

УДК 378.147



**М.С. Сафонов**  
аспірант, Одеський  
національний  
політехнічний  
університет



**А.О. Маляренко**  
адміністратор бази даних  
(обслуговування  
ЄДЕБО), Херсонський  
політехнічний коледж  
Одеського національного  
політехнічного  
університету  
email:malyarenkoalena21@  
gmail.com

## МЕТОД УЗГОДЖЕННЯ ГЕТЕРОГЕННИХ БАЗ ДАНИХ ЗА ДОПОМОГОЮ ЕМУЛЯЦІЇ ДІЙ КОРИСТУВАЧА

*М.С. Сафонов, А.О. Маляренко.*  
*Метод согласования гетерогенных баз данных с помощью эмуляции действий пользователя на WEB-странице.*  
Проанализированы существующие методы согласования данных с WEB-страницами. Представлен метод эмуляции действий пользователя для согласования данных с сайтом.

*M.S. Safonov, A.O.Maljarenko.*  
*A method of the coordination of heterogeneous databases by means of emulation of actions of the user on WEB-page.* Existing methods of the coordination of the data with WEB-pages are analysed. The method of emulation of actions of the user for the coordination of the data with a site is presented.

**Вступ.** Інтеграція гетерогенних баз даних (БД) з різними структурами, різними форматами або різними системами керування є однією з найстаріших завдань у розвитку БД та інформаційних систем (ІС), які використовуються в територіально розподілених об'єктах [1]. Ще в 1970-х роках цим питанням займався кібернетик світового рівня Глушков В. М. у рамках розробки системи електронного документообігу державного управління "Електронний уряд" [ 2].

За останні роки задача гетерогенної інтеграції виникла в Україні у зв'язку з впровадженням Єдиної державної електронної бази з питань освіти (ЄДЕБО), яка є обов'язковою для використання всіма навчальними закладами. В той же час, впродовж десятиріч в навчальних закладах було створено власні ІС, що призвело появи збиткових процесів узгодження даних в БД закладу та БД ЄДЕБО. Виключення такої збитковості можливе лише за рахунок автоматизованого імпорту та експорту даних між цими БД.

Автоматизація і комп'ютерні технології

**Аналіз існуючих методів узгодження даних.** Припустимо, є декілька гетерогенних джерел даних, які певним чином пов'язані на логічному рівні. Існує завдання розробити програмне забезпечення, яке б забезпечувало можливість уніфікованого доступу до цих даних таким чином, нібито вони мають єдине логічне і фізичне представлення.

Існують два фундаментальні підходи до вирішення цієї проблеми. Перший підхід пов'язаний з побудовою сховищ даних, коли інтегровані дані з різних джерел трансформуються відповідно до цільової моделі даних і розміщуються в одній локальній базі даних.

Другий підхід пов'язаний з поняттям віртуальної інтеграції гетерогенних джерел даних, коли дані не матеріалізуються в локальній базі даних, а використовують проміжне програмне забезпечення, яке транслює призначені для користувача запити в підзапити до джерел і формує кінцевий результат.

Але дані підходи ускладнюються тим фактом, що не завжди існує безпосередній доступ до гетерогенного сховища даних, а лише до інтерфейсу доступу до нього. Здебільшого це стосується WEB-сайтів, де доступ до даних організовано лише через WEB-сторінку.

Виникнення XML і супутніх технологій викликало сплеск нових розробок з цієї проблеми. Встановлена у БД XML DB дозволяє засобами СУБД, без залучень зовнішнього контейнера, організувати доступ до даних у базі [3, 4].

Для взаємодії ВНЗ, у яких є власні програмні продукти по обліку абітурієнтів, з базою даних ЄДЕБО існує ряд web-сервісів, за допомогою яких розробники можуть отримувати, змінювати, і оперувати інформацією, яка зберігається у базі даних ЄДЕБО. Для роботи з web-сервісами використовується протокол SOAP, що підтримується усіма основними платформами розробки програмного забезпечення та дозволяє виконувати обмін даними у форматі XML.

Але даний метод має ряд недоліків:

- обмежена універсальність – даний метод можна використовувати лише для тих сайтів, де організований доступ до БД за допомогою XML (наприклад, доступ в Єдиний державний реєстр судових рішень можливий лише через WEB-інтерфейс);
- вивчення структури XML займає багато часу та потребує високої кваліфікованості програміста;

- в результаті великої гнучкості мови і відсутності суворих обмежень, одна і та ж структура може бути представлена великою кількістю способів, наприклад, значення може бути записане як атрибут тега або як тіло тега (що також збільшує час на узгодження сховищ даних)[5].

**Мета та завдання дослідження.** Виходячи з вище зазначеного, метою даного дослідження стало зменшення часу узгодження за рахунок розробки методу емуляції дій користувача.

Для узгодження даних з ЄДЕБО був обраний метод емуляції дій користувача на WEB-сторінці. Метод базується на наступних принципах:

- дії користувача повинні бути задалегідь визначеними;
- імпорт та експорт даних повинен відбуватися автоматизовано;
- елементи управління на WEB-сторінці повинні мати унікальні імена або властивості;
- програмне забезпечення повинно точно відтворювати поведінку користувача в стандартних ситуаціях.

Метод полягає в автоматизованому повторенні послідовностей дій, які зазвичай виконує користувач при роботі з сайтом.

Всі дії були розділені на декілька видів:

- переходи по гіперпосиланням;
- натиснення кнопки на WEB-сторінці;
- введення даних в текстове поле;
- читання даних з таблиць форми.

**Метод узгодження гетерогенних баз даних за допомогою емуляції дій користувача на web-сторінці.** Для знаходження відповідності значень на WEB-сторінці значенням у БД був використаний метод пошуку неявних зв'язків атрибутів. Цей метод потребує модифікації, виходячи з того, що одна з БД представлена у вигляді множини HTML-сторінок.

Нехай існує деяка сутність  $S_1$ , яка складається з  $m$  значень атрибутів:

$$S_i = \{v_1, v_2, v_3, \dots, v_m\} \quad (1)$$

Також існує web-сторінка, на якій представлені дані з сутності  $S_2$ , яка складається з атрибутів, що зберігають схожі знання з  $S_1$  та представлених у якості тегів. Безпосередній доступ до сутності  $S_2$  відсутній. Тому доступ до нього можливо організувати лише через

інтерфейс WEB-сторінки та відповідні теги з використанням спеціального програмного модуля узгодження даних ( рис. 1).

Будь-яке узгодження можливе лише при наявності схеми однозначної відповідності атрибутів різних БД. Для цього запропоновано пошук функції залежності між значеннями атрибутів. Кожний елемент множин  $S_i$  є носієм знань про визначений суб'єкт сутності. Залежність між атрибутами визначена певною функцією  $F$  у вигляді нечіткої множини:

$$F = \{ \mu_F(v_1)/v_1, \mu_F(v_2)/v_2, \dots, \mu_F(v_m)/v_m \}, \quad (2)$$

де  $\mu_F(v_i) \in [0..1]$  — оцінка статистичної вірогідності зв'язку по атрибуту  $v_i$ .

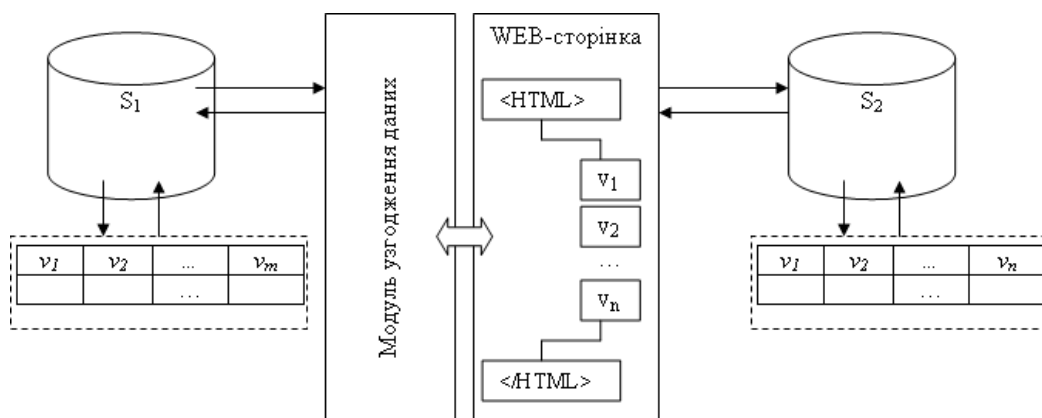


Рис. 1 – Схема взаємодії двох БД через WEB-інтерфейс

Операції об'єднання нечітких множин відповідає операція  $\max$ , що виконується над їх функціями приналежності:

$$\mu_F(v_j) = \max(\mu_{F_i}(v_j)), i \in [1..n], j \in [1..m] \quad (3)$$

В умовах існування сутностей коефіцієнт  $\mu_F$  визначається як статистична приналежність визначеним умовам.

Для знаходження функції  $F(S_1, S_2)$  потрібно визначити матрицю залежностей значень атрибутів  $z_{ij}$   $i \in [1..n], j \in [1..m]$  обраних сутностей.

Залежність між двома значеннями атрибутів визначимо як статистичну величину відношення між значеннями  $v_{i1} \rightarrow v_{i2}$ ,  $i \in [1, n]$  (рис. 2).

Представимо функцію  $f(S_1, S_2)$  у вигляді об'єднання декількох простих функціональних залежностей:

$$F(S_1, S_2) \rightarrow f_1(v_{11}, v_{12}) \vee f_2(v_{11}, v_{22}) \vee \dots \vee f_n(v_{11}, v_{n2}) \vee f_{n+1}(v_{21}, v_{12}) \vee \vee f_{n+2}(v_{21}, v_{22}) \vee \dots \vee f_{2-n}(v_{21}, v_{n2}) \vee \dots \vee f_{n-n}(v_{n1}, v_{n2}) \quad (4)$$

Визначимо функціональні залежності для значень  $v_{11} \rightarrow v_{12}, v_{22}, \dots, v_{n2}$ , тобто значення  $f_1(v_{11}, v_{12})$ . Для цього розділимо всі значення  $v_{j1}$  на  $g$  груп в залежності від унікальності значення. Кожній із груп відповідає умова критеріального відбору  $k_{li}, i \in [1, g], l \in [1, g]$ . При чому для кожної сутності існує свій критерій відбору.

Наступний розподіл значень атрибутів в сутностях за групами відбувається з урахуванням першого розподілу.

Перший розподіл відбувається з урахуванням  $k_{li}, i \in [1, g]$ . Тобто всі значення першого атрибуту  $v_l$  розподіляються на групи у відповідності з критерієм відбору. Тобто формуються множини  $G_{li}$  значень атрибутів, де всі елементи множини відповідають критерію  $k_{li}$ . Другий атрибут  $v_2$  зв'язаний з першим, як носій інформації про один і той самий суб'єкт аналізу: студент, співробітник, тощо, або за ключем, якщо ці атрибути знаходяться в межах однієї БД.  $v_2$  також розподіляється за групами у відповідності до  $k_{2i}, i \in [1, g]$ .

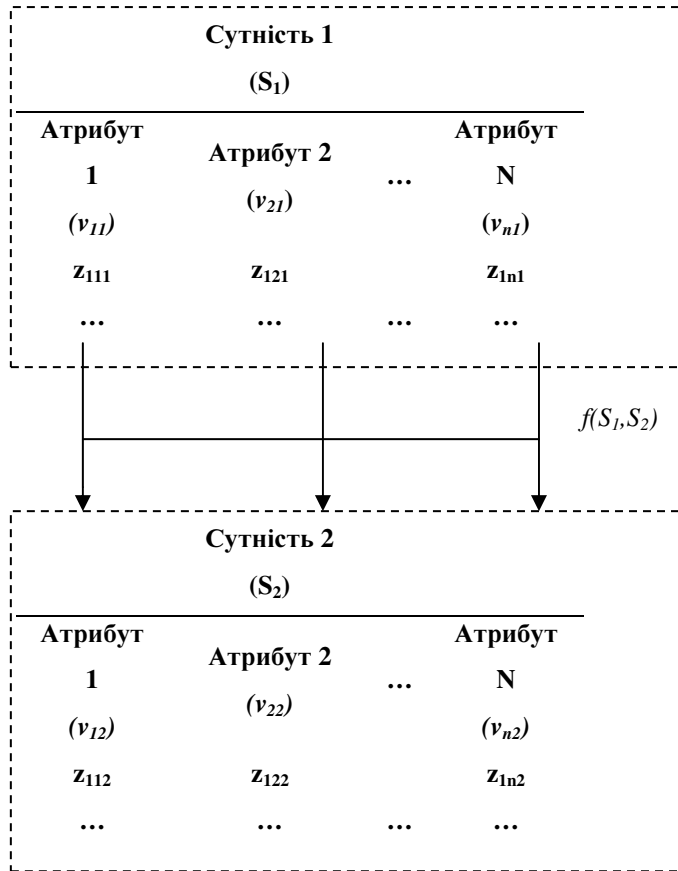


Рис. 2. Схема залежностей атрибутів в двох сутностях.

$$\left[ \begin{array}{l} S_1 = \left\{ \begin{array}{l} \forall k_{11} \exists v_i \in (v_1, v_2, v_3, \dots, v_n) \\ \forall k_{12} \exists v_i \in (v_1, v_2, v_3, \dots, v_n), v_i \in k_{12} \\ \dots \\ \forall k_{1z} \exists v_i \in (v_1, v_2, v_3, \dots, v_n), v_i \in k_{1z} \end{array} \right. \\ S_2 = \left\{ \begin{array}{l} \forall k_{21} \exists v_i \in (v_1, v_2, v_3, \dots, v_n), v_i \in k_{21} \\ \forall k_{22} \exists v_i \in (v_1, v_2, v_3, \dots, v_n), v_i \in k_{22} \\ \dots \\ \forall k_{2z} \exists v_i \in (v_1, v_2, v_3, \dots, v_n), v_i \in k_{2z} \end{array} \right. \end{array} \right. \quad (5)$$

Розподіл значень атрибутів в сутностях представлений на рис. 3.

Автоматизація і комп'ютерні технології

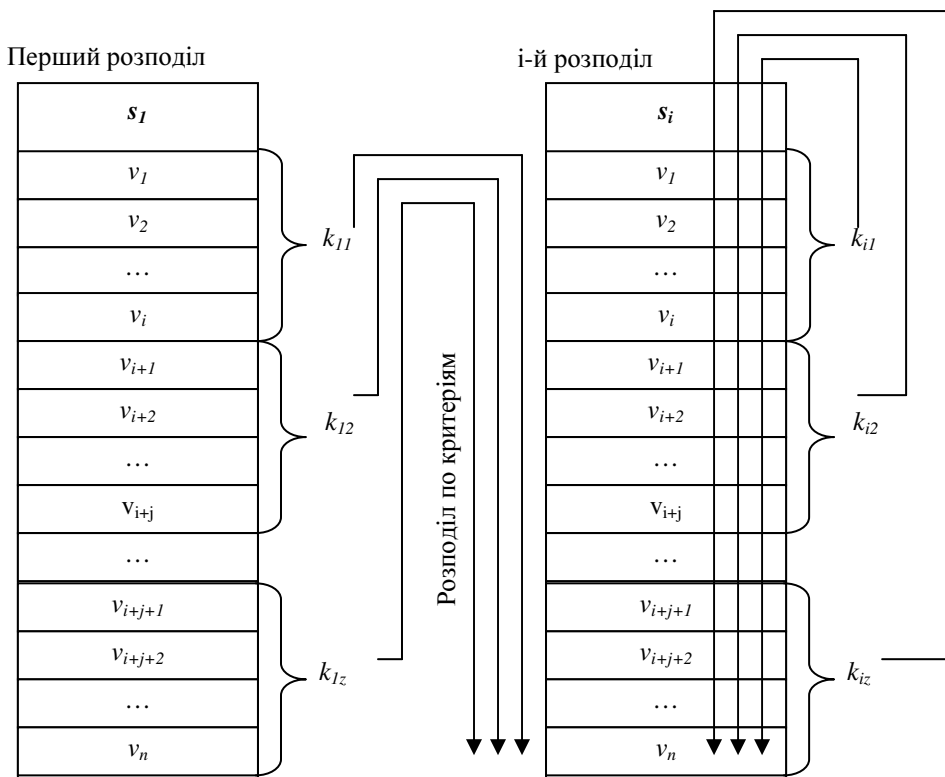


Рис. 3. Розподіл значень атрибутів в сутностях.

Спочатку потрібно визначити, який відсоток з груп другого розподілу  $G_{2i}$  відповідає групам з першого. Дані таких розрахунків розміщуються в таблиці, формат якої представлений нижче.

Таблиця 1

Таблиця розрахунків відповідності першого розподілу

		Другий розподіл			
		$G_{21}$	$G_{22}$	...	$G_{2m}$
Перший розподіл	$G_{11}$	$\mu_1(G_{11}, G_{21})$	$\mu_1(G_{11}, G_{22})$	...	$\mu_1(G_{11}, G_{2m})$
	$G_{12}$	$\mu_1(G_{12}, G_{21})$	$\mu_1(G_{12}, G_{22})$	...	$\mu_1(G_{12}, G_{2m})$
	...	...	...	...	...
	$G_{1n}$	$\mu_1(G_{1n}, G_{21})$	$\mu_1(G_{1n}, G_{22})$	...	$\mu_1(G_{1n}, G_{2m})$

Таким чином, в таблиці будуть знаходитися дані по відсотковому відношенню груп одного розподілу до груп іншого, або композиція нечітких відношень  $v_1$  до  $v_2$ .

Для того, щоб визначити функцію  $\mu_1(G_{1n}, G_{2m})$ , необхідно визначити відношення кількості співпадань по ключу елементів розподілів до загальної кількості елементів другого розподілу:

$$\mu_1(G_{1n}, G_{2m}) = \frac{\text{count}(G_{1n} \cap G_{2m})}{m} \quad (6)$$

Наступним кроком буде визначення відсотка груп першого розподілу  $G_{1i}$  який відповідає групам з другого. Дані таких розрахунків розміщуються в таблиці, формат якої представлений нижче:

Таблиця 2

**Таблиця розрахунків відповідності другого розподілу**

		<i>Перший розподіл</i>			
		$G_{11}$	$G_{12}$	...	$G_{1n}$
Другий розподіл	$G_{21}$	$\mu_2(G_{21}, G_{11})$	$\mu_2(G_{21}, G_{12})$	...	$\mu_2(G_{21}, G_{1n})$
	$G_{22}$	$\mu_2(G_{22}, G_{11})$	$\mu_2(G_{22}, G_{12})$	...	$\mu_2(G_{22}, G_{1n})$
	...	...	...	...	...
	$G_{2m}$	$\mu_2(G_{2m}, G_{11})$	$\mu_2(G_{2m}, G_{12})$	...	$\mu_2(G_{2m}, G_{1n})$

Таким чином в таблиці будуть знаходитися дані по відсотковому відношенню груп другого розподілу до груп першого. Для того, щоб визначити функцію  $\mu_2(G_{2m}, G_{1n})$ , необхідно визначити відношення кількості співпадань по ключу елементів розподілів до загальної кількості елементів першого розподілу:

$$\mu_2(G_{2m}, G_{1n}) = \frac{\text{count}(G_{1n} \cap G_{2m})}{n} \quad (7)$$

Для відображення більш суб'єктивної оцінки ступеню залежностей використаємо критерій Севіджа. Для цього необхідно визначити так звану «матрицю сподівань», елементи якої визначаються за правилом [6]:

$$G'_{ij} = G_{ij} - \max_j(G_{ij}) \quad (8)$$

До цієї матриці використовуємо критерій мінімакса, згідно якого в кожному рядку необхідно визначити мінімальне значення  $G'_{ij}$ , а потім знайти максимальне значення з них:



$$\max_i (\min_j G_{ij}^j) = \max_i (\min_j (G_{ij} - (\max_i G_{ij}))) \quad (9)$$

Саме критерій мінімакса буде відобразити ступінь зв'язку між атрибутами в композиції.

Всі перелічені вище дії повинні виконуватись автоматизованою системою в певному порядку автоматично без втручання користувача [6, 7].

**Програмна реалізація методу емуляції дій користувача.** На основі запропонованого методу розроблено програмний модуль.

При написанні системи головним класом виступає TWebBrowser. Елемент TWebBrowser являє собою ядро логіки Internet Explorer та зберігається у файлі shdocvw.dll. Для виводу HTML елемент управління TWebBrowser створює екземпляр елемента управління Document, що реалізований у файлі mshtml.dll. Елемент управління Document надає реалізацію об'єктної моделі Dynamic HTML (DHTML). Ця об'єктна модель представляє кожний іменований об'єкт HTML-сторінки як об'єкт COM.

Об'єкт Document реалізує інтерфейс IHtmlDocument2. Перехід від елемента управління TWebBrowser до елемента Document зводиться до отримання властивостей IwebBrowser2 елемента TWebBrowser і запиту інтерфейса IHtmlDocument2. Доступ до будь-якого іменованого елемента HTML-сторінки – тобто до будь-якого тега з атрибутом name або id – виконується через властивість all інтерфейса IHtmlDocument2.

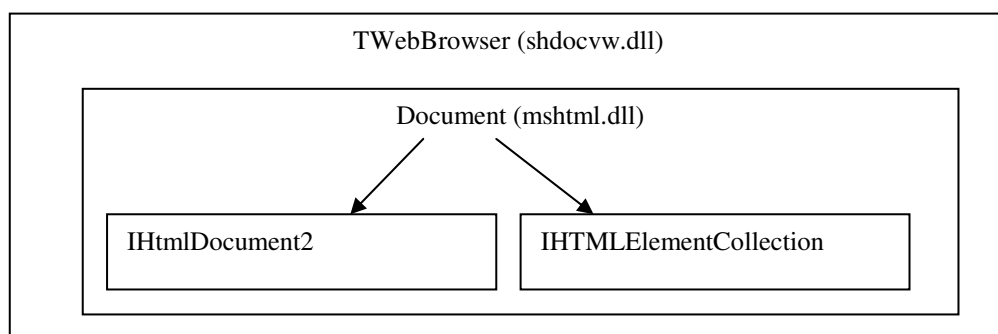


Рис. 4 – Ієрархічна схема класу TWebBrowser

Приведемо приклад вивантаження списку студентів визначеної групи з ЄДЕБО в локальну БД.

Відбувається пошук всіх елементів WEB-сторінки. Після знаходження елемента з ID 'MainContent\_ASPxRoundPanel1\_ASPxTextBox\_Login' в його властивість *value* записується логін доступу. Відповідно до елементу 'MainContent\_ASPxRoundPanel1\_ASPxTextBox\_Password' записується пароль доступу. Після знаходження елемента 'button' відбувається виконання його методу *inpulement.click*.

Після авторизації програма повина перейти до сторінки зі списком груп. Для цього використовується метод *navigate*, класу *TwebBrowser*. Після чого потрібно знайти визначену групу. Тому знову організуємо пошук потрібного текстового поля для фільтру. Після знаходження присвоюємо властивості *value* значення групи *gr* та емулюємо натиснення мишою.

Код фільтрації наведено нижче:

```
if inpulement.id =  
'MainContent_ASPxRoundPanel_University_UniversityHeadContent_ASPxGridView_  
Groups_DXFREditorcol2_I' then  
begin  
WebBrowser.OleObject.Document.forms.item(0).elements.item(i-1).value:=gr;  
inpulement.focus;  
WebBrowser.OleObject.Document.forms.item(0).elements.item(i-1).onchange;  
inpulement.onkeydown; inpulement.onkeyup;  
end;
```

Доступ до елементів таблиці організуємо за допомогою тегу *TABLE*. Після цього емулюється натиснення кнопки «Студенти» для переходу до таблиці списку групи. Так як кнопка організована, як комірка таблиці, то метод *click* застосовуємо до неї:

```
for i:=0 to WebBrowser1.OleObject.Document.all.tags('TABLE').length-1 do  
if WebBrowser.OleObject.Document.all.tags('TABLE').item(i).id=  
'MainContent_ASPxRoundPanel_University_UniversityHeadContent_ASPxMenu2'  
then  
begin  
ovTable := WebBrowser.OleObject.Document.all.tags('TABLE').item(i+1);  
ovTable.Rows.Item(0).Cells.Item(36).click;  
break; end;
```

Після переходу до таблиці списку студентів, потрібно визначити їх загальну кількість та кількість сторінок списку. Це потрібно для організації автоматичного переходу по списку. Для прикладу, експорт списку організований у текстовий файл *students.txt*. Код експорту списку приведений нижче.

```

while j<pages do begin
j:=j+1;
Webbrowser.Document.QueryInterface(IHTMLDocument2, iDoc1);
for i:=0 to WebBrowser.OleObject.Document.all.tags('TABLE').length-1 do
if
WebBrowser.OleObject.Document.all.tags('TABLE').item(i).id='cGV_Students_DXM
ainTable' then
begin
ovTable := WebBrowser1.OleObject.Document.all.tags('TABLE').item(i);
ovTablePage :=
WebBrowser1.OleObject.Document.all.tags('TABLE').item(i+2);
break;
end;
pages:=round((ovTablePage.Rows.Item(0).Cells.Length-5)/2);
for i:=0 to ovTable.Rows.Length-1 do
begin
Memo1.Lines.Add(ovTable.Rows.Item(i).Cells.Item(1).innertext+
' '+ovTable.Rows.Item(i).Cells.Item(2).innertext);
end;
s:='javascript:aspxGVPagerOnClick("cGV_Students","PBN");';
ExecuteScript(iDoc1,s,'JavaScript');
end; Memo1.Lines.SaveToFile('students.txt');

```

Перед виконанням кожних дій потрібно очікувати повного завантаження сторінки. Тільки після цього можна переходити до наступної дії. Для цього використовується подія *OnDocumentComplete* класу *TWebBrowser*.

**Висновки.** Розроблений метод емуляції дій користувача є достатньо універсальним та може бути застосований для будь-яких WEB-сайтів чи сервісів, навіть якщо безпосередній доступ до БД закритий. Метод точно відтворює всі дії користувача в стандартних ситуаціях, тому він значно пришвидшує роботу по перенесенню даних з ЄДЕБО до будь-яких інших сховищ даних.

Після впровадження даного методу був проведений експеримент, який полягав в імпорті списку всіх студентів по групам першого курсу.

Автоматизація і комп'ютерні технології

Імпорт відбувався за допомогою можливості експорту списку у таблицю Excel, з подальшим копіюванням у власну автоматизовану систему, та за допомогою методу емуляції дій користувача на WEB-сторінці. Всього організувалось копіювання 263 студентів з 10 груп. Дані представлені в таблиці 3.

Таблиця 3

**Порівняльні дані експерименту**

Метод експорту в Excel з подальшим ручним перенесенням до АС		Метод емуляції дій користувача на WEB-сторінці		
№ з/п	Опис дій	Час (сек.)	Опис дій	Час (сек.)
1	Пошук визначених груп студентів	21	Пошук визначених груп студентів	14
2	Відкриття списку студентів	56	Відкриття списку студентів	28
3	Експорт списку в Excel	33	-	-
4	Відкриття документів для копіювання даних	23	-	-
5	Копіювання даних	1257	Безпосереднє копіювання з таблиці WEB-сторінки	124
Загальний час		1390	Загальний час	166

Проведений експеримент довів ефективність впровадження методу емуляції дій користувача на WEB-сторінці, за рахунок організації автоматизованих стандартних дій, таких як пошук інформації, відкриття визначеної сторінки та копіювання даних. Перенесення 263 студентів за допомогою запропонованого методу відбулося у 8,4 рази швидше, ніж класичним методом, що каже про доцільність використання емуляції дій користувача на WEB-сторінці для узгодження з різноманітними сховищами даних.

**Література**

1. Гир Дэвид, Федеративный доступ к базам данных [Электронный ресурс] / Режим доступа: [http://citforum.ru/consulting/BI/federative\\_access/](http://citforum.ru/consulting/BI/federative_access/) (17.01.2015)
2. Глушков, В.М. Основы безбумажной информатики [Текст] / В. М. Глушков. – испр: Наука, 1987. – 552 с.
3. Дейт, К.Дж. Введение в системы баз данных / Пер. с англ. – М.: Издательский дом “Вильямс”, 2005. – 1328 с.

Автоматизация і комп’ютерні технології

4. Tuyet-Tram Dang-Ngoc and Georges Gardarin. Federating heterogeneous data sources with xml / PRiSM Laboratory University of Versailles, FRANCE.

5. Neha Sehta, Dr. Suresh Jain. A Fine Grained Access Control Model for Relational Databases/ Sehta Neha, Jain Suresh // (IJCSIT) International Journal of Computer Science and Information Technologies. – № 3(1) . – 2012 .

6. Яглом, И. Вероятность и информация [Текст] / И. Яглом, А. Яглом — М.: Наука, 1973 — 512 с.

7. Богачёв, В.И. Действительный и функциональный анализ [Текст]: Университетский курс / В.И. Богачёв, О.Г. Смолянов — Ижевск: НИЦ Регулярная и хаотическая динамика, 2009.

*Надійшла до редакції 22.12.2014*