

## РЕАЛИЗАЦИЯ СИСТЕМЫ ПРЕДСТАВЛЕНИЯ КОНТЕКСТОВ В ИНФОРМАЦИОННОМ ЛИНГВИСТИЧЕСКОМ РЕСУРСЕ

© 2020 О. В. Дони́на✉

*Воронежский государственный университет  
Университетская пл., 1, 394018 Воронеж, Российская Федерация*

**Аннотация.** В статье предлагается рассмотреть второй этап обновления информационного лингвистического ресурса «COEL» (Cryptotypes of the English Language), разработанного на кафедре теоретической и прикладной лингвистики ВГУ в качестве инструмента для исследовательской работы в рамках теории криптоклассного анализа на материале данных сочетаемости абстрактных имен английского языка. В рамках данного этапа была реализована система представления контекстов. Среди основных используемых технологий стоит выделить: PHP, SQL, JavaScript, язык разметки HTML, язык формирования внешнего вида документа CSS и технология AJAX. В работе рассматривается веб-интерфейс информационной системы, обосновывается необходимость вносимых изменений и подробно описываются проведенные усовершенствования. Основными достигнутыми результатами являются: 1) система выдачи контекстов изменена в соответствии с особенностями новых данных: в новых разделах ресурса была реализована возможность фильтрации выводимого списка словоупотреблений по варианту английского языка; 2) реализована система выгрузки контекстов для предоставления возможности работы с информацией на машине пользователя и локального хранения в случае соответствующей необходимости, что может оказать пользу лингвистам в процессе проведения исследований или обучающимся при изучении английского языка. Указанные изменения позволили адаптировать систему для осуществления работы с данными, полученными из лингвистических баз данных — корпусов GloWbE (The corpus of Global Web-based English), NOW (News On the Web) и iWeb, а также сделать ее более функциональной и удобной для использования в рамках исследовательской лингвистической деятельности и при изучении английского языка.

**Ключевые слова:** информационный лингвистический ресурс, PHP, SQL, JavaScript, язык разметки HTML, язык формирования внешнего вида документа CSS, технология AJAX, корпусные исследования, криптоклассный анализ.

### ВВЕДЕНИЕ

В нашей предыдущей статье [1] был описан первый этап расширения функционала информационного лингвистического ресурса «COEL» (Cryptotypes of the English Language) «Криптоклассы английского языка», заклю-

чающийся в определении оптимального расположения новых данных, настройке доступа к БД и обновлении системы пополнения базы данных.

Рассматриваемый ресурс была разработана в качестве инструмента для исследовательской работы в рамках теории криптоклассного анализа на материале данных сочетаемости абстрактных имен английского языка [2]. Необходимость его обновления возникла в связи с расширением криптоклассной теории

---

✉ Дони́на Ольга Валерьевна  
e-mail: [olga-donina@mail.ru](mailto:olga-donina@mail.ru)



Контент доступен под лицензией Creative Commons Attribution 4.0 License.

The content is available under Creative Commons Attribution 4.0 License.

и использованием новых источников лингвистического материала: если изначально данные для лингвистического анализа отбирались только в корпусе (т.е. лингвистической базе данных) COCA (Corpus of Contemporary American English) [3], то теперь анализируется также материал из ресурсов «GloWbE» (The corpus of Global Web-based English) [4], «NOW» (News On the Web) [5] и «iWeb» [6], основной особенностью которых является наличие контекстов употребления различных вариантов английского языка.

В данной работе будет описан порядок действий, связанных со вторым этапом обновления информационного ресурса «COEL» — с реализацией системы представления контекстов.

## 1. МАТЕРИАЛЫ И МЕТОДЫ

Как и у большинства лингвистических ресурсов, интерфейс «COEL» рассчитан на пользователей двух типов: неавторизованный пользователь и администратор. На всех страницах ресурса пользователь имеет доступ к меню, позволяющему просмотреть страницы «Существительные», «Классификаторы», «Криптоклассы», «Контексты». Страница «Контексты» содержит три выпадающих меню («Криптокласс», «Классификатор» и «Существительное»), используя которые,

пользователь может получить доступ к списку контекстов. Программный код данных выпадающих списков, содержащийся в файле contexts.php, представляет собой набор контейнеров select. Пример данного контейнера изображен на рисунке (рис. 1).

Каждое выпадающее меню имеет набор опций. В HTML для этого используется тег option. В данном случае содержимым контейнеров является значение переменных options1, options2 или options3. На рис. 2 (рис. 2) изображен участок кода, выполняющий определение значения переменной options3.

Данным значением является тег option, в качестве текста которого выступает массив noun с индексом name. В свою очередь значением \$noun['name'] является содержимое поля name таблицы [pr]nouns, полученное в результате запроса к базе данных с применением функции «select\_assoc». Параметры данной функции — переменные \$select, \$table, \$where, \$order, \$limit. Пример SQL-запроса (для «COCA»), выполняемый функцией «select\_assoc», изображен на рис. 3 (рис. 3). Результатом данного запроса является информация, содержащаяся в строках полей id и name, которые используются при создании выпадающего списка. Записи поля name используются как содержимое тега option, которое видит пользователь в выпадающем списке, а записи поля id являются атрибутами value данного тега.

```
<select id=»noun_select» name=»noun_select» onChange=»noun_change( )»>
  <?php echo $options3; ?>
</select>
```

Рис. 1. Программный код выпадающего списка «Существительное»  
[Fig. 1. The program code of the drop-down list «Noun»]

```
$noun_id = 0;
$nouns $db->select_assoc('', '[pr]nouns', «id in (SELECT noun_id FROM [pr]compatibility
WHERE classifier_id = $classifier_id)», 'name', '');
$options3 = '';
$fl3 = false;
foreach ($nouns as $noun) {
  $options3 .= '<option value=»' . $noun['id'] . '»>' . $noun['name'] . '</option>';
  if (!$fl3) {
    $noun_id = $noun['id'];
    $fl3 = true;
  }
}
```

Рис. 2. Определение значения переменной options3  
[Fig. 2. Determining the value of the options3 variable]

```
SELECT id, name
FROM coca_nouns
WHERE id in (SELECT noun_id
             FROM coca_compatibility
             WHERE classifier_id = 351)
ORDER BY name
```

Рис. 3. SQL-запрос для получения данных полей *id* и *name* таблицы *coca\_nouns*  
 [Fig. 3. SQL-request for receiving the data of the *id* and *name* fields of the *coca\_nouns*

```
if ($_SESSION['database'] != 1) {
    $language_id = 0;
    $languages = $db->select_assoc('', '[pr]languages', '', 'name', '');
    $options4 = '';
    $f14 = false;
    foreach ($languages as $language) {
        $options4 .= '<option value=' . $language['id'] . '>' . $language['name'] .
            '</option>';
        if (1$14) {
            $language_id = $language['id'];
            $f14 = true;
        }
    }
}
```

Рис. 4. Программный код, определяющий значение переменной *options4*  
 [Fig. 4. Program code defining the value of the *options4* variable]

```
$cnt = $db->select_assoc('id', '[pr]contexts', «combination_id in (SELECT id FROM [pr]
compatibility WHERE noun_id = $noun_id
AND classifier_id = $classifier_id)», '', '1');
If ($cnt['id']) {
    $out = '';
    $contexts = $db->select_assoc('', '[pr]contexts', «combination_id in (SELECT id FROM
[pr]compatibility WHERE noun_id = $noun_id AND classifier_id = $classifier_id)», '',
'');
    $i = 0;
    foreach ($contexts as $context) {
        ...
    }
}
```

Рис. 5. Программный код, выполняющий поиск контекстов  
 [Fig. 5. Program code of searching for contexts]

Четвертое выпадающее меню (используется в разделах «GloWbE», «NOW» и «iWeb»), предназначенное для выбора варианта английского языка, было реализовано по тому же принципу (рис. 4). Добавлена переменная *options4*, значением которой является тег *option* с данными, полученными в результате выполнения запроса, аналогичного изображенному выше. Единственное отличие — значение переменной *table* меняется на *[pr]languages*. В качестве особенности данного выпадающего списка можно выделить то, что он отображается только в том случае,

если значением переменной сессии является число, не равное единице (т.е. пользователь находится в разделе «GloWbE», «NOW» или «iWeb»).

Отображение списка контекстов происходит в результате выполнения двух запросов к базе данных и построения таблицы. На рис. 5 (рис. 5) изображен участок кода, отвечающий за поиск контекстов в таблице *[pr]contexts*.

Первым запросом к базе данных является проверка существования комбинации из идентификаторов существительного и классификатора. На рис. 2 (рис. 2) видно, что пе-

```
$out .= '<tr' . $class . '>'
      '<td id=' . $context['id'] . '>>' . $context['context'] . '</td>'
      '<td>' . $editing_buttons . '</td>'
      '</td>';
      $i++;
```

Рис. 6. Программный код таблицы контекстов на странице contexts.php

[Fig. 6. The program code of the context table on the contexts.php page]

```
if ($_SESSION['database'] != 1) {
    $cnt = $db->select_assoc('id', '[pr]contexts', «combination_id in (SELECT id
FROM [pr]compatibility WHERE noun_id = $noun_id AND classifier_id = $classifier_id AND
language_id = $language_id )», '', '1');
    if ($cnt['id']) {
        $out = '';
        $contexts = $db->select_assoc('', '[pr]contexts', «“combination_id in (SELECT id
FROM [pr]compatibility WHERE noun_id = $noun_id AND classifier_id = $classifier_id AND
language_id = $language_id )», '', '');
        $i = 0;
        foreach ($contexts as $context) {
            ...
        }
    }
    ...
}
```

Рис. 7. Программный код поиска контекстов, вызываемый в новых

[Fig. 7. Context search program code called in new sections]

ременной `noun_id` присваивается значение массива `noun['id']`, которое используется при поиске контекстов по параметрам, выбранным в выпадающем списке. Таким образом, значение переменных `noun_id` и `classifier_id` равно `id` выбранных пользователем существительных и классификаторов. Если комбинация значений `$noun_id` и `$classifier_id` найдена, выполняется второй запрос, результатом которого является список контекстов по данной комбинации. В случае успешного выполнения второго запроса переменной `out` присваивается значение в виде HTML-таблицы, содержимым которой является результат второго запроса (рис. 6).

Для реализации возможности получения таблицы с контекстами в разделах «GloWbE», «NOW» и «iWeb», запросы, описанные выше, были продублированы и расширены. На рис. 7 (рис. 7) показано, что данные запросы содержат дополнительное условие в виде идентификатора языка и выполняются в новых разделах. Результатом проведения вышеописанных действий является возможность получения списка контекстов в новых разделах системы «СОЕЛ».

На наш взгляд, недостатком первоначальной версии страницы «Контексты» является необходимость выбирать существительные и языки среди всех доступных в системе. Так, в разделе «СОСА» выпадающее меню «Существительные» содержит около 500 имен, многие из которых не обладают метафорической сочетаемостью с некоторыми классификаторами. Таким образом, существующая система обладает избыточностью, что существенно усложняет задачу поиска: выбирая определенные параметры, пользователь может получить пустую таблицу. Именно этот факт выступил основной отправной точкой последовавшего процесса оптимизации системы предоставления контекстов, результаты которого рассмотрены далее.

## 2. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Для оптимизации процесса получения контекстов по заданным параметрам было принято решение использовать принцип, реализованный в меню выбора криптокласса. При изменении текущего параметра на необходимый, содержимое списка «Класси-

```
function crypt_change_context( ){
    var crypt_id = document.getElementById(«crypt_select»).value;
    var xmlhttp = getXmlHttp( );
    xmlhttp.open(‘GET’, site +’/ajax.php?action=context_get_classifiers&crypt_id=’ +
                                                    crypt_id, true);

    xmlhttp.onload = function() {
var html = xmlhttp.responseText;
        document.getElementById(«classifier_select»).innerHTML = html;
        cf_change( );
    }
    xmlhttp.send(null);
}
```

Рис. 8. Функция, выполняющаяся при смене активного параметра в меню «Криптоклассы»  
[Fig. 8. The function that is performed when changing the active parameter in the «Cryptoclasses» menu]

```
function context_get_verbs($crypt_id) {
    GLOBAL $db;
    $cnt = $db->count(‘[pr]classifiers’, «id in (SELECT classifier_id FROM [pr]
cryptoclasses_classifiers WHERE cryptoclass_id = $crypt_id )»);
    if ($cnt) {
        $out = ‘’;
        $classifiers = $db->select_assoc(‘’, ‘[pr]classifiers’, «id in (SELECT classifier_
idcFROM [pr]cryptoclasses_verbs WHERE cryptoclass_id = $crypt_id )», ‘name’, ‘’);
        $i = 0;
        foreach ($classifiers as $classifier) {
            $out .= ‘<option value=»’ . $classifier[‘id’] . ‘>>’ . $classifier[‘name’] . ‘</
option>’;
            $i++;
        }
    }
    else {
        $out = ‘’;
    }
    echo $out;
}
```

Рис. 9. Функция, выполняющая определение списка параметров меню «Классификатор»  
[Fig. 9. Function that defines the list of parameters of the menu «Classifier»]

фикатор» меняется без перезагрузки страницы. Отсутствие необходимости обновления страницы при совершении серверных запросов реализовано при помощи технологии AJAX (Asynchronous Javascript And Xml). На рис. 2 (рис. 2) мы можем увидеть событие `onChange="noun_change()"`, происходящее после изменения активного пункта выпадающего меню. При смене выбранного криптокласса выполняется функция «`crypt_change_context`». Данная функция прописана в файле `ling.js` и отражена на рис. 8 (рис. 8).

Переменной `crypt_id` присваивается значение, равное идентификатору выбранного при помощи выпадающего списка криптокласса. После получения данного идентифи-

катора происходит обращение к функции «`context_get_classifiers`» (рис. 9), которая расположена в файле `ajax.php`.

При смене активного пункта выпадающего меню выполняются повторные запросы к базе данных для изменения параметров следующего выпадающего списка и вывода обновленной таблицы. Как видно из рис. 10 (рис. 10), после получения нужной информации вызывается функция «`classifier_change`». Действием, выполняемым функцией «`classifier_change`» (вызывается событием `onChange` при изменении текущего значения меню «Классификатор»), является вызов функции «`noun_change`» (также вызывается при изменении текущего значения меню «Существительное»),



```
function classifier_change( ){
    noun_change( );
}
function noun_change( ){
    var classifier_id = document.getElementById(«classifier_select»).value;
    var noun_id = document. getElementById(«noun_select»). value;
    var xmlhttp = getXmlHttp( );
    xmlhttp.open( 'GET', site +' /ajax.php?action=context_get_contexts&classifier_id=' +
classifier_id + '&noun_id=' + noun_id, true);
    xmlhttp.onload = function() {
        var html = xmlhttp. responseText;
        document. getELEMENTById(«context_table»).innerHTML = html;
        ...
    };
    xmlhttp.send(null);
}
```

Рис. 10. Функции, выполняющиеся при изменении активных параметров списков «Классификатор» и «Существительное»

[Fig. 10. Functions performed when changing the active parameters of the «Classifier» and «Noun» lists]

```
<div class=»lLabel-div-2»>Криптокласс:</div>
<?php if ($_SESSION['database'] == 1) {?>
<select name=»crypt_select» id=»crypt_select» onChange=»crypt_change_context1()»<?php
echo $options1; ?></select>
<?php } ?>
<?php if ($_SESSION['database'] != 1) {?>
<select name=»crypt_select» id=»crypt_select» onChange=»crypt_change_context()»<?php
echo $options1; ?></select>
<?php } ?>
```

Рис. 11. Выпадающие списки, зависящие от выбранного раздела системы  
[Fig. 11. Drop-down lists depending on the selected system partition]

которая получает данные из списков «Классификатор» и «Существительное», а затем, обращаясь к функции «context\_get\_contexts», меняет таблицу контекстов на ту, что соответствует выбранным пользователем параметрам. При этом содержимое выпадающего списка «Существительное» не зависит от примененных ранее параметров.

Для реализации возможности работы с выпадающими списками, содержимое которых меняется в зависимости от выбранных параметров, необходимо учесть, что на странице «Контексты» в разделах системы «СОЕЛ» присутствует разное количество выпадающих списков. В зависимости от выбранного раздела функции, вызываемые изменением активного пункта любого из меню, должны отличаться. Для этого было решено продублировать функции «crypt\_change\_context», «classifier\_change» и «noun\_change»,

а те, что уже используются, переименовать в «crypt\_change\_context1», «classifier\_change1» и «noun\_change1» соответственно. В файле contexts.php было указано, что переименованные функции должны вызываться, если выбран раздел 1 («СОСА»), а новые функции — если выбран раздел 2 («GloWbE»), 3 («NOW») или 4 («iWeb») (рис. 11).

Программный код функций «crypt\_change\_context» и «crypt\_change\_context1» аналогичен по своей структуре. Отличием является только то, что последняя вызывает «classifier\_change1», а не «classifier\_change». Далее в файле ajax.php необходимо создать функцию, вызываемую при выполнении «classifier\_change1» и «classifier\_change», которые отвечают за получение списка существительных (рис. 12).

Для выполнения «context\_get\_nouns» необходимо создать javascript-функции «classifier\_

```
function context_get_nouns($classifier_id) {
    GLOBAL $db;
    $cnt = $db->count('[pr]nouns', «id in (SELECT noun_id FROM [pr]compatibility WHERE classifier_id = $classifier_id)»);
    if ($cnt) {
        $out = '';
        $nouns = $db->select_assoc('', '[pr]nouns', «id in (SELECT noun_id FROM [pr]compatibility WHERE classifier_id = $classifier_id)», 'name', '');
        $i = 0;
        foreach ($nouns as $noun) {
            $out .= '<option value=' . $noun['id'] . '>' . $noun['name'] . '</option>';
            $i++;
        }
    }
    else {
        $out = '';
    }
    echo $out;
}
```

Рис. 12. Функция, выполняющая определение списка параметров меню «Существительное»  
[Fig. 12. Function that determines the list of parameters of the menu «Noun»]

```
function classifier_change1( ){
    var modify_id = document.getElementById(«classifier_select»).value;
    var xmlhttp = getXmlHttp();
    xmlhttp.open('GET', site +'/ajax.php?action=context_get_nouns&classifier_id=' + classifier_id, true);
    xmlhttp.onload = function() {
        var html = xmlhttp.responseText;
        document.getElementById(«noun_select»).innerHTML = html;
    };
    xmlhttp.send(null);
    noun_change1( );
}
```

Рис. 13. Обновленная функция «classifier\_change»  
[Fig. 13. Updated classifier\_change function]

change1» и «classifier\_change» по аналогии с функцией «stupt\_change\_context». На рисунке (рис. 13) изображен «classifier\_change1»: получая данные элемента «classifier\_select» (т. е. выпадающего списка «Классификатор»), функция обращается к «context\_get\_nouns», и, получив результат, обновляет содержимое элемента «noun\_select» (меню «Существительное»). Функции «classifier\_change» и «classifier\_change1» обращаются к «context\_get\_nouns» независимо от раздела системы. Единственным отличием данных функций друг от друга является функция, вызываемая ими: «noun\_change1» и «noun\_change» в зависимости от выбранного пользователем раздела «COEL».

Для получения списка контекстов в ajax.php были созданы функции «context\_get\_contexts» для корпусов «GloWbE», «NOW», «iWeb» и «context\_get\_contexts1» для корпуса СОСА. Указанные функции частично дублируют программный код, отвечающий за получение списка контекстов в context.php, а отличием является то, что «context\_get\_contexts» выполняет запрос к базе данных с учетом трех параметров: существительное, классификатор и язык (параметрами, используемыми функцией «context\_get\_contexts1» при обращении к базе данных, являются существительное и классификатор).

В первом разделе системы «COEL» javascript-функция «noun\_change1» обращается

```
function context_get_languages($classifier_id, $noun_id) {
    GLOBAL $db;
    $cnt = $db->count('[pr]languages', «id in (SELECT language_id FROM [pr]compatibility
WHERE classifier_id = $classifier_id AND noun_id = $noun_id)»);
    if ($cnt) {
        $out = '';
        $languages = $db->select_assoc('', '[pr]languages', «id in (SELECT language_id
from [pr]compatibility WHERE classifier_id = $classifier_id AND noun_id = $noun_id)»,
'name', '');
        $i = 0;
        foreach ($languages as $language) {
            $out '<option value=' . $language['id'] . '>' . $language['name'] .
            '</option>';
            $i++;
        }
    }
    else {
        $out = '';
    }
    echo $out;
}
```

Рис. 14. Функция, выполняющая определение списка параметров меню «Язык»  
 [Fig. 14. Function that determines the list of parameters of the menu «Language»]

```
function language_change( ){
    var classifier_id = document.getElementById(«classifier_select»).value;
    var noun_id = document.getElementById(«noun_select»).value;
    var Language_id = document. getElementById( 'language_select' ).value;
    var xmlhttp = getXmlHttp( );
    xmlhttp.open( 'GET', site + '/ajax.php?action=context_get_contexts&classifier_id=' +
classifier_id + '&noun_id=' + noun_id + '&language_id=' + language_id, true);
    xmlhttp.onload = function() {
        var html = xmlhttp.responseText;
        document .getElementById(«context_table»).innerHTML = html;
        ...
    }
    xmlhttp.send(null);
}
```

Рис. 15. Функция, выполняющаяся при смене активного параметра меню «Язык»  
 [Fig. 15. Function that is performed when changing the active parameter of the “Language” menu]

к «context\_get\_context1» и строит таблицу с данными. Таким образом, работая с разделом «СОСА», пользователь получает возможность выбирать только те существительные, контексты с которыми существуют в базе данных. При выборе пункта в меню «Классификатор» происходит обновление содержимого списка «Существительное». Данное меню обновляется и при изменении выбранного пункта в меню «Криптокласс», так как это вызывает обновление списка параметров в меню «Классификатор». Построение таблицы при изменении выбранного пункта в лю-

бом из меню происходит благодаря тому, что каждая из функций вызывает следующую. Последняя выполняемая функция выводит таблицу. Для раздела «СОСА» таковой выступает «noun\_change1». В разделах «GloWbE», «NOW» и «iWeb» последним выпадающим списком является «Язык». Соответственно, функцией, выполняющей отображение таблицы, должна быть не «noun\_change», а та, что вызывается при выборе пункта в данном меню. Также содержимое выпадающего списка «Язык» должно обновляться по аналогии с меню «Существительное».



```

$languages = $db->select_assoc('', '[pr]languages', '', '', '');
$i = 0;
foreach ($languages as $language) {
    if (($i % 2) == 0) {
        $class = ' bgcolor=#f2f2f2';
    }
    else) {
        $class = '';
    }
    $out .= '<tr' . $class . '>
        <td id=»td_' . $language['id'] . '»>' . $language['name'] . '</td>
    </tr>';
    $i++;
}

```

Рис. 16. Программный код, выполняющий наполнение таблицы на странице «Языки»  
 [Fig. 16. The program code that performs the filling of the table on the page «Languages»]

```

<div class=»choose-container get-context-info»>
<form action=»export_noun.php» method='POST'>
<input type='hidden' id='crypt_id' name='crypt_id'>
<input type='hidden' id='classifier_id' name='classifier_id'>
<input type='hidden' id='noun_id' name='noun_id'>
<input type=»hidden» id=»language_id» name=»Language_id»>
<input class=»get-btn» onload=»input_ids()» type='submit' value='»Выгрузить контексты'»>
</form>
</div>

```

Рис. 17. Форма, содержащая скрытое поле для передачи информации в `export_contexts.php`  
 [Fig. 17. A form containing a hidden field for transmitting information to `export_contexts.php`]

```

function input_ids(){
    document. getElementById('crypt_id' ).value=$( 'select[name=crypt_select]').val();
    document. getElementById('classifier_id' ).value=$( 'select[name=classifier_select]').
val();
    document. getElementById('noun_id' ).value=$( 'select[name=noun_select]').val();
    document. getElementById('language_id' ). value=$( 'select[name=language_select]').
val();
}

```

Рис. 18. Функция, передающая данные в скрытые поля  
 [Fig. 18. A function that passes data to hidden fields]

Для решения данной задачи в файле `ajax.php` была создана функция «`context_get_languages`» (рис. 14). Данная функция аналогична функциям «`context_get_nouns`» и «`context_get_classifiers`» и отличается только набором параметров запросов к базе данных. Далее в `javascript`-функции «`noun_change`» необходимо с помощью технологии AJAX вызвать функцию «`context_get_languages`» и разместить ответ в выпадающем меню с индексом «`language_select`» с помощью атрибута `responseText`. Далее необходимо создать функцию «`language_change`» и вызвать ее в «`noun_change`».

На рис. 15 (рис. 15) изображена функция «`language_change`».

Она вызывает PHP-функцию, параметрами которой являются классификатор, существительное и язык, а текст ответа размещает в таблице «`context_table`». Для того, чтобы список «Язык» адаптировался под изначально выбранные параметры предыдущих меню при загрузке страницы, программный код, выполняющий определение значения переменной `options4` (рис. 3) был модифицирован так, чтобы функция «`select_assoc`» получала не все идентификаторы языков, а только те, которые комбинируются с `noun_id`

```

if ($_SESSION['database'] != 1) {
    $classifier = $db->select_assoc('', '[pr]classifiers', «id =» . $classifier_id, '', '');
    $noun = $db->select_assoc('', '[pr]nouns', «id =» . $noun_id, 'name', '');
    $language = $db->select_assoc('', '[pr]languages', "id =» . $language_id, '', '');
    $context = $db->select_assoc('context', '[pr]contexts', «combination_id in (select
id from [pr]compatibility where noun_id = « . $noun_id . « and classifier_id =» .
$classifier_id . « AND language_id = « . $language_id .)», '', '');
}
    
```

Рис. 19. Переменные, значениями которых являются результаты SQL-запросов

[Fig. 19. Variables which values are the results of SQL queries]

и classifier\_id в таблице [pr]compatibility. В результате проведенных действий пользователь получает возможность выбирать данные, используя выпадающие списки (в т. ч. «Язык»), содержимое которых меняется в зависимости от выбранных ранее параметров. При изменении выбранного параметра выпадающего меню происходит обновление содержимого всех списков, расположенных ниже. Единственным условием для появления того или иного пункта меню, «Существительное» и «Язык» является наличие соответствующей с идентификатором выбранного классификатора комбинации в таблице [pr]compatibility. Комбинация в данной таблице существует только при наличии соответствующего (-их) ей контекста (-ов) в таблице [pr]contexts. Если все контексты по данной комбинации удаляются, то выполняется ее удаление. Таким образом, работая на странице «Контексты» любого из разделов информационной системы «СОЕЛ», пользователь получает возможность быстро ознакомиться с необходимой информацией, так как набор параметров выпадающих списков зависит от наличия в базе данных контекстов.

В боковом меню размещена ссылка на страницу «Языки» (languages.php). На данной странице размещена таблица, содержимым которой является результат перебора данных, полученных с помощью SQL-запроса. На рис. 16 (рис. 16) изображен программный код, отвечающий за выполнение SQL-запроса для поиска данных о языках и наполнение таблицы. Конструкция «foreach» представляет собой способ перебора массивов. На каждой итерации цикла значение \$languages присваивается переменной language, а внутренний указатель массива увеличивается на единицу (т. е. на следующей итерации обрабатывает-

ся следующий элемент), и происходит запись полученных данных в строку таблицы. Переменной class присваивается атрибут bgcolor, устанавливающий цвет фона элемента. Значение данной переменной используется в контейнере для создания строки таблицы (tr). При этом окрашиваются только те строки, порядковые номера которых являются четными числами. За это отвечает оператор «%», возвращающий остаток от деления. Если остаток равен нулю, число (\$i) является четным. Подобная стилизация позволяет визуально разделять записи в таблицах, содержащих большое количество строк. Таблица, полученная в результате выполнения запроса и перебора, присваивается переменной content, вызываемой в коде разметки данной страницы. Помимо таблицы, на странице «Языки» было расположено боковое меню (файл menu.php, который выполняется с помощью выражения include). Результатом вышеописанных действий является появление в меню страницы «Языки», при переходе на которую пользователь может ознакомиться с полным списком языков, содержащихся в выбранном разделе системы «СОЕЛ».

Возможность сохранения выборки на локальном хранилище может быть использована не только в исследовательских целях, но и в процессе изучения языка. Более того, в случае появления необходимости данная функция позволит без затруднений получить резервную копию информации, содержащейся в базе данных. Для реализации возможности выгрузки контекстов было принято решение использовать страницу «Контексты» и выпадающие списки, расположенные на ней. Программный код, отвечающий за экспорт данных, был расположен в файле export\_contexts.php. Процесс выгрузки контекстов по вы-

```

echo «<table border='1'>»;
  if ($_SESSION['database'] != 1) {
    while ($context1= $context[$i++]) {
      echo «<tr><td>». $crypt[0]['name'] . «</td>»;
      echo «<td>» . $classifier[0]['name'] . «</td>»;
      echo «<td>» . $noun[0]['name'] . «</td>»;
      echo «<td>» . $language[0]['short_name'] . «</td>»;
      echo «<td>» . $context1['context'] . «</td></tr>»;
    }
  }
echo «</table>»;

```

Рис. 20. Программный код, выполняющий построение таблицы и ее заполнение  
 [Fig. 20. The program code that performs the construction of the table and its filling]

бренным пользователем параметрам проходит в три этапа: 1) Получение идентификаторов выбранных параметров; 2) Построение таблицы на основе результатов SQL-запросов с полученными параметрами; 3) Выгрузка построенной таблицы в формате xls.

Для получения данных из выпадающих списков было принято решение использовать скрытые поля. Под данным термином понимается «особый элемент управления, не отображаемый Web-обозревателем» [7, с. 283]. Скрытое поле служит для передачи информации от страницы к странице (рис. 17). Для записи данных в поле используется скрипт (рис. 18), написанный на языке javascript с использованием библиотеки jQuery.

Данная функция выполняет передачу идентификаторов выбранных при помощи выпадающих списков параметров в скрытые поля. Таким образом, при нажатии кнопки «Выгрузить контексты» информация о выбранных параметрах передается в файл export\_contexts.php.

Идентификаторы криптокласса, классификатора, существительного и языка, переданные PHP-скрипту методом POST присваиваются соответствующим переменным crypt\_id, classifier\_id, noun\_id и language\_id. Далее полученные данные используются при выполнении SQL-запросов. На рис. 19 (рис. 19) изображены переменные, которым присвоены запросы к базе данных, выполняемые в разделах «GloWbE», «NOW» и «iWeb».

Рис. 20 (рис. 20) содержит программный код, выполняющий построение HTML-таблицы для данных разделов.

Содержимым таблицы является перебор результатов вышеупомянутых запросов к базе данных. В результате выполнения данного скрипта произойдет построение первой строки таблицы, а затем добавление новых строк, количество которых зависит от количества найденных в базе данных контекстов. За расширение таблицы отвечает операция постфиксного инкремента, которая «возвращает значение переменной, после которой записывается, а потом увеличивает ее значение на 1» [8, с. 36]. В данном случае увеличивается значение переменной context с помощью увеличения значения \$i (изначально равна нулю) на единицу (рис. 18). Используя кнопку «Выгрузить контексты», пользователь получит таблицу, содержащую контексты по выбранным параметрам. Данная таблица отображается на отдельной странице export\_contexts.php. Для записи и выгрузки таблицы формата Excel используется функция «header», выполняющая отправку HTTP-заголовка (рис. 21).

Заголовок «Content-Type» использован для определения типа файла и его кодировки. В качестве формата выгружаемого файла был выбран xls как наиболее поддерживаемый.

Заголовок «Content-Disposition» подставляет имя файла (в нашем случае значения первых записей полей «name» таблиц [pr]cryptoclasses, [pr]classifiers и [pr]nouns) и указывает тип контента как скачиваемый (параметр «attachment»). Для разделов «GloWbE», «NOW» и «iWeb» имя файла отличается наличием аббревиатуры языка. Таким образом, работая на странице «Контексты», пользователь может осуществлять выгрузку данных

```
header('Content-Type: applicatton/vnd.ms-excel; charset=utf-8' );
header('Content-Disposition: attachment; filename=»'. $crypt[0]['name']. '»'. $classifier[0]
['name']. '»'. $noun[0]['name']. '»'. xls»');
if ($_SESSION[ 'database' ] 1){
    header('Content-Disposition: attachment; filename=»'. $crypt[0]['name']. '»'.
.$classifier[0]['name']. '»'.
.$noun[0]['name']. '»'. $language[0]['short_name']. '»'. xls»');
}
```

Рис. 21. Функция «header»  
[Fig. 21. The header function]

на локальное хранилище с помощью выпадающих списков, используемых для получения контекстов в окне браузера. При нажатии кнопки «Выгрузить контексты», пользователь останется на текущей странице системы «СОЕЛ», а браузер отобразит диалог загрузки.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения действий с системой представления контекстов лингвистического ресурса «СОЕЛ», описанных в статье, были реализованы следующие основные шаги:

1. Страница «Контексты» новых разделов была дополнена выпадающим списком «Язык», используя который пользователь может фильтровать получаемую информацию по варианту английского языка. Функции отображения содержимого данного меню, а также меню «Существительное», были обновлены, в результате чего пункты указанных списков теперь изменяются в зависимости от примененных параметров. Для хранения информации, полученной на странице «Контексты», и работы с ней в офлайн-режиме была реализована возможность выгрузки на локальное хранилище.

2. Система выдачи контекстов была изменена в соответствии с особенностями новых данных: в новых разделах ресурса была реализована возможность фильтрации выводимого списка словоупотреблений по варианту английского языка, так как «iWeb» содержит 6 вариантов английского, а «GloWbE» и «NOW» — по 20. Также реализована система выгрузки контекстов для предоставления возможности работы с информацией на машине пользователя и локального хранения в

случае соответствующей необходимости, что может оказать пользу лингвистам в процессе проведения исследований или обучающимся при изучении английского языка.

## КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

## СПИСОК ЛИТЕРАТУРЫ

1. Дони́на, О. В. Проблема расширения функционала информационного лингвистического ресурса / О. В. Дони́на, В. В. Филатов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2019. – № 1. – С. 84-94
2. Борискина, О. О. Криптоклассы английского языка / О. О. Борискина. – Воронеж: издательство «Истоки», 2011. – 333 с.
3. Davies, M. Corpus of Contemporary American English (COCA) [Электронный ресурс] / M. Davies. – 2008. – Режим доступа: <https://corpus.byu.edu/COCA/> (дата обращения: 12.12.2018).
4. Davies, M. The corpus of Global Web-based English (GloWbE) [Электронный ресурс] / M. Davires. – 2013. – Режим доступа: <https://corpus.byu.edu/glowbe/> (дата обращения: 17.12.2018).
5. Davies, M. News On the Web (NOW) [Электронный ресурс]. – 2016. – Режим доступа: <https://corpus.byu.edu/now/> (дата обращения: 14.12.2018).
6. Davies, M. The iWeb corpus [Электронный ресурс]. – 2018. – Режим доступа:



<https://corpus.byu.edu/iweb/> (дата обращения: 29.12.2018).

7. Дронов, В. А. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. А. Дронов. – СПб.: БХВ-Петербург, 2007. – 480 с.

8. Бенкен, Е. С. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен. – Санкт-Петербург: БХВ-Петербург, 2008. – 352 с.

**Дони́на Ольга Валерьевна** – канд. филол. наук, преподаватель кафедры теоретической и прикладной лингвистики факультета романо-германской филологии Воронежского государственного университета.

E-mail: [olga-donina@mail.ru](mailto:olga-donina@mail.ru)

ORCID iD: <https://orcid.org/0000-0002-1053-540X>

DOI: <https://doi.org/10.17308/sait.2020.2/2923>

ISSN 1995-5499

Received 06.04.2020

Accepted 15.06.2020

## A SYSTEM FOR A CONTEXT PRESENTATION IN A LINGUISTIC RESOURCE

© 2020 O. V. Donina 

*Voronezh State University*

*1, Universitetskaya Square, 394018 Voronezh, Russian Federation*

**Annotation.** The article considers the second stage of updating the linguistic resource «COEL» (Cryptotypes of the English Language) developed by the Department of Theoretical and Applied Linguistics of Voronezh State University as a research tool applied in studies on the cooccurrence of abstract English nouns carried out within the framework of the cryptotype theory. The second stage involved the implementation of a system for context representation. The key tools used in the study were PHP, SQL, JavaScript, the HTML markup language, the CSS style sheet language, and the AJAX technique. The article considers the web interface of the information system, substantiates the need for modifications and details the introduced improvements. The following results were obtained: 1) the system for issuing contexts was adjusted to the specifics of the new data: in the new sections of the resource, it is possible to filter the displayed list of word usage by the dialects of the English language; 2) a system for downloading the contexts and local storage was implemented to enable users to work with information on their PCs, if necessary, which may be useful to linguists conducting their research or to students of English. These changes helped to adapt the system to the data obtained from such linguistic corpora as GloWbE (the corpus of Global Web-based English), NOW (News On the Web), and iWeb, as well as to make it more functional and convenient for users (either researchers or students of English).

**Keywords:** linguistic resource, PHP, SQL, JavaScript, the HTML markup language, the CSS style sheet language, the AJAX technique, corpus-based research, cryptotype analysis.

### CONFLICT OF INTEREST


The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

### REFERENCES

1. Donina O. V., Filatov V. V. The problem of expanding the functionality of the information linguistic resource // Bulletin of Voronezh State University. Series: System Analysis and Information Technology. 2019. No. 1. P. 84–94.

2. Boriskina O. O. Cryptoclasses of the English language. Voronezh : Istoki Publishing House, 2011. 333 p.

---

 Donina Olga V.  
e-mail: [olga-donina@mail.ru](mailto:olga-donina@mail.ru)



3. *Davies M.* Corpus of Contemporary American English (COCA) [Electronic resource]. 2008. Available at: <https://corpus.byu.edu/COCA/> (accessed: 12.12.2018).
4. *Davies M.* The corpus of Global Web-based English (GloWbE) [Electronic resource]. 2013. Available at: <https://corpus.byu.edu/glowbe/> (accessed: 17.12.2018).
5. *Davies M.* News On the Web (NOW) [Electronic resource]. 2016. Available at: <https://corpus.byu.edu/now/> (accessed: 14.12.2018).
6. *Davies M.* The iWeb corpus [Electronic resource]. 2018. Available at: <https://corpus.byu.edu/iweb/> (accessed: 29.12. 2018).
7. *Dronov V. A.* PHP, MySQL, and Dreamweaver. Development of interactive web-sites. St. Petersburg : BHV-Petersburg, 2007. 480 p.
8. *Benken E. S.* PHP, MySQL, XML: programming for the Internet. St. Petersburg : BHV-Petersburg, 2008. 352 p.

**Donina Olga V.** — PhD in Philology, Lecturer, Department of Theoretical and Applied Linguistics, Faculty of Romance and Germanic Philology, Voronezh State University.  
E-mail: [olga-donina@mail.ru](mailto:olga-donina@mail.ru)  
ORCID iD: <https://orcid.org/0000-0002-1053-540X>