

INTRANET АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ

© Берко А.Ю., Висоцька В.А., 2001

The problems of e-teaching systems based on the Intranet technologies development are considered in this paper.

В статті розглядаються проблеми створення систем електронного навчання на основі застосування Intranet технологій.

Сьогодні активне запровадження Internet-технологій поширилось і на галузь навчання та освіти. Термін “електронне навчання” (E- teaching) стає так само звичним, як електронна пошта, електронна комерція, електронні платежі. Застосування таких засобів дозволяє не обмежувати навчальний процес часовими та просторовими рамками навчального закладу і надавати студентові чи учневі доступ до якнайширших інформаційних ресурсів засобами мережі Internet. Незважаючи на те, що Internet-системи орієнтовані в основному на дистанційне та екстернатне навчання, їх застосування не виключає традиційних методів. В поєднанні із очними формами, такий підхід дозволяє значно розширити та урізноманітнити як форми так і зміст матеріалу, що вивчається.

На протязі майже ста років психологи значну частину своїх наукових зусиль витрачали на аналіз процесу навчання. При цьому досліджувалися, переважно, чинники, що впливають на швидкість засвоєння і втрату отриманих знань. У результаті цих зусиль були визначені основні принципи, що можуть бути використані для побудови схем навчання. Ці ж принципи мають безпосереднє відношення до розробки систем електронного навчання. Зокрема, можна виділити такі положення.

Навчальний процес йде швидше і знання засвоюються краще, якщо учень виявляє активний інтерес до досліджуваного предмету.

Навчання є більш ефективним, якщо форми придбання знань і навичок такі, що без додаткових зусиль можуть бути перенесені в умови реального життя, для яких вони призначені, що означає, що учню важніше навчитися знаходити правильні відповіді на питання, ніж просто довідуватися їх.

Навчання йде швидше, якщо учень дізнається про результат кожної своєї відповіді негайно. Якщо відповідь правильна, то учень повинний одночасно одержати підтвердження цього, якщо неправильний – він настільки ж швидко повинний дізнатися про це. Навіть незначна затримка різко гальмує навчання. В даний час учні змушені часто довго чекати результатів своєї відповіді. Знання результатів своєї роботи стимулює виконання чергового завдання.

Навчання йде швидше, якщо програма по предмету побудована за принципом послідовного ускладнення матеріалу, для цього заняття варто починати з найпростіших завдань, для виконання яких учень уже володіє необхідними навичками і знаннями, постійно рівень складності матеріалу підвищується, і це продовжується доти, поки не буде досягнутий бажаний ступінь досвідченості й уміння. Складності, що учень має переборювати, повинні виникати перед ним послідовно одна за іншою, а успішне їхнє подолання розвиває високий рівень активності.

Оскільки навчання саме по собі індивідуальне, процес навчання варто організувати так, щоб кожний учень міг проходити програму відповідно до своїх індивідуальних особливостей. З певних причин одні засвоюють матеріал швидше інших, тому навчання тих і інших в одній групі важке.

Програмні засоби, призначені для передачі тому, кого навчають, знань і/або умінь, почали розроблятися в кінці 60-х на початку 70-х років і одержали назву «Автоматизовані навчальні системи» (АНС). Основною проблемою, яка розв'язувалась за допомогою таких систем була задача формалізованого подання навчального матеріалу та контроль його засвоєння. В подальшому розвиток навчальних систем відбувався в напрямку надання їм властивості адаптації до цілей і умов навчання, підвищення рівня інтелектуальності, оптимізації навчального процесу.

За останні десятиліття спостерігається істотне збільшення обсягів і складності навчальних матеріалів, досліджуваних у середній і вищій школах та інститутах. При цьому в багатьох навчальних закладах спостерігається нестача висококваліфікованих викладацьких кадрів. Великі ускладнення часто виникають при оперативному підготуванні, виготовленні і поширенні навчальних посібників та матеріалів різноманітного спрямування. Зазначені чинники негативно позначаються на якості підготування тих, яких навчають. У зв'язку з цим велика увага приділяється застосуванню прогресивних методик навчання, у тому числі використання комп'ютерних, зокрема, Internet та Intranet технологій.

Часто під системами електронного навчання розуміються будь які Internet – засоби, що застосовуються в навчальному процесі. На думку авторів, такими можуть називатись системи, які відповідають наступним вимогам:

- орієнтовані виключно на реалізацію освітніх та пізнавальних функцій.
- містять необхідний обсяг спеціальним чином підбраного та структурованого інформаційного матеріалу, необхідного для вивчення вибраних тем та курсів.
- побудовані на основі науково обґрунтованої методики викладання.
- мають відповідним чином організований інтерфейс, що відповідає рівневі та потребам навчання.
- застосовують методи і технології штучного інтелекту в представленні, подачі та оцінці знань.

Системи Internet – навчання можна класифікувати за рівнем складності та набором освітніх функцій таким чином.

1) Тематичні оглядові Web-сайти – галереї, музеї, виставки, екскурсії, подорожі. Засоби такого типу характеризуються великою кількістю ілюстративного матеріалу, короткими коментарями і поясненнями. Розраховані на школярів, та користувачів, які мають ознайомчу мету.

2) Проблемні оглядові сайти – журнали, довідники, енциклопедії, телеконференції. На таких сайтах розміщуються матеріали підбрані за однією тематикою, орієнтовані на фахівців певної галузі та студентів.

3) Електронні книги – Web-сторінки де в спеціальному форматі подається зміст цілої книги сумісно з ілюстраціями, засобами навігації та необхідними поясненнями.

4) Електронні бібліотеки – спеціалізовані Web-сервери на яких зосереджуються як власні книжкові ресурси так і доступ до інших аналогічних систем.

5) Сайти для online – тестування дозволяють в інтерактивному режимі здати певний набір тестів з певного фаху і отримати відповідний сертифікат. Такі системи дозволяють користувачам набути підтвердження свого фахового рівня незалежно від місцезнаходження і безпосереднього контакту з екзаменатором.

6) Internet – курси забезпечують можливість не лише фахового тестування але й попереднє проходження відповідної підготовки до їх здачі.

7) Internet – школи – це комплексні системи, які поєднують засоби вивчення комплексу дисциплін із можливостями планування, контролю та керування навчальними процесами.

8) Internet – університети – найбільш складні та об'ємні системи електронного навчання користувачі яких мають можливість реалізувати в інтерактивному режимі всі

стадії здобування вищої освіти. Такі системи, як правило, не обмежуються рамками одного сервера, а й забезпечують якнайширший доступ до інформаційних ресурсів відповідного змісту.

На даний час вироблено декілька концептуальних підходів до побудови навчальних систем і засобів їх створення. За способом подання навчального матеріалу їх можна розділити на три основні види - у вигляді простого, мультимедійного або гіпертекстового документів.

Системи на основі лінійного тексту. Представлення матеріалу у вигляді звичайних документів, тобто лінійного тексту, припускає наявність деякого текстового матеріалу, розбитого на теми і сторінки. Ознайомлення того, хто навчається, із даним текстом йде в задалегідь визначеній послідовності, яку він не може змінити. У кращому випадку подібна система пропонує повернутися на крок назад або почати навчання із самого початку. Системи з подібною організацією даних звичайно не припускають яких-небудь тестових програм, а якщо такі є, то основною їх можливістю є повернення до попередньої пройдені теми у випадку недостатнього її засвоєння або просте оцінювання прочитаного матеріалу (саме прочитаного, а не вивченого). Тому системи подібного типу мало підходять для реалізації серйозних задач навчання в рамках проєктів електронних кафедр, шкіл чи університетів.

Мультимедійні навчальні системи. Мультимедійні навчальні системи дозволяють гармонійно об'єднати лекцію з демонстрацією навчального матеріалу, практику у вигляді комп'ютерного імітатора, що тестує систему і всі додаткові матеріали в єдиному інтерактивному комп'ютерному підручнику. Мультимедійний підручник не просто розвантажує викладача від щоденних рутинних функцій, але значно підвищує інтерес тих, яких навчають, до предмета, прискорює навчання і забезпечує краще засвоєння знань. Але мультимедійні системи навчання потребують відповідної апаратної та програмної підтримки, а також значних комп'ютерних ресурсів, що дещо обмежує область їхнього застосування. В мультимедійних системах основна увага звертається на спосіб подання матеріалу а не його зміст та організацію навчального процесу.

Системи на основі гіпертексту. Третім різновидом навчальних систем є гіпертекстові системи навчання. Гіпертекст як підхід до керування інформацією відрізняється від інших підходів (наприклад, СУБД) тим, що основний вид діяльності користувача при роботі з ним складається не стільки в пошуку потрібної інформації, скільки в ознайомленні з визначеним предметом за допомогою перегляду ряду інформаційних фрагментів, пов'язаних між собою за змістом. Ознайомлення здійснюється у визначеній послідовності, обумовленої цілями користувача. Можливість варіювання послідовності ознайомлення з змістом гіпертекста, на відміну від лінійного тексту, здійснюється за рахунок розбивки інформації на фрагменти (теми) і встановлення між ними зв'язків, як правило, що дозволяють користувачу перейти від досліджуваної в сучасний момент теми до однієї з декількох пов'язаних із нею тем. Очевидно, що більшою гнучкістю в сенсі задоволення різноманітних цілей користувачів володіє гіпертекст із великою кількістю зв'язків між темами.

В залежності від змісту та мети навчання в навчальному процесі може застосовуватись як один так і поєднання різних засобів електронного навчання.

У багатьох випадках розробникові комп'ютерної навчальної системи потрібно наочно уявити її структуру не тільки в загальному виді, із точністю в кращому випадку до цілої теми, як це дозволяє зробити більшість систем, але і більш конкретно, із деталізацією до більш дрібних структур, таких як визначення, теореми, алгоритми й ін. Це дозволить побачити можливі недоробки, неповноту матеріалу, відсутність яких-небудь проміжних елементів, необхідних для логічного зв'язку понять.

По такій структурі відразу можна буде побачити базові поняття, що є основними для даної системи, знання котрих необхідно перед початком процесу навчання. По такій

структурі можна легко визначити слушність послідовності подачі матеріалу для того, якого навчають, перевірити коректність уведених визначень. Наявність подібної структури може послужити відправною точкою для побудови інтелектуальної системи навчання, що дозволяє в залежності від рівня знань користувача вказувати оптимальний шлях навчання і контролювати засвоєні знання, виробити рекомендації по зміні плану навчального процесу. Все це в цілому дозволить удосконалити цикл навчання і зменшити часові витрати, необхідні на вивчення.

Рішення багатьох із цих проблем можливо тільки з використанням навчальних програм. Лише деякі з тих, хто працює в області створення таких програм, мають намір створити засіб, призначений для заміни викладача в аудиторії. Найбільше, на що можна реально розраховувати, - це сподіватися, що ці системи полегшать працю викладача, звільнивши його від функцій, які викладача і так майже не може виконати, як-от протягом усього курсу предмета, на кожному етапі негайно після засвоєння матеріалу контролювати результат. Тоді у викладача буде більше можливостей для виконання задач, що під силу тільки людині-викладачу й у виконанні котрих ніяка машина не може його замінити. Для реалізації більшості викладених вище принципів навчання в комп'ютерній навчальній системі просто необхідна чітка структуризація навчального матеріалу. Більшість же наявних на сьогоднішній день систем розробки не забезпечують можливості докладної структуризації навчального матеріалу.

На думку авторів розв'язання такого комплексу проблем лежить в галузі інтелектуальних інформаційних систем з використанням засобів телекомунікації та Internet засобів розробки користувацьких інтерфейсів.

Стан розвитку інформаційних технологій та засобів телекомунікації в Україні на сьогодні не дозволяє говорити про широке розповсюдження електронного навчання. Однак проблема потребує вирішення, і одним з варіантів може бути розробка локальних навчальних систем архітектури Intranet. Впровадження Intranet - систем електронного навчання є багатоступеневим процесом, який передбачає проходження певної послідовності етапів - від створення електронних підручників, навчальних курсів до електронних кафедр, електронних коледжів і т.д. Найбільш ефективним, на погляд автора, є створення комплексних навчальних систем. Такі засоби забезпечують паралельне вивчення групи пов'язаних дисциплін в обсязі теоретичного і практичного курсу, а також поточний та підсумковий контроль і оцінювання засвоєння вивченого матеріалу.

Основні функції Intranet-систем електронного навчання можуть бути визначені:

1) *Відбіркове тестування з метою визначення рівня підготовки та придатності до навчання.* На основі даних, отриманих на цьому етапі приймаються рішення про можливість навчання, вибір варіанту навчальної програми (мінімальна базова, стандартна, розширена, тощо), інтенсивності навчального процесу, змісту та способів подання навчального матеріалу і практичних завдань і реєстрація користувача.

2) *Теоретичне навчання.* В рамках цієї функції користувач має змогу в інтерактивному режимі отримувати певні порції навчального матеріалу (електронні лекції) з необхідними коментарями, поясненнями і прикладами, завдання на самостійну підготовку та контролювати процес засвоєння за допомогою відповідей на питання для самоперевірки.

3) *Практичний курс.* Полягає у виконанні циклу практичних завдань з безпосереднім контролем правильності і необхідними поясненнями. В ході практичного заняття можливе, по мірі необхідності, застосування елементів теоретичного курсу для розв'язання тих чи інших практичних проблем.

4) *Поточний контроль*. На основі відповідей на контрольні запитання та виконання практичних завдань за обраною системою відбувається постійне відслідковування та фіксація рівня засвоєння навчального матеріалу. Інтегровані результати можуть бути застосовані викладачем для внесення коректив чи прийняття рішень по зміні організації навчального курсу, а також як складова підсумкового випускного результату.

5) *Підсумкове тестування по курсу*. Це заключний етап, який полягає у відповіді користувача на систему тестів і завдань, які дозволяють оцінити результат навчального процесу і прийняти рішення про зарахування чи незарахування результатів навчання за вибраною системою оцінювання.

6) *Випускне випробування*. Ця функція передбачає комплексне тестування за результатами вивчення групи курсів виконання комплексних кваліфікаційних завдань, а також врахування результатів підсумкових тестувань.

На основі цих функцій можна визначити загальну архітектуру інтелектуальної системи Intranet – навчання, а реалізація перерахованих вище можливостей слугує основою для розробки системи формування й опрацювання структури електронного підручника як складової інтелектуальної системи електронного навчання, реалізованої за Intranet - технологією.

При розробці багатьох експертних систем свідомо робились зусилля розділити задачу на дві частини - машину виведення і базу знань. Ідея полягала у тому, що машина логічного виведення є універсальною думуючою машиною, а БЗ - це те, з чим вона буде працювати.

Формулювання вимог до підручника інтелектуальної системи електронного навчання. Підводячи результат усьому вищесказаному, можна відзначити відсутність або недостатню розвиненість у всіх розглянутих системах деяких засобів, дуже важливих і корисних для розроблювачів і користувачів автоматизованої навчальної системи. Можна сформулювати список можливостей, що повинні бути в автоматизованій навчальній системі.

Для користувачів:

- Організація навчання різного рівня – від початкового знайомства до докладного засвоєння матеріалу.
- Можливість надання матеріалу виходячи з цілі навчання.
- Компонування матеріалу за результатами тестових перевірок.

Для розроблювачів:

- Перевірка коректності уведених визначень.
- Формування списку невизначає понять.
- Побудова для виділених понять (і для всього підручника) графа зв'язку з визначальними поняттями - ієрархічний граф понять.

Система має виконувати такі основні функції:

- Побудова структури понять електронного гіпертекстового підручника.
- Відображення отриманої структури в наочному і зручному для користувача виді: пошук елемента в структурі; можливість переходу від перегляду структури до перегляду підручника;
- Опрацювання отриманої структури:
- перевірка коректності визначень у структурі;
- виділення списку вихідних (невизначає) понять;
- виділення підструктури по заданій множині понять.

Розробка структури системи. Система призначена для опрацювання гіпертекстового електронного підручника, написаного на мові HTML. Мова HTML - це мова опису Web-сторінок, тому для перегляду певного підручника використовується Web-браузер.

Виходячи з вимог до автоматизованих навчальних систем, очевидно, що для реалізації таких функцій, як компонування матеріалу за результатами тестових перевірок або виходячи з заданої цілі навчання, необхідна дуже тісна інтеграція системи з підручником. Звідси ми маємо, що, так як підручник проглядається через браузер, то для тісного з ним зв'язку необхідно, щоб розроблювальна система також працювала під керуванням браузера.

Справа в тому, що основне призначення браузерів – це перегляд Web-сторінок у Internet. Тому з розуміння безпеки на програми, що можуть виконуватися браузерами, накладені серйозні обмеження. Вони не мають доступу до ресурсів комп'ютера користувача, не можуть читати або писати файли, запускати якісь програми з його комп'ютера.

Очевидно, що розроблювальна система, побудувавши структуру підручника, повинна десь її берегти. У принципі, можливий варіант збереження отриманої структури в оперативній пам'яті комп'ютера. Але в цього варіанта є серйозні недоліки.

- *По-перше*, витрачається зайва пам'ять.
- *По-друге*, структуру підручника прийдеться щораз при запуску системи формувати наново.

На стадії розробки підручника це може бути не так критично, зважаючи на те, що структура буде змінюватися досить часто. Але при використанні даної системи для аналізу рідко мінливого підручника формування структури при кожному запуску стає неприйнятним, тим більше що при великому обсязі підручника цей процес може займати декілька хвилин. По-третє, структура існує тільки під час роботи системи, що не дає можливості працювати зі структурою окремо від підручника, зберігати історію зміни структури підручника.

Виходячи з усього вищесказаного, виникнула ідея розділити систему на два окремі функціонально-закінчених модулі. Перший модуль буде робити опрацювання електронного підручника, формування його структури і запис даної структури у файл у визначеному форматі. Даний модуль є самостійним додатком, що не залежить від браузера і тому на нього не накладаються усі вищеописані обмеження, пов'язані з політикою безпеки браузерів. Назвемо його модулем формування структури.

Другий модуль візьме на себе усі функції, пов'язані з відображенням і опрацюванням отриманої структури. Цей модуль є програмою, що виконується під керуванням браузера, як-от Java-апплетом. Так як Java-апплети мають можливість читати файли з тих серверів, відкіля вони запущені, то проблем із завантаженням файла структури, сформованого першим модулем, не буде. Назвемо даний модуль модулем опрацювання і відображення структури.

Формування структури підручника. Електронний підручник подає із себе сукупність параграфів визначених типів. Ці типи - визначення, теореми, пояснення, приклади, доведення, алгоритми й ін.

Правила побудови підручника. Для забезпечення можливості формування структури підручника останній повинний бути побудований по визначених правилах. Були розроблені такі правила побудови вищезгаданих параграфів.

Для оцінки початку і кінця параграфів було вирішено використовувати такі конструкції. Початок параграфа відзначається в такий спосіб:

```
<A NAME="мітка_початку_параграфа"></A>
```

де мітка_початку_параграфа подає із себе рядок, складений із ключового слова, що ідентифікують факт початку і тип параграфа, і рядки, що представляє із себе стислу назву параграфа. Наприклад, ``. У даному прикладі ключовим словом є startdef, що означає початок параграфа типу визначення, у якому визначається поняття "Автомат Мура". Всі прогаліни в назві параграфа повинні бути замінені на символи підкреслення. Це пов'язано з тим, що деякі засоби генерації і перегляду HTML сторінок не припускають прогаліни у параметрі NAME тегу `<A>`.

Кінець параграфа відзначається аналогічною конструкцією, із тієї тільки різницею, що ключове слово замінюється на інше, що ідентифікує кінець параграфа.

Вибір подібних конструкцій заснований на таких розуміннях. По-перше, вставка даних конструкцій ніяк не відбивається на зовнішньому вигляді HTML-документа. По-друге, дані конструкції одночасно є мітками параграфів із погляду HTML, тобто не вводять ніяких додаткових міток, ми можемо побудувати посилання на любий описаний подібним образом параграф. По-третє, використання саме таких конструкцій полегшує побудову гіпертекстового документа, тому що багато засобів розробки гіпертекстів, наприклад, Microsoft Word, дозволяють робити в тексті закладання, що перетворюються саме в подібні теги.

Таким чином, параграф, наприклад, типу визначення, має такий вигляд:

`текст визначення .`

Якщо в тексті визначення зустрічаються посилання на інші параграфи, вони повинні бути оформлені в такому вигляді:

`.`

Подібне оформлення параграфів дозволяє побудувати структуру понять з урахуванням усіх наявних зв'язків між ними.

Алгоритм формування структури. Структура електронного підручника формується в такий спосіб. Весь процес розбитий на два етапи.

Перший етап - перегляд підручника й упорядкування списку всіх понять, побудованих по описаній вище схемі. При цьому для кожного параграфа складається список усіх посилань, виявлених усередині нього, у вигляді імені сторінки плюс безпосередньо імені посилання.

Другий етап - аналіз даного списку понять, із метою побудови зв'язків між ними. Аналізуються внутрипараграфні посилання і на їхній основі будуються зв'язки між поняттями.

Формат вихідного файлу модуля формування структури. Файл, що містить структуру електронного підручника, подає із себе звичайний текстовий файл. Перший рядок файлу завжди містить повний шлях до папки, у якому знаходиться оброблений електронний підручник.

Такі рядки являють собою описи вершин графа понять і мають такий формат:

ідентифікатор, ім'я, адреса, тип,

де

ідентифікатор - унікальний номер вершини,

ім'я - назва вершини, що збігається з обумовленим поняттям,

адреса - відносний шлях і ім'я файлу, що містить дане поняття,

тип - числове значення, що визначає тип вершини (0 - визначення, 1- теорема). Всі ці параметри перераховані через кому.

Закінчується частина описів вершин графа рядком, що містить слово «Edges». Після цього рядка до кінця файлу йдуть списки суміжностей кожної вершини. Один рядок містить один список суміжностей, що починається з номера вершини, до котрого цей список відноситься, і далі через кому перераховані номери суміжних із нею вершин.

Алгоритм перевірки коректності визначень. Визначення рахується коректним, якщо воно не визначається через саме себе, або через поняття, що визначаються через нього. Іншими словами, якщо застосувати це до графа понять, то можна сказати, що в графі повинні відсутні контури. Для знаходження контурів у графі понять використовується метод пошуку в глибину зі зберіганням шляху.

Суть алгоритму пошуку в глибину полягає в такому. Нехай у нас є граф $G = \langle V, E \rangle$.

1) Починаємо пошук із довільної вершини v_0 .

2) Вибираємо довільну вершину u , суміжну з $v0$ і повторюємо пошук від вершини u .

3) У загальному випадку припустимо, що ми знаходимося в деякій вершині v . Якщо існує ще не переглянута вершина u , суміжна для v , то повторюємо пошук, починаючи з вершини u . Якщо непереглянутих вершин, суміжних із v , немає, то вважаємо вершину v використаною і повертаємося у вершину, із якої потрапили в v . Якщо при поверненні одержуємо $v = v0$, то вважаємо алгоритм завершеним і усіма вершинами є використаними.

Ідея пошуку контурів за допомогою методу пошуку в глибину полягає в наступному. При пошуку зберігається шлях, що складається з переглянутих вершин, якщо ми зустрічаємо вже переглянуту вершину, значить виявлений контур. При цьому по збереженому у пам'яті шляху ми можемо відновити вершини даного контуру.

Алгоритм виділення списку вихідних понять. Вихідними поняттями є ті поняття, при визначенні яких не використовуються інші поняття, що містяться в підручнику. У графі поняття такими є початкові вершини графа, тобто не мають вхідних дуг.

Алгоритм виявлення подібних понять тривіальний. Проглядаються усі вершини графа, і ті вершини, що не мають вхідних дуг, включаються в даний список.

Алгоритми виділення підструктур. Підструктури по заданій множині понять можуть виділятися двома засобами. Перший засіб - виділення підграфу від вихідних понять, тобто від початкових вершин, до обраних вершин. Другий - виділення підграфа від обраних вершин до кінцевих вершин графа понять. Для обох варіантів використовується в принципі однаковий підхід.

У першому варіанті для кожної початкової вершини графа відшукується шлях у кожному з обраних вершин. Отримані в результаті цього шляху об'єднуються й утворюється шукана підструктура.

В другому варіанті відмінність складається в тому, що для кожної обраної вершини графа відшукуються шлях до кожної кінцевої вершини. Потім також відбувається об'єднання отриманих шляхів.

Знаходження шляху між двома вершинами провадиться з використанням методу пошуку в глибину, вже описаного вище. Пошук шляху за його допомогою здійснюється в такий спосіб. Щоб знайти шлях між вершинами v і u методом пошуку в глибину, потрібно почати пошук у вершині v і продовжувати його до відвідування вершини u . У момент відвідування вершини u послідовність переглянутих (але невикористаних) вершин буде визначати шлях із v у u .

Висновок

Так, для реалізації електронного підручника необхідним є в першу чергу :

- формування структури понять електронного гіпертекстового підручника;
- візуалізація отриманої структури в наочному і зручному для користувача виді;
- обробка отриманої структури, тобто:
- перевірка коректності визначень у структурі;
- виділення списку вихідних (невизначених) понять;
- виділення підструктури по заданій множині понять, тобто за метою навчання чи за результатами тестування.

1. Андриенко Г.Л., Андриенко Н.В. *Интеллектуальная гипертекстовая система для исследования проблем и обучения.*// Конференция по искусственному интеллекту КИИ-94.Сб-к трудов. Тверь, 1994.- С.58-62. 2. Норенков Ю.И., Михайловский О.В. *Адаптивная автоматизированная обучающая система.*//Конференция по искусственному интеллекту КИИ -94.Сб-к трудов. Тверь, 1994.- С.72-76. 3. Алгоритмы и программы решения задач на графах и сетях / Нечепуренко М.И., Попков В.К., Майнагашев С.М. и др. - Новосибирск: Наука. Сиб. Отд-ние, 1990. - 515с