

Електронні задачники. Вимоги користувача та технології реалізації (на прикладі ПМК Терм)

У найпростішому варіанті електронний задачник – це сховище навчальних задач, які користувач може розв'язувати „на папері”. Але, оскільки процес розв'язування навчальних задач з математики, фізики, інших природничо-наукових дисциплін у більшості випадків може бути формалізований, електронні задачники можуть мати достатньо багату функціональність. Цим обумовлена наявність серед програмних засобів навчального призначення достатньо великої кількості різноманітних електронних задачників. Електронні задачники, як правило, входять до складу програмних систем навчального призначення. Одні з них створюються як спеціалізовані програмні засоби, інші розробляються в спеціалізованих середовищах - редакторах електронних підручників.

Очевидно, що найбільш перспективними є педагогічні технології, в яких електронний задачник є спеціалізованим програмним модулем, який містить навчальні задачі, призначені для розв'язування у програмних середовищах, спеціально призначених для розв'язування цих задач. Таким чином, електронний задачник (Задачник) – це об'єкт, який є сховищем задач, доступних для розв'язування в програмному модулі «Середовище розв'язування задач» (СРЗ)[1].

Зміст (навчальний матеріал) Задачника має бути розділеним на навчальні модулі або класи навчання, причому користувач може обирати навчальний модуль або клас навчання.

Задачник має задовольняти загальноприйнятим вимогам щодо інтерфейсу електронних підручників та створеним із залученням засобів мультимедіа, тобто вимогам зручності і наочності навігації в електронних навчальних ресурсах, простоти та оперативності переходів до необхідних розділів, об'єктів та інших електронних засобів навчання[10,11].

Перелік функцій задачника найбільш природним чином можна визначити, використовуючи метод опорних точок зору. Об'єкт Задачник підтримує наступні функції[3]:

1. Точка зору учня:

1. зберігання навчальних задач, доступних для розв'язування в СРЗ;
2. роздрукування або окремих задач, або серій задач;
3. вибір навчальних задач та їх відправлення для розв'язування в СРЗ;
4. мічення задач, які вже розв'язані користувачем, спеціальною позначкою;

5. доступ у режимі читання до ходу розв'язування задачі. (Хід розв'язування навчальної задачі зберігається в спеціальному програмному модулі „Робочий зошит“.)

2. Точка зору вчителя:

1. можливість виконання всіх функцій учня;
2. зберігання (у кількох варіантах) тематичних комплексних поточних та контрольних робіт для організації самостійних та контрольних робіт у класі;
3. можливість створення кількох варіантів тематичних комплексних поточних та контрольних робіт з навчальних задач Задачника, яка включає можливість зміни числових даних умов задачі для створення кількох варіантів умови навчальної задачі;

3. Точка зору укладача Задачника:

1. можливість виконання всіх функцій учня та вчителя;
2. редагування змісту Задачника, яке полягає як у вставлянні нових навчальних задач та вилученні наявних задач, так і у редагуванні змісту окремих задач (зокрема, зміни числових даних умови задачі).
3. підтримка відображення статичних й анімованих графічних зображень, а також відеофрагментів, що підвищують наочність подання матеріалу.

Крім вимог з боку вказаних категорій користувачів, перелічимо вимоги до Задачника з боку інших програмних модулів:

4. Середовище розв'язування, Робочий зошит[2]:

1. Список команд (перетворень), які підтримуються СРЗ, має бути функціонально повним відносно Задачника. Це означає, що кожна з навчальних задач Задачника може бути розв'язана засобами СРЗ (Хід розв'язування навчальної задачі є послідовністю команд (перетворень) зі списку команд СРЗ).
2. Умови навчальних задач мають зберігатися у форматі, сумісному з форматом подання умови в СРЗ і Зошиті.

Нарешті, перелічимо основні технологічні та системні вимоги:

5. Точка зору розробників програмного засобу, до складу якого входить Задачник:

1. Технологія наповнення Задачника навчальними задачами має підтримувати оперативну зміну його змісту без перекомпіляції модуля (і тим більше всієї системи). Із цією метою повинен бути розроблений спеціальний редактор Задачника, користувачем якого є методист-укладач Задачника.
2. Поточна версія Задачника для вдосконалювання, так і зареєстрованим користувачам системи.

Технологія реалізації Задачника повинна підтримувати:

1. користування Задачником у мережі (Інтранет та Інтернет).

2. нормальна швидкодія на персональних комп'ютерах середньої продуктивності з типовим набором апаратно-програмних засобів.
3. переносимість електронних навчальних ресурсів на різні обчислювальні платформи.
4. ліцензійна чистота використовуваних при розробці інструментальних засобів.
5. коректне функціонування під управління різних операційних систем (Windows 98/Me, Windows 2000/XP, Linux, MacOS у стандартних комплектаціях).

Щодо вимог до програмного забезпечення, то вони диференціюються в залежності від того, на яку операційну систему орієнтовані.

На сьогоднішній день склалася така ситуація, що в більшості освітніх закладів поширені операційні системи компанії Microsoft – сімейство ОС Windows®.

Але не слід орієнтуватися лише на ці операційні системи, оскільки поступово і неупинно зростає кількість навчальних закладів, які все частіше в процесі навчання користуються комп'ютерами зі встановленою операційною системою Linux, та вводять навчально-методичні курси з вивчення роботи в цій операційній системі.

Основним об'єктом Задачника є навчальна задача (Задача). Тому правильне визначення цього об'єкта атрибутами, властивостями та методами та правильний вибір технологій реалізації мають вирішальне значення.

Одна з найбільш важливих проблем, які підлягають вирішенню при проектуванні системи підтримки навчальної математичної діяльності – проблема генерації методично правильного ходу розв'язування завдання. Іншими словами, система повинна «знати», як вирішуються завдання й «уміти» підказувати користувачеві, як вирішити те або інше завдання.

Зазначена проблема є однією із центральних проблем у дослідженнях зі штучного інтелекту. Як правило, у цих дослідженнях вивчаються методи моделювання поведінки експерта (що знає багато методів розв'язування широкого класу завдань), при вирішенні даного завдання. Наш підхід опирається на іншу парадигму: завдання повинно знати й підказувати користувачеві метод свого розв'язування. Таким чином метод вирішення даного завдання повинен бути сформульований методистом у вигляді послідовності зрозумілих користувачеві дій, що зберігається в задачнику разом з умовою завдання. Система «методичних дій» повинна бути реалізована в спеціальному модулі системи (базі вмінь методиста).

Для того, щоб вирішити прикладне завдання, необхідно спочатку скласти відповідну математичну модель, а потім проаналізувати її. Тому система спочатку повинна перевіряти правильність складеної моделі, а потім підтримати процес її

аналізу. Оскільки текст умови завдання не формалізований, процедура перевірки правильності складання моделі повинна мати справу із правильною моделлю завдання. Ця модель також складається методистом і зберігається в задачнику разом з її умовою[3].

Для підтримання цієї концепції, до кожної текстової задачі додаються ряд атрибутів, за якими СРЗ може “взнати”, як розв’язувати цю задачу:

- ✓ тип задачі;
- ✓ схована формула;
- ✓ Відповідь.

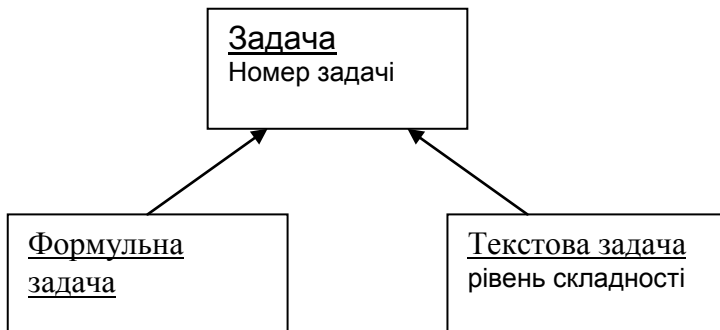
Велику роль при проектуванні та розробці задачника відіграла необхідність “взаємодії” „Задачника” та „Середовища розв’язання задач”. Для розуміння цієї “взаємодії” необхідно було визначити, що обрати за мінімальну одиницю обміну даними.

Мінімальною одиницею обміну даними є задача. Фрейм задачі має наступні атрибути:

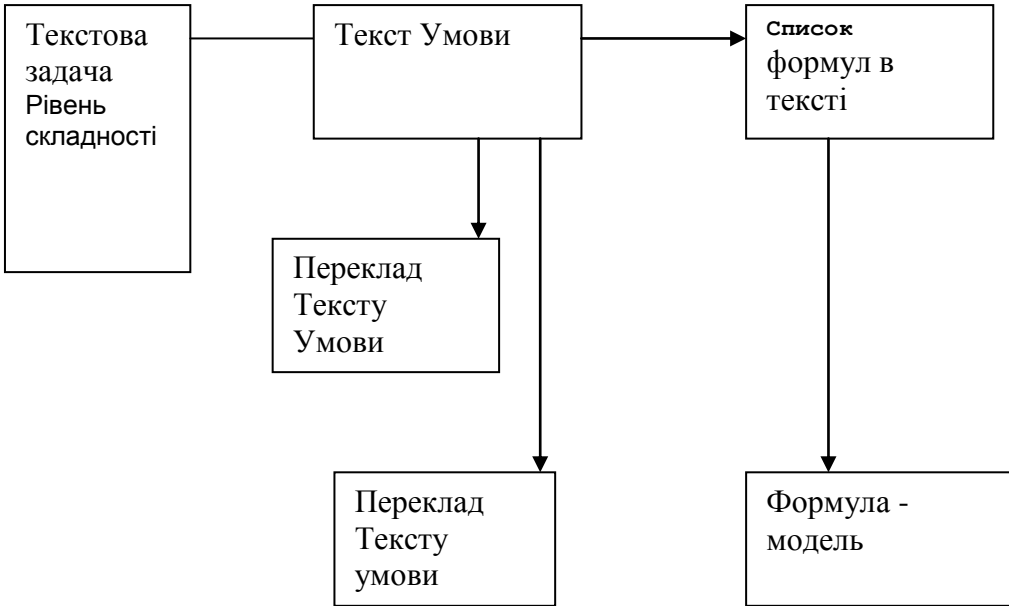
- Номер задачі;
- Тип задачі
- Вид задачі
- Статус задачі;
- Походження задачі (з задачника або створена учнем);
- Умова (текстової задачі або формульної)
 - Текст умови на 3-х мовах;
 - Початкова формула;
- Хід розв’язування задачі;
- Відповідь.

Оскільки мова йде про обмін даними між задачником та модулями, що використовуються для розв’язування та зберігання задач, то необхідно подбати про можливість простого узгодження структур даних та врахувати майбутні доповнення або зміну формату. Для цього необхідно ввести деяку загальну мову опису структури даних і створити сховище-каталог, у якому ці описи зберігалися б.

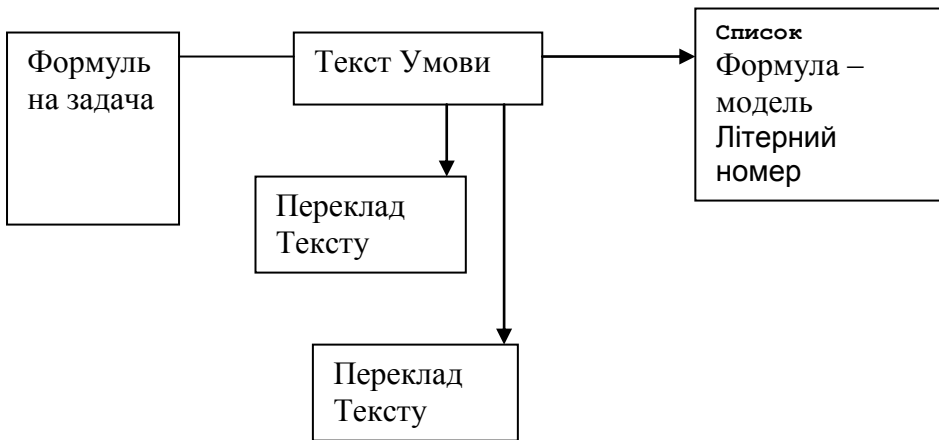
Діаграма класів **Задача**



Діаграма об'єктів Текстова задача



Діаграма об'єктів Формульна задача



Схованою формулою (Формула-модель) – називається формула, якою задається математична модель задачі, на основі якої користувач починає її розв'язувати, а також СРЗ може контролювати правильність виконуваних дій.

Тип задачі – дозволяє визначити, які перетворення йому можуть знадобитися для розв'язування цієї задачі.

Існує велика кількість можливих реалізацій електронних задачників, кожна з яких базується на виборі одного із можливих варіантів побудови того чи іншого окремого компонента, з яких складається сучасний електронний задачник з точних та природничих дисциплін.

Такими компонентами є:

- ✓ сховища даних;
- ✓ процесори опрацювання даних;
- ✓ відображувачі;
- ✓ оболонки, що надають зручності в користуванні.

Найбільш прийнятними для реалізації є такі варіанти, коли сховище задач міститься у базі даних (MS SQL, MySQL, PostgreSQL, Oracle, Access), управління яким здійснюється за допомогою відповідної СУБД, вибір якої зумовлений наявністю відповідних драйверів у комплекті дистрибутиву операційної системи та правом використовувати її у власних цілях. Але такий вибір не виправданий через відсутність необхідності зберігати та опрацьовувати великі масиви даних та затрачувати для цього значну кількість процесорного часу. Альтернативою збереження задач у базах даних можуть слугувати:

- ✓ звичайні текстові файли;
- ✓ бінарні файли;
- ✓ HTML-файли;
- ✓ XML-файли].

Сформулювавши ці вимоги, спробуємо спроектувати їх на існуючі технології подання даних. Найбільш простим та очевидним рішенням для організації загальних даних буде вибір мови розмітки та представлення – XML[5,7,8]. Цей вибір, окрім того вирішує проблему платформонезалежності. Як засоби опису структури виберемо XML-схеми (у термінах XSD). Для опису формул обрана спеціальна мова розмітки тексту MathML[12], що в вже є стандартом для представлення формул при розміщенні їх у електронному вигляді.

У якості мови програмування для написання графічної оболонки можна обрати будь-яку мову написання програм, в якій передбачено можливість створювати програми з графічним інтерфейсом. Але кожна з них має свої переваги і недоліки: за допомогою одних можна розробляти швидкі програми (C, C++)[6,7], але складність написання їх не дозволяє стверджувати, що вони задовольняють вимогам щодо економічної доцільності їх використання, інші навпаки, легкі в написанні (VB, Delphi, C#), але обмежені у функціональності або гнучкості використання. В деяких мовах програмування передбачено можливість розробляти програми, що будуть працювати під управління різних операційних систем без перекомпіляції коду програм (Java, Python, Perl), але для написання таких програм необхідно значно вища кваліфікація розробників, проектувальників програмного забезпечення, та знання архітектур різних ОС.

Але кожна з цих мов може бути використана для написання графічної оболонки для сучасних електронних задачників.

Навчальні задачі згруповано в кількох розділах. Розділи мають назви та номери. Кожен з розділів містить кілька параграфів. Параграфи також мають назви та номери. Нумерація параграфів є внутрішньою стосовно розділу. Розділи містять задачі для розв'язування під час практичних занять, самостійної домашньої роботи або задачі тематичних атестацій.

Задачник є гіпертекстом, який структурований змістом. Зміст задачника поданий у лівій частині вікна Задачника – полі Зміст. Для того, щоб відкрити потрібний розділ та параграф Задачника, треба вказати на відповідний рядок поля Зміст з заголовком потрібного розділу або параграфу. Задачник відкриється на потрібному розділі або параграфі, зміст якого відобразиться у вікні для відображення вмісту.

За способом нумерації задачі можна поділити на дві групи – алгебраїчні задачі та текстові задачі. Кожна задача має свій номер. Нумерація алгебраїчних задач подвійна. Нумерація текстових задач звичайна. Нумерація задач перших розділів є наскрізною. Номер задачі тематичних атестацій складається з номеру атестації, номеру варіанта та номера задачі у цьому варіанті.

Для того, щоб приступити до розв'язування обраної задачі в Середовищі розв'язування, треба натиснути на кнопку Розв'язувати, яка розташована біля кожної з тих задач Задачника, які розв'язуються алгебраїчними перетвореннями. Відкриється головне вікно Середовище розв'язування з умовою обраної задачі, що розташована в полі умови.

Оболонка, що відображає вміст електронного задачника та редактор задачника описані мовою програмування Visual C++. Вибір цієї мови визначався тим, що вся система була описана мовою Visual C++[6], тобто для більшої інтеграції окремих модулів була обрана саме вона. Але в іншій ситуації найкращим вибором є використання для побудови оболонки мови програмування JAVA.

Поєднання технологій XML & XSLT[14] з JAVA дозволяють створювати гнучкі, платформонезалежні додатки, що можуть бути використані без перекомпіляції як на локальних комп'ютерах так і в мережі Internet.

Особливістю даної реалізації електронного задачника є надання кожній задачі деяких інтелектуальних властивостей, а саме „підказувати” середовищу розв'язування задач як її треба розв'язувати. Тобто разом із умовою задачі зберігається і її математична модель. Це виділяє дану реалізацію електронного задачника серед подібних програм, ставлячи при цьому його на значно вищій щабель порівняно з іншими подібними.

Такий підхід дозволяє модифікувати оболонки задачників для конкретної платформи, при цьому не змінюючи дані, архітектуру, процес “взаємодії” модулів, дозволяє створювати

задачники, придатні для роботи з локального диска, через локальну мережу й через Інтернет.

Пропонована технологія має на меті об'єднати в собі позитивні риси інших підходів, а також звести до мінімуму негативні.

ЛІТЕРАТУРА

1. Львов М.С., Спиваковский А.В. Методы проектирования систем компьютерной поддержки математического образования. «Математические модели и современные информационные технологии» // Материалы международной конференции по математическому моделированию, 3-6 сентября 1998 г. – Херсон: с. 101-110.
2. Львов М.С., Спиваковский А.В. Методы проектирования систем компьютерной поддержки математического образования. «Математические модели и современные информационные технологии» // Материалы международной конференции по математическому моделированию, 3-6 сентября 1998 г. – Херсон: с. 101-110.
3. Львов М.С., Бейко Ю.В. Электронный задачник системы поддержки учебной математической деятельности // Информатизация освіти України: стан, проблеми, перспективи: Зб. наук. пр./ Херсонський державний університет. – Херсон, 2003. – 90 с.
4. Кей М. XSLT. Справочник программиста. - Пер. с англ. – СПб.: Символ-Плюс, 2002. – 1016 с., ил.
5. Янг Майкл. XML. Шаг за шагом: Практ. пособ. – Пер. с англ. – М.: Издательство ЭКОМ, 2000. – 384 с., ил.
6. Microsoft Corporation. Разработка приложений на Microsoft Visual C++ 6.0. Учебный курс /Пер. с англ. – 2-е изд. – М.: Издательско-торговый дом "Русская редакция", 2001. – 704 с., ил.
7. Как программировать на XML / Х.М. Дейтл, П.Дж. Дейтел, Т.Р. Нието, та інші /Пер. с англ. – М.: ЗАО "Издательство БИНОМ", 2001. – 944 с., ил.
8. Чернышов Л.Н. Технологии, основанные на XML // Новые информационные технологии: Тезисы докладов XI Международной студенческой школы-семинара в 2-х томах. – М.: МГИЭМ, 2003. – 641с.
9. Кот С.М. Технологические аспекты дизайна обучающих программ // Информатизация освіти України: стан, проблеми, перспективи: Зб. наук. пр./ Херсонський державний університет. – Херсон: Айлант, 2001. – 176 с.
10. Кот С.М. Дизайн обучающих программ // Информатизация освіти України: стан, проблеми, перспективи: Зб. наук. пр./ Херсонський державний університет. – Херсон, 2003. – 173 с.
11. Mathematical Markup Language (MathML) Version 2.0 (Second Edition) / W3C Recommendation 21 October 2003
<http://www.w3.org/TR/MathML2/>

12. HTML 4.01 Specification / W3C Recommendation 24 December 1999
<http://www.w3.org/TR/html4/>
13. XSL Transformations (XSLT) Version 1.0 / W3C Recommendation 16 November 1999
<http://www.w3.org/TR/xslt>
14. Extensible Markup Language (XML) 1.0 (Third Edition) / W3C Recommendation 04 February 2004
<http://www.w3.org/TR/2004/REC-xml-20040204/>