

РОЗРОБКА ТЕХНОЛОГІЧНОЇ АРХІТЕКТУРИ АДАПТИВНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ

к.т.н., доцент каф.ОТ, ВНТУ О.І.Гороховський, асп. ВНТУ Т.І.Трояновська

Вступ

Автоматизовані системи дистанційного навчання зазвичай технологічно не йдуть далі форми електронного підручника разом із певною системою тестування, яка містить певний елемент випадковості, завдяки якому й забезпечується так звана, індивідуальність навчання. Складання електронних підручників, на відміну від традиційних академічних освітніх видань, може бути нелінійним, і відтак теж забезпечити свободу дій студента у процесі навчання. Однак ця свобода дій зазвичай ніяк не відображається ані в системі тестування, ані в інших системах контролю якості навчання. Незалежно від того, в якому саме порядку студент проходив курс, або яку частину його переглядав уважно, а яку – поверхнево, контрольні елементи лишаються тими самими. Одним із способів визначення, наскільки студент вивчив тему, в такому випадку є комплексне тестування, яке складається із таких питань, відповіді на які можуть дати апостеріорний (тобто по завершенню навчання) аналіз пройденого ним шляху. При цьому дати студенту можливість покращити свою оцінку можна тільки за допомогою повторного проходження ним курсу дистанційного навчання.

Суттєвим способом зняти проблему апостеріорного аналізу є перехід від автоматизованої системи до адаптивної. Фактично адаптивна система є подальшим розвитком автоматизованої системи, яка доповнюється модулем (або набором модулів) зворотного зв'язку, яка дозволяє оперативно коригувати процес навчання студента в реальному часі. Дана стаття присвячена аналізу існуючих засобів забезпечення такого зворотного зв'язку.

Мета: аналіз технологічних можливостей персоналізації курсу дистанційного навчання для підвищення показників якості навчання (Quality of Education) за допомогою засобів динамічного формування контенту.

Постановка задачі: аналіз засобів для врахування особливостей студента, які впливають на динаміку навчання та сприймання матеріалу; розробка архітектури для забезпечення персоналізації курсу.

Персональний фактор в процесі навчання

Для зручності на початку введемо два основних терміни.

Академічною або традиційною моделлю навчання будемо вважати статичну модель організації навчального процесу, яка використовується в освітніх установах – школах, вузах, спеціальних училищах тощо. Адаптивною моделлю вважатимемо динамічну модель, яка заснована на зворотному зв'язку в процесі навчання.

Академічна модель персональний фактор взагалі ніяк не враховує. Якщо узяти у якості прикладу підручник, то поділ на окремі «поняття», розділи та теми відбувається незалежно від того, як саме його сприйматиме той чи інший студент, а базується тільки на інформаційній та логічній завершеності окремого фрагменту курсу. Крім того, часовий фактор при компоновці навчальних матеріалів в цьому випадку теж практично не аналізується, замість нього використовується константа – академічна година, на яку розраховується подача матеріалу. Це рішення цілком доцільне

для стаціонарних та напівстаціонарних форм навчання, однак для дистанційних форм навчання є цілковито хибним [1].

Поняття предметно-орієнтованої домінанти (ПОД) [2] передусім базується на понятті уваги студента. При дистанційному навчанні увага студента сконцентрована виключно на навчальних матеріалах за відсутності основного концентруючого увагу фактора – викладача. Під час академічної форми навчання викладач виконує роль додаткового чинника, який викликає у студента сконцентрованість на розглядуваній темі, а відтак підвищує увагу і рівень сприйняття матеріалу. У дистанційній формі фізичної присутності викладача немає, відтак і єдиним фактором, який може підвищити увагу студента, є сам матеріал.

Для підвищення якості навчання в дистанційній формі увага студента повинна бути максимізована у кожній окремо взятій частині курсу.

Автоматизовані системи дистанційного навчання вирішують цю проблему за допомогою мультимедіа-вставок, які додають до суто теоретичного матеріалу наочність, практичні завдання, які дозволяють студенту до певної міри на практиці опанувати пропоновану тему, і таке інше. Однак такі методи теж розраховані на деякий усереднений проміжок часу, за який студент теоретично повинен розглянути теоретичний матеріал та виконати певні практичні заняття. А насправді цей часовий проміжок часу може бути різним і дуже сильно варіюватись в залежності від особистих даних студента.

Припустимо, що студент А має змогу приділити навчанню 20 хв., матеріал при цьому розрахований на 45 хв. (академічна година). В зв'язку із цим пропонується відмовитись від класичного планування курсу дистанційного навчання із усередненим розрахунком сприйняття матеріалу і перейти до персоналізованого планування, де компоновка кожного розділу залежить від особливостей кожного окремо узятим студента.

Зазвичай, не існує практичної можливості створення системи, яка б автоматично планувала курс по існуючому теоретичному та практичному матеріалу. Пропонується створити часткове планування, засноване на понятійній компоновці.

Визначення: Понятійна компоновка – розбиття теоретичного матеріалу на окремі «поняття», які є логічно та інформативно завершеними.

Введемо поняття *мінімального часу сприйняття матеріалу* T_{mat} . Цей показник складається із швидкості читання тексту t_{text} і середньої швидкості виконання практичного завдання t_{motor} .

$$T_{mat} = t_{text} + t_{motor} \quad (1.1)$$

Будемо вважати, що матеріал складається із понять. Поділимо всі можливі поняття в теоретичному матеріалі на дві категорії, скориставшись для цього математичними термінами – *аксіоми* та *теореми*.

Як відомо, аксіома – це вислів, що не потребує доведення.

Аксіомою будемо називати таке поняття, яке не вимагає практичного або наочного засобу для полегшення його сприйняття. Для такого поняття мінімальний час сприйняття матеріалу дорівнює:

$$T_{mat} = t_{text} \quad (1.2)$$

В даному випадку, як це зрозуміло, $t_{motor} = 0$ а тому в вираз відповідний доданок не включається.

Як відомо, теорема потребує доведення і базується на аксіомах.

Теоремою будемо називати таке поняття, яке вимагає практичного або наочно-го засобу для полегшення його сприйняття. В такому випадку формула мінімального часу сприйняття приймає свій повний вигляд:

$$T_{mat} = t_{motor} \quad (1.3)$$

Будемо вважати, що теоретичний матеріал складається із аксіом та теорем. Чі-льною особливістю такої компоновки полягає в тому, що одна аксіома та ряд теорем формують логічну послідовність, яка утворює логічно та функціонально завершене «*поняття*». Відповідно, мінімальний час сприйняття «*поняття*» дорівнюватиме сумі всіх мінімальних часів сприйняття, відповідних тим аксіомам та теоремам, які вхо-дять до нього:

$$t_{поняття} = \sum_{T_{mat}, i = \overline{1, n}}^i \quad (1.4)$$

Використовуючи цей показник, а також такий залежний від особистості студе-нта параметр як *мінімальний час навчання*, можна вважати, що можлива динамічна компоновка фрейму [3] для кожного студента шляхом вирішення «задачі покриття» [...]. Нехай існує деякий курс, до складу якого входить 4 лекції, кожна з яких, в свою чергу, містить пояснення для 4 логічно завершених понять. Складемо таблицю мінімальних часів для цих понять (в хвиликах).

Таблиця 1 – Таблиця мінімальних часів

Лекція \ Поняття	1	2	3	4
1	10	20	5	20
2	10	10	20	5
3	20	25	10	10
4	20	20	30	25

Варіюючи подачу понять в залежності від часу, який студент може приділити навчанню, ці чотири лекції легко можна перетворити на вісім, десять, або більше лекцій, включаючи до динамічного фрагменту фрейму необхідну кількість понять. На студента при цьому накладається лише одне обмеження: він повинен приділяти навчанню часу ($t_{навч}$) принаймні стільки, скільки складає максимальне значення мінімального часу сприйняття «*поняття*» в даному курсі.

$$t_{навч} \geq \max(T_{mat}^1, T_{mat}^2, \dots, T_{mat}^n), n = \overline{1, n} \quad (1.5)$$

Такий підхід дозволяє значно підвищити якість сприйняття матеріалу, який по-дається не за усередненими показниками, а враховує особливість процесу навчання конкретно узятю студента.

Іншим аспектом академічної компоновки є усереднення курсу в цілому, і це стосується як теоретичного матеріалу, так і контролю знань. Курси розраховані на певну стандартну послідовність, яка диктує структуру лекцій та практичних занять, при цьому, якщо студент якийсь «*поняття*» розглядав глибше, ніж це вимагається стандартною послідовністю, це ніяк не впливає на контроль знань, а відтак немож-ливо відслідкувати, наскільки студент *реально* оволодів тематикою курсу. Вище ви-значений термін «*поняття*» фактично представляє собою кортеж виду:

$$H = \langle T, P \rangle, \quad (1.6)$$

де T – теоретичний матеріал, а P – практичний.

При цьому елемент контролю знання (наприклад, тест) будується поза залежністю від того, які саме теми і «поняття» були студентом переглянуті. Таким чином, контролюється не загальна якість навчання в межах даного курсу, а часткова, не враховуючи ні самостійну роботу студента, ні випадки, коли через неповне розуміння теми були переглянуті додаткові матеріали. Це можна виявити тільки за допомогою додаткових питань в апостеріорній формі.

Для того, щоб створити відповідність між тим, які саме «поняття» були переглянуті студентом, і генерувати тест, адекватний процесу його навчання загалом та ознайомлення з конкретним теоретичним матеріалом зокрема, до «поняття» слід включити додатковий компонент, який умовно назовемо «вектором контрольних елементів», які надалі трансформуватимуться в питання за допомогою засобів DCD [...]. Аналогом такого додаткового компоненту є набір контрольних запитань в кінці кожної теми в підручнику традиційної академічної компоновки. Різниця полягає в тому, що при академічній моделі такі питання формулюються по темі в цілому, а при запропонованій вони формулюються по кожному «поняттю» окремо.

Таким чином, «поняття» ми можемо тепер сформулювати як кортеж виду:

$$H = \langle T, P, C \rangle, \quad (1.7)$$

де C - контрольний елемент; $C = (C_1, C_2, \dots, C_K), K = \overline{1, n}$

Зазначимо, що кількість контрольних елементів, які асоціюються з даним «поняттям», може бути довільною і залежить від сутності «поняття», для якого формулюються елементи.

Розглянемо принципи формування підсумкових та проміжних тестів для студента, які будуть адекватними його процесу навчання. Нехай існує деякий курс з теоретичних T понять, переглянутих і засвоєних студентом курсів H_1, H_2, \dots, H_k . Для кожного з них, відповідно існує множина контрольних елементів:

$$C^t = (C_1^t, C_2^t, \dots, C_K^t), t = \overline{1, T}, K = \overline{1, n} \quad (1.8)$$

Для того, щоб сформулювати тест, необхідно і достатньо, щоб множина питань в підсумковому тесті C_{nidc} була підмножиною загальної множини контрольних елементів:

$$C = (C^1, C^2, \dots, C^t), t = \overline{1, T}, \quad (1.9)$$

причому $C_{nidc} \in C$, така, щоб для кожної підмножини C^t :

$$M(C_{nidc} \cap C^t) \geq 1, t = \overline{1, T},$$

де M - потужність множини (кількість елементів у множині).

Формально, в підсумковому тесті C_{nidc} повинно міститись принаймні 1 питання, згенероване по контрольному елементу із кожного з понять, переглянутих студентом.

Зазначимо, що подібний підхід може призвести до надмірної комплексності тесту, який у випадку великих і довгочасних курсів може перевищити будь-який доступний студентові час, який він може присвятити навчання. В такому випадку можна застосувати два підходи: або розбивати весь тест на окремі фрагменти, які проходитиме студент щодня із таким розрахунком, щоб тестування охоплювало не лише щойно пройдені теми, а й ті, які були пройдені ним; або застосувати систему ваг понять, що дозволить формувати більш короткі і більш відповідні навчальному рівню студента тести.

Для навчального процесу не кожне «поняття» є важливим. Існують такі «поняття», які пояснюються в межах даного курсу як основні, значущі для майбутніх тем, або для роботи користувача в сфері, до якої відноситься курс; інші «поняття», відповідно, грають роль доповняльних. В такому ракурсі кожному поняттю має бути співставлена відповідна вага, по градуйованій шкалі значущості в межах даного курсу. Емпірично можна визначити мінімальні ваги підсумкових контрольних елементів для різних варіантів складності, і тоді можна модифікувати рівняння 1.9, виключивши з нього останню умову і замінивши її ваговим рівнянням.

$$C = (C^1, C^2, \dots, C^t), t = \overline{1, T},$$

причому $C_{\text{нідс}} \in C$, така, щоб для кожної підмножини C^t :

$$W(C_{\text{нідс}}) \geq W_{\min} \quad (1.10)$$

Додаючи до вагового розподілу часовий фактор (час, необхідний для відповіді на питання), отримуємо, як і у випадку з розподілом понять, «задачу покриття» з двома параметрами, розв'язки якої дадуть нам множину тестів, які будуть адекватні процесу навчання студента.

Зазначимо, що в даному випадку існує три різних стратегії розв'язку цієї задачі, які відповідають різним ступеням персоналізації курсу дистанційного навчання.

1. Стратегія «часових рамок». В цьому випадку на вирішення задачі покриття по рівнянню 1.10 накладається додаткова умова:

$$t(C_{\text{нідс}}) \leq t_{\text{навч}} \quad (1.11)$$

Ця стратегія дозволяє максимально пристосувати процес навчання до часових рамок, в яких знаходиться користувач. Така стратегія прийнятна для створення систем СДН для так званого «бізнес-класу» користувачів, які в силу своєї основної зайнятості можуть витратити небагато часу на свою освіту.

2. Стратегія «вагових рамок». В цьому випадку додаткових умов не накладається, і жорстко виконуються умови рівняння 1.10. При цьому, якщо оптимальний розв'язок задачі покриття перевищив $t_{\text{навч}}$, користувач може бути сповіщений про це, і внести зміни в свій графік для проходження підсумкового тесту. Така стратегія прийнятна для створення систем дистанційного навчання (СДН) для людей із гнучким непостійним графіком (вільний графік основної роботи тощо), і для яких основним пріоритетом є максимізувати власні знання з обраного предмету.

3. Стратегія «коливання». В цьому випадку зберігаються умови рівняння 1.10 разом із додатковою умовою, однак додається два параметри, які мінімізуються в залежності від того, в яких умовах на даний момент тестування знаходиться користувач. Ці додаткові параметри називаються максимально допустимим відхиленням:

- по часу: $\Delta t_{\text{навч}} = |t(C_{\text{нідс}}) - t_{\text{навч}}|$;
- по ваговим коефіцієнтам:

$$\Delta W = |W(C_{\text{нідс}}) - W_{\min}|, \quad (1.1)$$

де $W(C_{\text{нідс}})$ - сума всіх ваг в даному тесті, W_{\min} - мінімальна вага, що встановлена викладачем в даному тесті.

При розв'язку «задачі покриття» згідно цієї стратегії також вираховуються два параметри:

$$\text{Max}(\Delta t) \text{ і } \text{Max}(\Delta W) \quad (1.2)$$

Ці параметри використовуються у якості інформативних, наскільки слід користувачеві переглядати свої особисті пріоритети – або вкластись в часовий графік, або якнайповніше проконтролювати власні знання.

Вводячи (1.13) до ПОД студента, як елемент автоматичної звітності або частину профілю, можна варіювати відхилення часового графіку або якісного контролю освіти, як автоматично на основі деякої системи прийняття рішень, так і в напівавтоматичній формі з боку викладача (т'ютора). А тому ця стратегія доцільна для напівакадемічної форми навчання, де існує постійне спілкування групи студентів СДН із своїм т'ютором.

Реалізація будь-якої з вище наведених стратегій вимагає переходу від автоматизованих систем, які відповідають лише за доставку студенту необхідної для самостійної роботи матеріалів та практичних вправ по закріпленню вивчення, до адаптивних систем, які також містять засоби для збору даних як по загальній (статичній) складовій ПОД [2], так і по динаміці зміни окремих параметрів ПОД (динамічна частина) [4], а також засоби для забезпечення зворотного зв'язку під час самого процесу навчання у вигляді прямих та непрямих оцінок діяльності студента [5].

Перехід від автоматизованої до адаптивної архітектури СДН

Основна відмінність автоматизованої системи дистанційного навчання від адаптивної полягає в двох характеристиках. Перша – наявність зворотного зв'язку, тобто на процес прийняття рішень впливає не лише попередньо внесені дані про студента та його особливості, а й сам процес його навчання. І друга – відсутність однозначної залежності переходів між станами системи.

Отже, дамо визначення *адаптивної* СДН як такої автоматизованої СДН, де за допомогою зворотного зв'язку реакція системи змінюється відповідно до поточного стану користувача.

Розглянемо автоматизовану систему як набір певних станів $S_1..S_N$, де під станом будемо розуміти будь-яку реакцію системи на дії користувача під час проходження курсу. Наприклад, таким станом може бути видача йому чергового фрейму, або проведення контрольного тестування. Така модель може бути показана у вигляді таблиці, де можливість переходу показана одиницею, або нулем.

Таблиця 2 – Таблиця переходів по курсу

	S ₁	S ₂	S ₃
S ₁	x	1	0
S ₂	0	x	1
S ₃	1	0	x

В цьому прикладі показано найпростішу схему автоматизованої СДН, де можливі тільки однозначні переходи. Така схема станів характерна для таких систем, де прийнята архітектура академічного «електронного підручника».

Натомість, в адаптивних системах за допомогою систем зворотного зв'язку між переходами жорсткої залежності немає, що й дає користувачу певний ступінь свободи, а також враховує особливості конкретного моменту часу, в якому знаходиться користувач, та його поточні результати по ряду оціночних параметрів. Для цього вводиться додаткове поняття «ваги переходу». Нехай існує множина характеристик ПОД $G = G_1, G_2, \dots, G_t, t = \overline{1, n}$, кожна з яких має якесь числове значення. Тоді для кожного стану слід співставити деяке значення:

$$W(S_i) = W(G^i), i = \overline{1, t}, G^i \in G \quad (1.3)$$

Згідно формули (1.14) на кожний стан впливає тільки підмножина характеристик, і це цілком логічно, оскільки кожний стан системи теж має свою специфіку та обмеження. Тепер, якщо існує такий стан, що

$$S_k \rightarrow \{S_{k+1}^1, S_{k+1}^2, \dots, S_{k+1}^i\}, k, i = \overline{1, t-1}, \quad (1.4)$$

то для кожного елементу підмножини S_{k+1} існують такі характеристики, які використовуються для визначення, який саме перехід відбудеться в той чи інший момент часу:

- $M(G_j^i), G^i \in G, j = \overline{1, i}$ (потужність підмножини характеристик);
- $W(S_{k+1}^i), k, i = \overline{1, t}$ (вага стану, в який відбувається перехід).

Зазначимо, що для такого загального випадку очевидно, що існує два варіанти:

1. Для двох будь-яких станів $M_1 = M_2$

Таку адаптивну систему будемо називати *регулярною*, оскільки в такому випадку всі переходи залежні від однакового числа параметрів, а отже, вибір переходу диктується максимізацією ваги наступного стану. Умову переходу в такому варіанті характеризує наступний вираз:

$$\text{Max}(W(S_{k+1}^i)) \rightarrow S_{k+1} \quad (1.16)$$

2. Для двох будь-яких станів $M_1 \neq M_2$

Таку адаптивну систему будемо називати *варіабельною*, і для визначення наступного стану використовується кілька різноманітних алгоритмів. Найбільш загальним із них є метод максимальних ваг [6]. Полягає він в тому, що крім числового значення характеристики G_i їй також співставляється вага $W(G_i)$, відповідно, при такій схемі перехід відбувається за умови максимальних значень ваг характеристик ПОД і максимальної ваги самого фрейма:

$$\text{Max}(\sum_{j=0}^r W(G_j^i)) \cup \text{Max}(W(S_{k+1}^i)) \rightarrow S_{k+1} \quad (1.17)$$

Перехід до такої схеми архітектури системи дистанційного навчання не є складним процесом. Для цього необхідно і достатньо створити тип даних, який містив би у собі множину $G = G_1, G_2, \dots, G_t, t = \overline{1, n}$ - т. з. *профіль студента*, і за допомогою підсистем зворотного зв'язку – систем оціночних параметрів (прямих та непрямих), наприклад, або можливості ручної зміни ряду характеристик самим користувачем, забезпечити зміну цих параметрів в процесі навчання.

Технологічні засоби реалізації адаптивності

Технологічні засоби переходу від автоматизованої до адаптивної системи дистанційного навчання передбачають насамперед перехід від статичного формування контенту до динамічного.

На поточний час більшість СДН побудовані на статичних сторінках стандарту HTML, які розпізнаються більшою частиною броузерів однаково, і таким чином дозволяють користувачам практично із будь-якою платформою займатись дистанційним навчанням без будь-яких обмежень. Однак така схема відрізняється надто низькою гнучкістю і не дозволяє пристосовувати систему до соціопсихологічних особливостей процесу навчання студентів. Тому було проведено дослідження щодо інших технологічних засобів забезпечення СДН, які б містили необхідні засоби для реалізації адаптивної системи дистанційного навчання.

1. JSP – Java Server Pages, і Servlet – модулі обробки HTML-форм

Подібні технології дозволяють включати до HTML-тексту сторінки, яка входить до СДН, програмний код, який при запиті сторінки користувачем, на льоту перетворюється на HTML-код. У цьому випадку користувач отримує статичний універсальний код, який розпізнається будь-яким браузером, отже сумісність зберігається. Така динамічність стосується лише формування кінцевого файлу, який пересилається користувачу. Зворотний зв'язок при такій схемі обмежується сервлетом, тобто модулем, який у якості параметру приймає форму, заповнену користувачем, і відповідно до неї конструює по заданому шаблону HTML-сторінку (див. рисунок 1) і вносить зміни до бази даних досвіду.

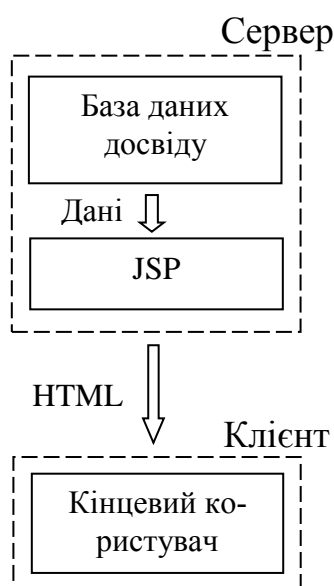


Рисунок 1 - Схема функціонування JSP

База даних досвіду – це набір всіх характеристик, які містяться в профілі студента і використовуються для корегування ПОД студента.

За такої схеми неможливо реалізувати динамічну частину ПОД [4], зокрема, непрямі оцінки [5], оскільки вони вимагають постійного збору даних на клієнтській частині і пересилання на сторону сервера. В цьому випадку пересилання даних можливе тільки по якійсь стандартній події – натискання кнопки, наприклад. Слід також сказати, що форми в технології JSP – теж статичні, тобто мають наперед визначену структуру, зміни в якій вимагають також змін в програмному код і для динамічної схеми формування підсумкових тестів така схема зовсім непридатна.

2. JavaScript AJAX

Використання JavaScript для динамічного формування HTML-сторінок теж можливе, але недоцільно із міркувань ефективності – модель цієї мови в різних браузерах відрізняється, а відтак – існують принаймні два фактори, які затруднюють формування сторінки: 1) необхідність відстеження конкретної моделі браузера, якими користуються студенти, та враховування різних їх типів безпосередньо у коді; 2) порівняно великий час формування сторінки, оскільки формується вона не на сервері, а на клієнтській машині користувача. Однак JavaScript можна використати для створення системи збору даних, яка забезпечить необхідні дані для динамічної частини ПОД, а саме – дані, критичні для непрямих оцінок, такі як час, лабільність (рі-

вень уваги під час читання тексту, або виконання практичних робіт чи тестів) [2], і таке інше. Досягається це за рахунок можливості використання локального об'єкту таймера, за яким запускається модуль збору необхідних даних по сторінці та поточного стану броузера, по цим даним формується «конверт даних» (data envelope XML-Remote Process Controlling) в форматі XML і за допомогою об'єкту XMLHttpRequest [7] в фоновому режимі надсилається на сервер, де може бути оброблене і занесене до бази даних досвіду.

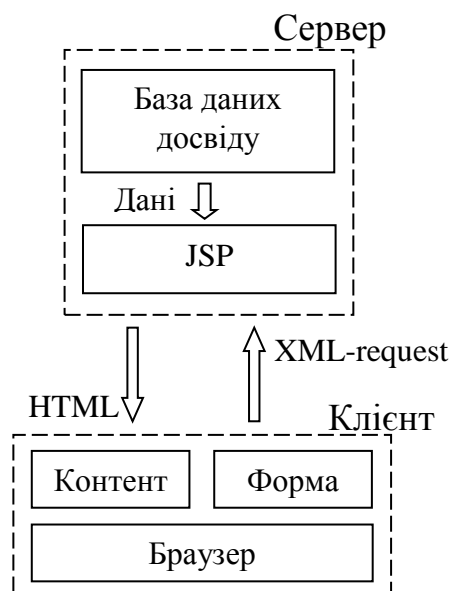


Рисунок 2 - Включення до схеми адаптивної СДН JavaScript AJAX

Для побудови адаптивної системи також критично, що надсилання та отримання даних відбувається саме у фоновому режимі, що знімає необхідність в перезавантаженні машини, і з'являється можливість вносити зміни в контент сторінки безпосередньо в процесі роботи студента із матеріалом. Це дозволяє значно збільшити рівень динамічності контенту, а відтак – і пристосування до соціопсихологічних особливостей студента.

3. Document Content Description (DCD)

У обох вище перелічених технологіях існує один суттєвий недолік. Всі вони передбачають *статичність форм*. Це, в свою чергу, означає, що будь-які види взаємозв'язку із серверною частиною регламентуються наперед заданими схемами, змінити які без втручання в код надзвичайно важко. Тому для досягнення динамічності слід створити якийсь спосіб задавати структуру форм динамічно, і таким чином: 1) створити можливість пристосовувати підсумкові тести до особливостей проходження студентом курсу, як показано вище; 2) мінімізувати втручання в код «движка» системи при внесенні змін до структури бази досвіду.

Для цього може бути використаний стандарт DCD, розроблений у 1998 р. як частина розробки системи метаданих для XML. Перевага цього стандарту полягає у тому, що за допомогою дуже простого коду він дозволяє формально описати будь-яку структуру даних, включаючи групування та контроль типів. В свою чергу це дозволяє формально описати будь-яку форму, і таким чином створити механізм формування підсумкових тестів за DCD-описом. Приклад такого опису приведено нижче.

```

<DCD>
  <ElementDef Type="Booking" Model="Elements" Content="Closed">
    <Description>Описує розташування користувача в класі</Description>
    <Group RDF:Order="Seq">
      <Element>LastName</Element> <Element>FirstInitial</Element>
      <Element>SeatRow</Element> <Element>SeatLetter</Element>
      <Element>Location</Element> <Element>Class</Element>
    </Group>
  </ElementDef>

  <ElementDef Type="SeatRow" Model="Data" Datatype="il" Min="1" Max="72" />
  <ElementDef Type="SeatLetter" Model="Data" Datatype="char" Min="A" Max="K"/>
  <ElementDef Type="Class" Model="Data" Datatype="char" Default="1"/>
</DCD>

```

Цей DCD-запис визначає форму із шести полів (реєстраційний запис користувача), в якому перші 3 поля згруповані в єдину послідовність, а три є незалежними елементами, які можуть бути розташовані будь-де.

Використовуючи для формування форми JSP, або JavaScript AJAX [8, 9], можна варіювати елементи керування на кінцевій сторінці навіть в процесі навчання, коригуючи контент відповідно до особливостей навчання студента, а відтак досягаючи значно вищого рівня адаптивності.

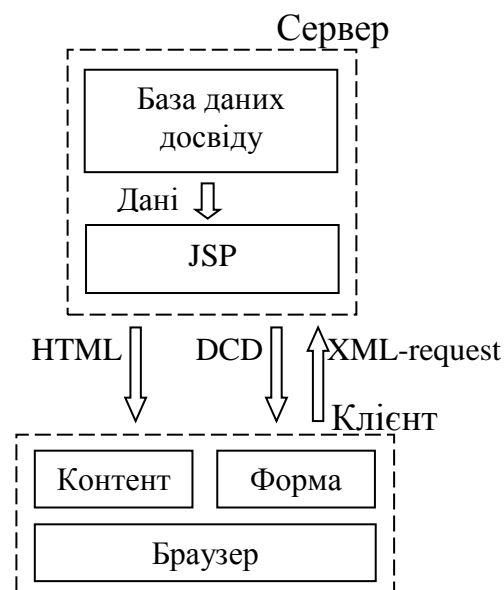


Рисунок 3 - Остаточна схема адаптивної СДН

Найефективнішого результату по створенню адаптивної системи можна досягти, лише використовуючи всі три вище перелічені технології в комплексі, коли кожна з них відповідає окремій ділянці СДН (як показано на рисунках). Таким чином приходимо до ще одної риси адаптивної СДН – модульності, що дозволяє вносити зміни в систему зі значно меншими витратами праці, ніж у інших системах дистанційного навчання, де код тісно пов’язаний з безпосереднім контентом.

Висновки

1. Проведено аналіз технологічних можливостей персоналізації курсу дистанційного навчання для підвищення показників якості навчання за допомогою засобів динамічного формування контенту. В результаті аналізу було виявлено, що

існуючі СДН не задовольняють необхідного рівня якості навчання, і необхідно внесення значних змін динамічного характеру в архітектуру подібних СДН для підвищення рівня персоналізації системи і пристосування її до різних факторів реального процесу навчання.

2. Вперше проведено синтез математичного апарату СДН з врахуванням особливостей студента (ПОД), які впливають на динаміку навчання та сприймання матеріалу. Математичний апарат описує компоненти адаптивності та аспекти її використання в базі досвіду, а також методики їх технологічного забезпечення на основі вже існуючих технологій для полегшення подальшої практичної реалізації.
3. Розроблено архітектуру для забезпечення персоналізації курсу СДН.

Література

1. О.І.Гороховський, Т.І.Трояновська - Аспекти створення, моделювання та практики автоматизованих систем дистанційного навчання – тези доповіді. «Автоматика 2006» XII МК, 25-28 вересня 2006р. В.:ВНТУ, 2006. с. – 453
2. О.І. Гороховський. к.т.н., доц, А.В. Снігур, Г.І. Гордійчук, к.ф.н, ст.вик., Т.І.Трояновська, асп. - Комп'ютерна підсистема аналізу предметно-орієнтованої домінанти студента дистанційної форми навчання. - Кременчук: КДПУ, 2007. Вип..6/2007(47) ч.1. С. 21-25
3. А.И.Гороховский, Т.И.Трояновская, В.П.Каюмов - Фреймовая основа дистанционного обучения «Гражданская авиация на современном этапе развития науки, техники и общества», МНТК 18-19 мая 2006, г. М.: МГТУГА, 2006. – с.272-276
4. О.І.Гороховський, Т.І.Трояновська, А.В.Снігур – Динамічна складова предметно-орієнтованої домінанти студента дистанційної форми навчання. Одеса
5. О.І.Гороховський, Т.І.Трояновська, Д.В.Кисюк – Автоматизація роботи викладача дистанційної форми навчання за допомогою непрямих оцінок. «Наукові дослідження – теорія та експеримент 2007», 14-16 травня, 2007. Полтава, «Інтерграфіка» 2007. с. – 127-131.
6. M. DeGroot, Optimal Statistical Decisions, McGraw-Hill, (1970).
7. T. Bray, C. Frankston, A. Malhotra Document Content Description for XML – World Wide Web Consortium, Submission 31.07.1998
8. D. Fields, M. Kolb, S. Bayern Web Development with JSP – Manning, 2002
9. J. Hunter, W. Crawford Java Servlet Programming – O'Reilly, 1998

Гороховський Олександр Іванович, к.т.н., доцент, декан ФПК СКТ, кафедра ОТ, Вінницький національний технічний університет, Хмельницьке шосе, 95, Вінниця, 21021, Україна, тел.: (0432) 59-84-05, e-mail: <mailto:goroh@vstu.vinnica.ua>

Грояновська Тетяна Іванівна, аспірант, Вінницький національний технічний університет, Хмельницьке шосе, 95, Вінниця, 21021, Україна, e-mail: trtet@mail.ru