для представлення в рядку	«default» = представлення в рядку значення елемента, якщо для нього не визначено власного представлення	Рядок із символів, який визначає стан моделі процесу навчання		
			Порядковий номер дії		
			«OpName»	«description»	«return type»

Модель дій зберігається в глобалі ^ACTIVITIES. Вона містить інформацію, пов'язану з можливими діями користувачів системи. Для певних курсів може існувати більш розвинений набір дій, ніж тестування та перегляд уроку, наприклад робота з практичним заняттями. Для інших – менш розвинений. Тому визначаються області, де групуються всі можливі дії користувачів при прослуховуванні певних курсів. Для кожної такої області визначаються такі три групи знань, що використовуються для визначення можливих дій студента.

Перша група визначає назви елементів моделі студента, стан яких важливий для вибору дії. Друга група визначає, які значення кожного з цих елементів моделі студента суттєві для вибору дії. Також вона визначає правила інтерпретації цих значень для виконання процедури вибору. Остання група містить відповідність між проінтерпретованими елементами моделі студента і набором можливих дій (сервісів, за якими він може звертатись). Для кожної з цих дій вказується: опис дії, що буде відображено користувачеві; назва операції, тип значення, що повертається, тип сервісного компонента, до якого потрібно звертатись; перелік назв елементів моделі студента, значення яких потрібно передати як параметри; перелік назв елементів моделі студента, змінні значення яких потрібно передати після виконання даної дії координаторові. Структуру глобалі ^ACTIVITIES наведено в табл. 2.

Описані глобалі ^STATE та ^ACTIVITIES є ключовими в роботі координаційної ЕС. Їх організація дає змогу мати в РЕНС одночасно кілька різних моделей користувача, параметри яких застосовуються для роботи різних сервісних ЕС, та визначати набір запитів, з якими компонент клієнта може звертатись до кожної з них.

Наступні дві моделі є допоміжними для координації процесів ініціалізації у моделі студен-

та, даних про новий курс та підтримки реєстрації на курси.

Модель конфігурації міститься в глобалі ^CONF. Вона пов'язує логічні адреси навчальної та тестувальної ЕС з ідентифікатором області можливих дій. Ця модель потрібна для виконання процесу реєстрації студента на курси для призначення його моделі початкових значень логічних назв ЕС тестування та навчання. У глобалі для кожної можливої конфігурації вказується: логічна назва тестувальної ЕС; логічна назва навчальної ЕС; ідентифікатор області можливих дій.

Реєстраційна модель міститься у глобалі ^REGISTRATION. Вона відображає стан реєстрації кожного студента на курси системи. Містить такі елементи: стан реєстрації на курс; ідентифікатори студентів, які перебувають у даному стані роботи з курсом; ідентифікатори відповідних курсів; ідентифікатори конфігурації, які потрібно призначити моделі процесу навчання при реєстрації студента на курс; результати заключних тестів для пройдених студентом курсів.

5.3.2. Об'єкти координаційної ЕС

При аналізі взаємодії студента з координаційною частиною ЕНС були виділені такі запити, які він може надсилати: отримати список курсів, на які зареєстрований студент; отримати список курсів, на які не зареєстрований студент; зареєструватись на курс; отримати перелік можливих дій студента при роботі з певним курсом; поновити модель студента. Логіка виконання цих запитів та необхідні процедури обробки БЗ інкапсульовано в об'єкті Reactor. Для передачі певних типів результатів виконання запитів Reactor використовує допоміжні об'єкти. Сигнатури методів Reactor наведено на рис. 2, а відповідні допоміжні об'єкти, якими

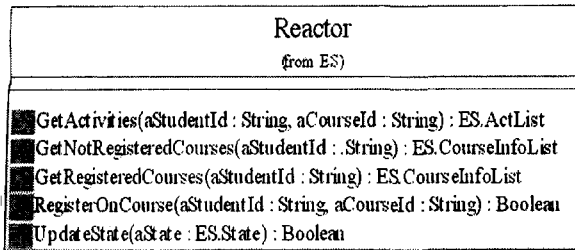


Рис. 2. Сигнатури методів об'єкта Reactor

вони користуються для передачі результату, на рис. 3.

Виклик методів об'єкта Reactor виконується з координаційного компонента, який буде розглянуто в наступному розділі.

5.4. Координаційний компонент (координатор) PEHC

Координатор реалізовано як CORBA-об'єкт. Він призначений для надання прозорого доступу до координаційної ЕС, а також для забезпечення функціонування PEHC на різномірних платформах. Реалізація його інтерфейсів забезпечується імпортованими в Java-об'єкти з Cache об'єктом Reactor та його допоміжними об'єктами.

При зверненні компонента клієнта до однієї з його операцій координатор конвертує параметри запиту клієнта у формат, який потрібний для виклику методів об'єкта Reactor, і викликає відповідний метод. Після отримання результату він конвертує його у форму, потрібну компоненту клієнта, і повертає клієнту.

5.5. Сервісна ЕС навчання

БЗ даної ЕС містить такі елементи: модель студента, що відображає історію роботи студента з ЕС; модель планування навчання; моделі, де визначаються структури для курсів, уроків та понять; знання про курс, про урок та про поняття; а також правила переведення результату проходження тесту попереднього уроку у відповідний рівень для використання при плануванні уроку. Також вона містить об'єкти, що інкапсулюють методи обробки цих моделей.

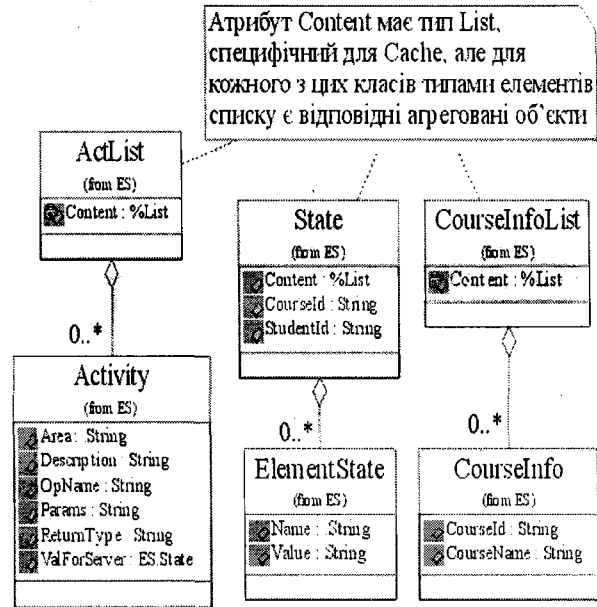


Рис. 3. Допоміжні об'єкти для передачі результатів

5.5.1. Моделі сервісної ЕС навчання

Модель студента відображає історію роботи студента з даною ЕС. Міститься у глобалі ^STUDENTMODELS. Історія роботи складається з трьох частин: історії роботи з курсами, уроками та поняттями. Для кожної з них відмічається число звернень студента для проходження відповідних навчальних елементів. Структуру глобалі ^STUDENTMODELS наведено в табл. 3.

Модель планування навчання містить плани викладення навчального матеріалу залежно від рівня студента та кількості разів прослуховування ним даного навчального матеріалу. Міститься в глобалі ^LEARNPLANS. Складається з трьох частин: планування для викладення курсу, уроку та поняття. Структуру глобалі ^LEARNPLANS наведено в табл. 4.

Модель структури курсу. Для різних планів викладення курсу визначає різну послідовність уроків і спільні для всіх планів ідентифікатори попереднього та заключного тесту курсу. Для тестів також вказуються ідентифікатори правил, за якими буде виконуватись оцінювання, а також правил переведення результатів тесту для попе-

Таблиця 3. Структура глобалі ^STUDENTMODELS

^STUDENTMODELS		
Student_id		
«courses»	«lessons»	«concept»
Course_id =	Lesson_id	Concept_id
Кількість разів прослуховування курсу	Course_id = кількість звернень до уроку в рамках прослуховування курсу	Lesson_id
		Course_id
	Кількість звернень до поняття в рамках прослуховування уроку	

Таблиця 4. Структура глобалі ^LEARNPLANS

^LEARNPLANS					
«courses»		«lessons»		«concept»	
Course_id		Lesson_id		Concept_id	
		Course_id		Lesson_id	
«many»	Кількість разів прослуховування	«many»	Кількість разів прослуховування	Course_id	
Рівень студента = id плану	Рівень студента = id плану	Рівень студента = id плану	Рівень студента = id плану	«many»	Кількість разів прослуховування
				Рівень студента = id плану	Рівень студента = id плану

Таблиця 5. Структура глобалі ^COURSESTRUCT

^COURSESTRUCT					
Course_id					
«plans»		«pretest» = test_id		«aftertest» = test_id	
Plan_id		«eval» = id правил оцінювання	«tolevel» = id переведення результату тесту в рівень студента	«eval» = id правил оцінювання	«tolevel» = id переведення результату тесту в рівень студента
Prev_Lesson_id = наступний урок	«first» = перший урок				

Таблиця 6. Структура глобалі ^LESSONSTRUCT

^LESSONSTRUCT					
lesson_id					
«plans»			«test» = test_id		
Plan_id			«eval» = id правил оцінювання	«tolevel» = id переведення результату тесту в рівень студента	
Prev_Lesson_id = наступний урок	«first» = перший урок				

Таблиця 7. Структура глобалі ^CONCEPTSTRUCT

^CONCEPTSTRUCT	
Concept_id	
Plan_id	
Тип інф. одиниці = тип наступної інф. одиниці	«first» = тип інф. одиниці

Таблиця 8. Структура глобалі ^CONCEPT

^CONCEPTSTRUCT
Concept_id
Unit_type
Order_number = текст інформаційної одиниці

реднього уроку у відповідний рівень студента. Модель міститься в глобалі ^COURSESTRUCT. Структуру глобалі ^COURSESTRUCT наведено в табл. 5.

Викладення матеріалу залежить від рівня студента. Оскільки різні курси можуть визначати різні можливі рівні студента, то у кожній з них є правила для переведення відповідних результатів тесту до своєї шкали рівнів. Результати тесту отримуються від клієнта.

Модель структури уроку. Для різних планів викладення уроку вона визначає послідовність викладення понять, а також ідентифікатор тесту, який має пройти студент після ознайомлення з матеріалом уроку. Міститься в глобалі ^LESSONSTRUCT (табл. 6).

Модель структури поняття. Для різних планів викладення містить різну послідовність типів інформаційних одиниць. Міститься в глобалі ^CONCEPTSTRUCT (табл. 7).

Знання про поняття курсу знаходяться у глобалі ^CONCEPT. Для кожного поняття курсу визначають типи інформаційних одиниць та їх вміст. Саме з них і формується урок для передачі студенту. Структуру глобалі наведено в табл. 8.

Знання про курс та урок. Знаходяться відповідно в глобалі ^COURSE та ^LESSON. Містять назву та опис відповідних навчальних елементів.

Правила, що встановлюють відповідність між результатами тесту попереднього уроку та значеннями можливого рівня студента, знаходяться в глобалі ^TESTLEVEL. Необхідність таких пра-

вил зумовлена тим, що один і той самий тест може використовуватись у рамках різних курсів, де визначено різну кількість рівнів студентів. Тому необхідно мати знання для інтерпретації результатів тесту згідно з існуючими для даного курсу рівнями студента.

5.5.2. Об'єкти сервісної ЕС навчання

При аналізі взаємодії студента з функціональною частиною ЕНС навчання були виділені такі запити, які він може надсилати: отримати наступний урок курсу; отримати урок курсу на докладнішому рівні; отримати опис курсу. Логіку виконання цих запитів та необхідні процедури обробки БЗ інкапсульовано в об'єкті Teaching. Для передачі певних типів результатів виконання запитів Teaching використовує допоміжні об'єкти. Сигнатури методів Teaching такі:

```
GetCourseIntro(StudId : String, CourseId : String,
StateForServer : State) : CourseInfo
GetDeeperLesson(StudId : String, CourseId : String,
ValForServer : State) : Lesson
GetNextLesson(StudId : String, CourseId : String,
PrevLessonId : String, Result : String, ValForServer :
State) : Lesson.
```

На рис. 4 представлено відповідні допоміжні об'єкти, якими вони користуються для передачі результатів методів.

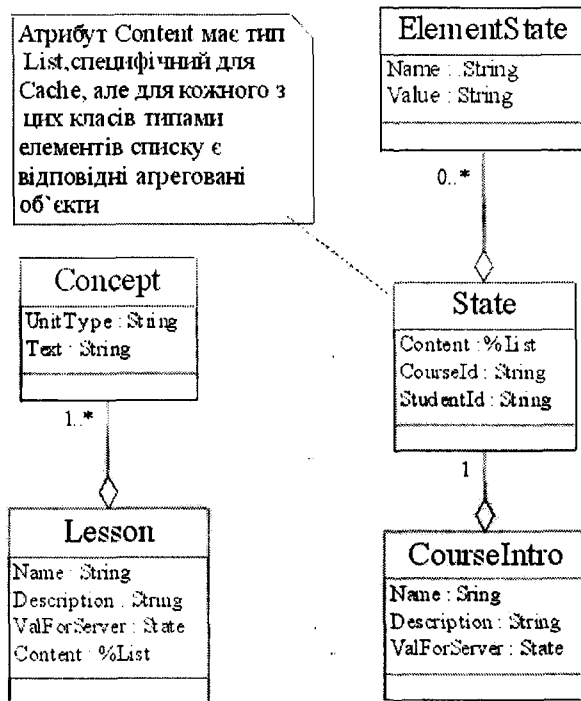


Рис. 4. Допоміжні об'єкти

5.6. Сервісна ЕС тестування

БЗ даної ЕС містить такі елементи: модель студента, що відображає історію роботи студен-

та з ЕС; модель планування тестування; моделі, де визначаються структури для тестів; знання про запитання, а також знання про методи оцінювання тестів. Також вона містить об'єкти, що інкапсують методи обробки цих моделей.

5.6.1. Моделі сервісної ЕС тестування

Модель студента відображає історію роботи студента з даною ЕС. Для кожного студента визначено ідентифікатори тестів, які він проходив, та їх варіанти. Міститься у глобалі ^TESTMODEL (табл. 9).

Таблиця 9. Структура глобалі ^TESTMODEL

^TESTMODEL
Student_id
Testl_id
Кількість разів проходження
Номер варіанта

Модель планування тестування. Ця модель кожному тесту ставить у відповідність кількість разів його проходження студентом з номером варіанта тесту, який видається. Якщо кількість разів не знайдено в моделі, використовується варіант, вказаний у полі «many». Міститься в глобалі ^TESTPLAN (табл. 10).

Таблиця 10. Структура глобалі ^TESTPLAN

^CONCEPTSTRUCT
Test_id
Plan_id
«many» = номер варіанта
Кількість разів проходження = номер варіанта

Модель структури тестів призначено для визначення наповнення конкретного варіанта конкретного тесту запитаннями та їх порядку. Міститься в глобалі ^TEST (табл. 11).

Таблиця 11. Структура глобалі ^TEST

^TEST
Test_id
Variant number
Порядок = Id питання

Модель структури запитання призначена для зберігання таких знань про запитання тесту: текст запитання; можливі варіанти відповідей на запитання; номери правильних варіантів відповідей; пояснення для кожного з варіантів, чому він є правильною або неправильною відповіддю на текст запитання. Модель міститься в глобалі ^QUESTION.

Знання про правила оцінювання тестів. Ця модель містить можливі правила для оцінки тестів. Вона визначає оцінку за кожний правильний варіант відповіді і за кожний неправильний варіант відповіді. Залежно від цього результат проходження одного й того ж тесту з однаковим набором відповідей може різнитись. Модель міститься в глобалі ^EVALUATION.

5.6.2. Об'єкти сервісної ЕС тестування

При аналізі взаємодії студента з функціональною частиною ЕНС тестування був виділений один із запитів, який він може надсилати: отримати тест.

Логіка виконання запиту та необхідні процедури обробки БЗ інкапсульовано в об'єкті Testing. Для передачі результату виконання запиту Testing використовує допоміжні об'єкти. Сигнатура методу Testing:

```
GetTest(StudentId : String, TestId : String,
        ValForServer : State) : Test
```

Відповідні допоміжні об'єкти, які використовуються для передачі результату, наведено на рис. 5.

5.7. Сервісні компоненти навчання та тестування РЕНС

Сервісні компоненти навчання та тестування реалізовано як CORBA-об'єкти. Вони призначені для надання прозорого доступу до ЕС навчання та тестування відповідно, а також для забезпечення функціонування РЕНС на різномір-

них платформах. Вони надають інтерфейси для виконання дій відповідних сервісних ЕС. Реалізація їх інтерфейсів забезпечується імпортованими в Java-об'єкти з Cache об'єктами Teaching та Testing відповідно, а також їх допоміжними об'єктами.

Клієнт обирає тип сервісного компонента, до якого йому належить звернутися з отриманих від координатора параметрів можливих дій. Тип результату, що буде повернено, також відомо. Він надсилає запит про виконання дії (DoActivity) відповідному сервісному компоненту. Сервісний компонент за переданими параметрами визначає, до якої сервісної ЕС звертатись і які методи відповідного сервісного об'єкта ЕС викликати.

Результат, що повертається клієнтові, завжди містить у собі значення параметрів (ValForServer) для зміни стану моделі студента в координаційній ЕС. Отримавши результат, цю частину клієнт надсилає до координатора, а інші значення аналізує для видачі користувачеві. У сервісного компонента тестування є одна особливість. Він містить тип-значення (valuetype) Test, який має операцію виконання оцінювання тесту. Цей тип-значення передається клієнтові, який може викликати певну операцію для оцінки результатів і вже потім передавати їх координаторові.

Висновки та рекомендації

З проведеного дослідження використання розподілених експертних систем (РЕС) у системах комп'ютерної підтримки дистанційної освіти можна зробити такі висновки. Для побудови незалежних від апаратної платформи, надійних та продуктивних РЕС доцільно використовувати технологію CORBA, як найуніверсальніший та найповніший стандарт порівняно з головними її суперниками – RMI та COM. Існуючі на сьогодні способи збереження даних розподілених систем (РС) за допомогою реляційних або об'єктно-орієнтованих СУБД мають певні недоліки, від яких страждає загальна швидкість РС. Проте використання постреляційної СУБД Cache може, по-перше, дозволити застосовувати обидва вищеназвані способи представлення даних, і, по-друге, забезпечити високу швидкість обробки цих даних. З огляду на вищесказане, варто рекомендувати використання в побудові РЕС стандарту CORBA, СУБД Cache, а також платформонезалежної мови Java.

Розгляд особливостей експертних систем привів до висновків про тісний взаємозв'язок їх структурних компонентів, розподілення яких викликало б значні витрати на забезпечення обміну повідомленнями між ними. Тому ми вирішили розглянути ЕС під іншим кутом зору –

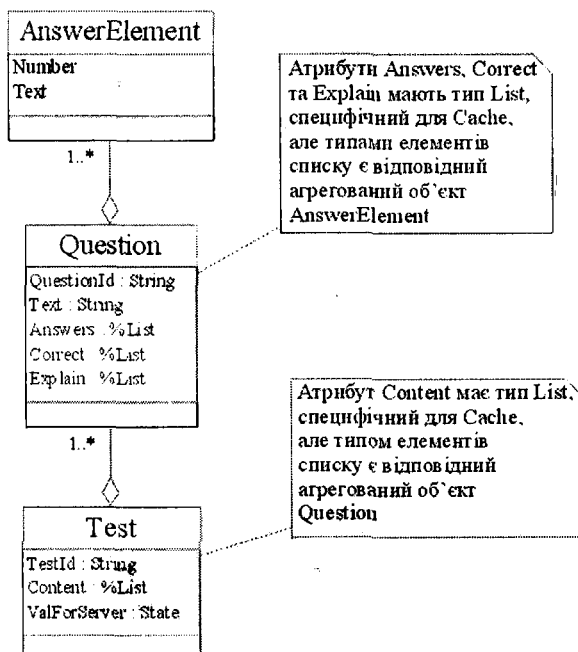


Рис. 5. Допоміжні об'єкти

функціональним. Такий погляд на ЕС привів до ідеї розподілення її на два основних типи частин: координаційну та сервісні. Перша відповідає за управління ходом вирішення проблеми та збереження поточного стану її вирішення у вигляді кінцевих результатів роботи сервісних частин. Також передбачається існування клієнтської частини в розподіленій експертній системі, за допомогою якої користувач звертається до координаційної частини для отримання набору можливих запитів, з якими він може звертатись до сервісних частин ЕС з метою отримання наступного проміжного результату розв'язку проблеми. певні параметри цього результату клієнт передає координаційній частині для оновлення даних про стан вирішення проблеми.

При застосуванні цього принципу на практиці у складі координаційної частини виділяються координаційні ЕС та компонент доступу (координатор) до них. Координатор реалізується як CORBA-об'єкт для забезпечення максимально прозорої взаємодії між клієнтською частиною та координаційною ЕС. У свою чергу, до

складу сервісної частини також входять сервісний компонент доступу та сервісні ЕС. Клієнтський компонент звертається виключно до компонентів доступу відповідних ЕС, які приховують деталі реалізації ЕС.

Такий підхід дає змогу використовувати різнотипні ЕС у складі PEC. Також він зменшує витрати на взаємодію між розподіленими частинами PEC завдяки обміну лише результатами їхньої роботи, а не знаннями з їхніх БЗ. Користувачам надається можливість контролювати вирішення проблеми на етапах проміжних результатів.

Майбутні роботи у цьому напрямі варто було б присвятити вдосконаленню прототипу PENC, в тому числі розробці компонентів для повторного використання та інструментальних засобів для наповнення БЗ, з метою створення оболонки PENC для дистанційної освіти, що здатна підтримувати різні моделі навчального процесу, в тому числі різні моделі учня та структури курсів у складі системи дистанційної освіти.

1. Кирстен В., Ингер М., Рериг Б., Шульте П. СУБД Cache: объектно-ориентированная разработка приложений. Учебный курс - СПб.: Питер, 2001 - 384 с.
2. Компьютерная технология обучения. Словарь-справочник. В 2-х томах - К.: Наукова думка, 1992.-641 с.
3. Петрушин В. А. Экспертно-обучающие системы- К.: Наукова думка, 1992- 196 с.
4. Уотермен Д. Руководство по экспертным системам- М.: Мир, 1989-388 с.
5. Цимбал А. Технология CORBA для профессионалов- СПб.: Питер, 2001-624 с.
6. Эммерих В. Конструирование распределенных объектов- М: Мир, 2002- 510 с.
7. Brusilovsky P. Adaptive and Intelligent Technologies for Web-based Education. II Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 1999, 4, 19-25 (<http://www2.sis.pitt.edu/~peterb/papers/KI-review.html>).
8. Brusilovsky P., Ritter S., Schwarz E. Distributed intelligent tutoring on the Web. In Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August, 1997.
9. Deitel, H. M. Java: How to program-New Jersey: Prentice Hall, 1998.-С 1063.

М. М. Gtybovets, А. А. Danchenko

ABOUT ONE APPROACH OF USING DISTRIBUTED TUTORING EXPERT SYSTEMS IN DISTi J4CE EDUCATION

In the article one approach of the distribution of one class of expert systems was proposed and its application in distance education systems was described.