УДК 003.053 : 621.377.037.3

Б. А. Аль-Нами
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Адаптация текстов противоположного направления (справа налево) в информационной продукции

Рассматриваются основные правила написания кода HTML для текстовых блоков фазового содержания или online, т. е. для блоков, в которых внутри абзаца смешиваются фразы текста с различными направлениями письма. Статья посвящена использованию разметки в HTML, но большинство концепций подходят и для других языков.

Информационная модель, адаптация, пользователи, особенности письменности и восприятия, интерфейс, интернет-браузеры

На данный момент при использовании на страницах сайтов или порталов существует проблема правильного отображения двунаправленного текста.

Если известно основное направление всего текста, то для корректного отображения фрагмента текста обратной направленности необходимо: плотно обернуть в разметке каждую фразу противоположного направления; добавить свойство в таблице стилей и использовать признак dir на томе разметки; обязательно отметить в гнезде разметку, чтобы показать структуру, например:

```
<p>название книги<span dir="rtl">مقدمة في
<span dir="ltr">C++</span></span>
на арабском языке.</p>
```

название книги C++ مقدمة في на арабском языке.

Чтобы создать надежный код для браузеров, не поддерживающих каскадные таблицы стилей (CSS), для плотно обернутого текста, который содержит действующее число или логически отдельную фразу противоположного направления, необходимо сразу после этой фразы добавить ‏ или &lrm, например { Курс "يتكون من " 10 уроков Звуковая и визуальная.} описывается следующим образом:

```
<p>Курс'<span dir='rtl' lang='he'>يتكون من
</span>'&lrm; 10 уроков
Звуковая и визуальная.</p>.
```

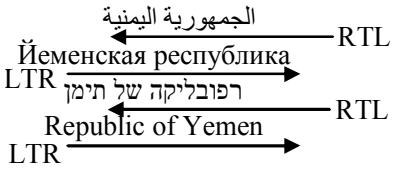
Если направление текста заранее неизвестно, то в предложения, которые будут вставлены во время работы, необходимо добавить dir=auto, для того чтобы фраза обратного направления была плотно обернута соответствующими тегами в любом случае. Если разметки нет, то необходимо местоположение обернуть тегом [1]:

```
<p><bdi>Кофе Мокка</bdi> - 5 reviews</p>
<p><bdi>الحرازي</bdi> - 4 reviews</p>
<p><bdi>يمني Шами</bdi> - 3 reviews</p>.
```

Двунаправленный алгоритм (биди-алгоритм). Важно запомнить, что в основных веб-браузерах физический порядок символов и визуальный не совпадают. Набор правил, применяемых в браузере для создания правильного порядка символов для отображения, описывается двунаправленным алгоритмом Unicode (биди-алгоритм или «bidi algorithm») [2], [3].

Порядок отображения текста зависит от базового направления фразы, абзаца или блока текста. Базовое направление – принципиально важное понятие, которое устанавливает направленность контекста. Биди-алгоритм применяется к различным точкам текста, чтобы решить, как обращаться с текстом. В HTML базовое направление должно либо явно указывать на ближайший родительский элемент, использующий dir-атрибут, или, если такой атрибут отсутствует, сохраняется от исходного направления документа, по умолчанию слева направо «LTR».

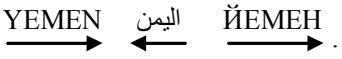
Представление символов смешанного текста. Исторически так сложилось, что в западноевропейской и славянской письменной традиции тексты читают и пишут слева направо. Биди-алгоритм позволяет расположить арабские или европейские символы в обратной последовательности: справа налево.



Как браузеру распознать, в какой последовательности расположены символы в тексте: слева направо или справа налево? В Unicode есть связанная направленная собственность для каждого направления. По умолчанию тексты представляются в формате LTR (слева направо), изредка тексты отображаются с обратной последовательностью RTL (справа налево). При этом символы RTL всегда отображаются справа налево, независимо от направления окружающего текста [4].

Направленные фразы. Когда в действующем тексте встречаются фразы с разным направлением символов, биди-алгоритм относится к отдельной фразе и запускает для каждой из них свою последовательность отображения смежных символов определенной направленности.

Например, существуют 3 разнонаправленные последовательности:



Обратите внимание, что применять разметку или стиль, чтобы так отображался текст, не нужно. Порядок направления символов на странице зависит от преобладающего базового направления.

В приведенном выше примере, у которого есть общее основное направление, слова читаются слева направо.

Если, например, изменить направление контекста в предыдущем примере, определяя, что направление HTML-элемента или исходного элемента, например «DIV» или «p», охватывает элемент «RTL», изменится направление всей фразы:

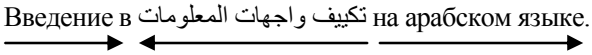
ЙЕМЕН اليمن YEMEN.

В обоих случаях в памяти сохранится первоначальный порядок символов, но визуальное направление знаков при отображении будет различаться [5], [6].

Нейтральные персонажи, пробелы и знаки пунктуации в Unicode не являются строго типизированными ни в LTR, ни в RTL, так как они могут быть использованы в любом типе сценария. Поэтому они классифицируются как нейтральные или слабые свойства текста. Когда биди-алгоритм сталкивается с подобными знаками, обладающими нейтральными свойствами, то их направление зависит от характеристики окружающих символов.

Символы с нейтральными признаками между двумя типизированными знаками с одинаковым типом направленности приобретают ту же направленность, что и контекст. Так пробел, обладающий нейтральными свойствами, приобретает свойство RTL между двумя символами RTL. Исходя из этого правила 3 арабских слова в следующем примере читаются справа налево как однонаправленная перспектива. Пробелы как нейтральные символы приняли то же направление, что и окружающие символы.

В следующем примере стрелки показывают порядок чтения:

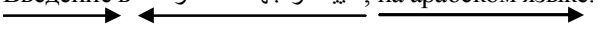


Таким же образом рассматриваются несколько нейтральных символов между двумя типизированными знаками.

Следует обратить внимание на то, что разметка или стилизация для данной операции все еще не нужна и что в этой фразе только 3 направления.

Если пробел или знак препинания попадает между двумя типизированными символами с различной направленностью (т. е. находится на границе разнонаправленных фраз), символы с нейтральными свойствами рассматриваются как обладающие одинаковой направленностью с преобладающим направлением.

Например, если в приведенном выше примере после последнего арабского слова добавить запятую, то она будет обладать свойством LTR (по направлению базового направления) и, следовательно, будет отображаться справа от текста на арабском языке.



К сожалению, данный метод не всегда работает корректно. Рассмотрим подобные случаи более подробно [7], [8].

Встраивание изменений в базовом направлении. Если название на арабском языке заканчивается восклицательным знаком, то вполне ожидаемо, что он должен появиться на левом краю арабского текста:

Введение в تكييف واجهات المعلومات!
на арабском языке.

Но, к сожалению, восклицательный знак, как и запятая, рассматривается как символ с нейтральными признаками и оказывается в том же месте, т. е. справа от арабской фразы.

Для исправления данной ситуации необходимо переопределить основное направление арабского текста и восклицательного знака.

Необходимо не только изменить основное направление текста, важное для обработки пунктуации на границе разнонаправленнных фраз, но и гарантировать правильный порядок направленных пробелов в двунаправленном тексте. Так, в следующем примере первая стрелка указывает ожидаемое направление, но вторая стрелка показывает отображение по умолчанию используя только биди-алгоритм двунаправленного текста:

Правильный перевод:".bdo تتحقق بإضافة العنصر المضمن XHMTL 1.0 في" «✓»

Правльный перевод:"في XHMTL 1.0 تتحقق بإضافة العنصر المضمن bdo." «✗»

«✓» – правильное отображение;

«✗» – неправильное отображение.

Проблема заключается в том, что в нижней строке, без определения базового направления, направляющие трассы внутри цитаты упорядочены слева направо. Чтобы решить эту проблему, нужно пересмотреть базовое направление для текста.

Примеры использования биди-алгоритма при статическом тексте. Рассмотрим написание кода, относящегося к случаям использования неизменяемого текста на странице.

Во всех примерах, связанных с использованием особенностей HTML, предполагается наличие в CSS надстройки для браузеров, которые поддерживают HTML, но не поддерживают изоляцию с элементом «dir».

Пример 1. «Название книги» направлено слева направо. Также само название книги содержит встроенную фразу справа налево.

Код без дополнительной биди-разметки:

 название книги «введение в C++»
на арабском языке.

Ожидаем увидеть:

название книги «C++ في مقدمة»
на арабском языке. «✓»

К сожалению, двунаправленный алгоритм не распознает, где должны быть границы вложенных изменений в основном направлении. В результате фраза выглядит так:

название книги «في مقدمة C++»
на арабском языке. «✗»

Чтобы решить эту проблему в HTML5, если нет никакой другой разметки вокруг словосочетания противоположного направления, нужно обернуть оба словосочетания так же, как в разметке с соответствующим значением «dir». Рассмотрим, как появляется разметка внутри кавычек, которые являются частью русского текста:

название книги "في مقدمة
C++"
на арабском языке.

Важно отметить, что каждая фраза обернута. Чтобы добиться правильного отображения, недостаточно просто обернуть фразу на арабском языке в один «span» с последующим обертыванием «span»-фразы, содержащей «C ++».

Замечания: в данном случае необходимо 2 элемента со свойством «dir», потому что в предложении существует две фразы противоположного направления. Если используется только одно направление, например:

 название книги "في مقدمة
C++".

то текст будет отображаться так, как показано ниже. Фраза «C++» перемещается влево, по мере необходимости, но «++» появляются на неправильной стороне «C»:

название книги "++Cفي مقدمة". «✗»

Это ошибка, потому что "C ++" обладает направлением LTR, в отличие от фразы в заголовке, которая заканчивается на нейтральных знаках и фразе, а теперь отображается с базового направления RTL. Биди-алгоритм не может распознать, что плюсы являются частью фразы LTR и входят в контекст RTL, а потому выводит их слева от «C», а не справа.

Чтобы решить эту проблему, необходимо обернуть полную фразу RTL в <SPAN DIR = = "RTL"> и фразу LTR, вложенную в ее собственном <SPAN DIR = "LTR">.

Если в исходном коде есть теги, подходящие для обертывания названия книги, такие, как cite-элементы, нужно добавить dir-атрибут в них:

 название книги <cite dir="rtl">في مقدمة
C++</cite>
на арабском языке.

название книги C++ في مقدمة
на арабском языке. «✓»

Символы порядка (LRM и RLM) используются для определения направления нейтральных знаков по отношению к основному направлению текста. Например, если двойные кавычки ставятся между арабскими (справа налево) и латинскими (слева направо) символами, то неясно, относятся ли кавычки к арабскому или к латинскому тексту? Символы «lrm» и «rlm» обладают свойством направления, но не имеют свойств ширины и разделения слов или строк [9].

Важно отметить, что при использовании LRM и RLM направленность фразы не учитывает нейтральные или слабые знаки в начале или в конце фразы противоположного направления, помещая отметку того же самого направления с основной фразой с другой стороны от тех нейтральных или слабых знаков. Например, вместо того, чтобы обернуть "C ++" в , можно добавить ‎ после второго плюса:

название книги <cite dir="rtl">مقدمة في C++‎</cite>.

В результате получится правильное отображение:

название книги C++ مقدمة في . «✓»

Пример 2. Фраза противоположного направления логически отделяется от контекста отдельным числом. HTML-код без биди-разметки:

Мы работаем 5"أيام" в неделю.

При правильном отображении должно получиться:

Мы работаем 5"أيام" в неделю. «✓»

Без использования биди-разметки получается:

Мы работаем "5"أيام в неделю. «✗»

Это происходит потому, что биди-алгоритм передает браузеру число «5» как часть арабского текста, но в данном случае такое отображение неверно. Необходимо найти способ, чтобы браузер различал текст и число, т. е. мог изолировать текст от числа.

Пример 3. Нейтральные символы между разнонаправленными текстами иногда неправильно распознаются по биди-алгоритму. Для примера, рассмотрим несколько названий сортов йеменского кофе на арабском языке со свойством LTR и следующую за ними логически отдельную фразу в противоположном направлении. Исходный код без биди-разметки выглядит так:

лучшие сорта йеменского кофе это аль-харази, аль-хайми и аль-матари, они являются самыми известными.

Верное отображение фразы:

лучшие сорта йеменского кофе это الحرازي, الحيمي и المطري, они являются самыми известными. «✓»

Без биди-разметки полностью изменены первые 2 арабских слова, и пробел расположен справа от них:

лучшие сорта йеменского кофе это الحرازي, الحيمي и المطري, они являются самыми известными. «✗»

Это произошло из-за того, что внутри строго типизированных символов со свойством справа налево (RTL) с обеих сторон биди-алгоритм распознает нейтральную запятую между (الحرازي и الحيمي) как часть арабского текста. В связи с этим он интерпретирует первые 2 арабских слова и запятую как однонаправленный текст на арабском языке, хотя на самом деле это часть текста на кириллице и алгоритм должен отметить границу между двумя отдельными фразами: кириллицей и арабской [10].

Чтобы решить эту проблему, необходимо обернуть каждое арабское слово с разметкой и добавить соответствующее значение «dir»:

лучшие сорта йеменского кофе это الحرازي, الحيمي and المطري они являются самыми известными.

Если разметка окружающего арабского текста уже есть, то нужно добавить атрибут «dir» к ней:

лучшие сорта йеменского кофе это الحرازي, الحيمي and المطري они являются самыми известными.

В браузерах, которые не поддерживают функции HTML5, необходимо добавить не только разметку вокруг арабского текста, но и свойство «LRM» после нее каждый раз, когда данный текст сопровождается фразой противоположного направления. Необходимо использовать свойство «RLM», если окружающий текст направлен справа налево:

 лучшие сорта йеменского кофе это الحرازي‎, الحيمي and المطري они являются самыми известными.

Необходимо отметить, что если арабский текст плотно обернут каким-либо тегом, то можно использовать данный тег, чтобы добавить атрибут «dir».

Примеры использования биди-алгоритма при динамической вставке текста. Рассмотрим написание кода, относящегося к случаям использования динамически изменяемого текста на странице.

Важно помнить, что изменять разметку внутри вводимого содержимого нельзя. Во всех случаях, описанных далее, вставленные фразы, если они содержат в себе текст противоположного направления, должны быть уже размечены при выводе на страницу. Если это не будет сделано, то вводимый текст будет правильно представлен только в простом случае, в сложных же случаях могут появиться проблемы.

Пример 4. Предположим, что пользователь ищет книгу под названием «CSS مقدمة في لغة» в книжном интернет-магазине, но не нашел ее и получил сообщение о том, что книга не найдена.

Ваш поиск CSS *مقدمة في لغة* не найдено
ни одного документа. «✓»

При этом «CSS» расположено слева от арабского текста, потому что является частью названия книги. Чтобы изменить результат, необходимо применить следующий исходный код:

Ваш поиск - <cite class="booktitle">CSS مقدمة في لغة</cite>- не найдено ни одного документа.

В этом случае изображение выглядит иначе: фраза «CSS» теперь находится справа от арабского текста:

Ваш поиск *مقدمة في لغة* CSS не найдено
ни одного документа. «✗»

Обязательное правило: когда вокруг введенного текста нет другого тега, необходимо обернуть его в «bdi»:

Ваш поиск ,<bdi> CSS مقدمة في لغة </bdi>,
не найдено ни одного документа.

Тег «bdi» автоматически присваивает всей фразе направление на основе первого сильно типизированного символа строки [4].

Заметим, что в данном примере строка поиска начинается слева направо. Например, если заглавие искомой книги начинается с «CSS», а не заканчивается им, то в таком случае нельзя сильно изменить разметку. Для решения этой проблемы необходимо использовать сценарии, чтобы определить направление строки в целом.

Если вокруг введенного текста есть еще один тег, необходимо использовать тег «dir="auto"» или обернуть вводимую фразу в «bdi»:

Ваш поиск ,<cite dir="auto">CSS مقدمة في لغة</cite>, не найдено ни одного документа.

Ваш поиск ,<cite><bdi>CSS مقدمة في لغة</bdi></cite>, не найдено ни одного документа.

Ваш поиск , CSS *مقدمة في لغة*, не найдено
ни одного документа.

В HTML4 нельзя решить данную проблему с помощью разметки, так как необходимо заранее знать направление текста. Решение может быть достигнуто, если направление текста известно или проведен анализ вводимой фразы перед вставкой.

Пример 5. Названия сортов йеменского кофе вставляются в страницу из базы данных и сопровождаются числом. При этом направленность введенного текста заранее не известна:

 Кофе Мокка
- 5 reviews

 Аль-Харази
- 4 reviews

 Йеменский
Шами - 3 reviews

Результат выполнения скрипта представлен ниже.

Как должно выглядеть:
Кофе Мокка – 5 reviews
الحرازي – 4 reviews
Шами **يمني** – 3 reviews
Как выглядит на самом деле:

Кофе Мокка – 5 reviews
4 – **الحرازي** reviews
يمني Шами – 3 reviews

Проблема возникает в средней части названия йеменского кофе «аль-харази». Браузер считает, что «–4» – часть арабского текста. Такое отображение объясняется работой биди-алгоритма Unicode. В большинстве случаев он отрабатывает корректно, но не в данном случае. Необходимо найти способ, чтобы разделять название от числа.

В третьей строке число «–3» располагается в нужном месте, но слово «Шами», являясь частью арабского названия, должно располагаться слева от арабского текста. Другими словами, необходимо применить основное направление RTL ко всему введенному тексту.

Важно отметить, что, когда вокруг введенного текста нет других тегов, достаточно добавить в него «bdi»-свойство, и оно автоматически изолирует введенную фразу от числа, а также задаст направление для фразы по ее первому сильному символу:

```
foreach $йеменского кофе
echo "<p><bdi>$йеменского кофе ['name']</bdi> –
$йеменского кофе ['count'] reviews</p>";
```

Заметим, что в приведенном примере свойство «bdi» ставит разметку вокруг названия «Кофе Мокка». Это упрощает необходимый код скрипта.

При наличии вокруг введенного текста какого-либо тега необходимо обернуть введенную фразу в «bdi» или использовать «dir="auto"»:

```
foreach $йеменского кофе
echo "<p><a href='...'
class='name'><bdi>$йеменского кофе
['name']</bdi></a> – $йеменского кофе ['count']
reviews</p>";
foreach $йеменского кофе
echo "<p><a href='...' dir='auto'
class='name'>$йеменского кофе ['name']</a> –
$йеменского кофе ['count'] reviews</p>";
```

Кофе Мокка – 5 reviews
الحرازي – 4 reviews
Шами يمني – 3 reviews

Пример 6. Пунктуация в конце фразы противоположного направления.

Расположение в конце фразы противоположного направления знака пунктуации, принадлежащего этой фразе, – достаточно распространенная ситуация. К сожалению, когда нейтральные символы находятся между разнонаправленными фразами, они, как правило, неправильно представляются.

В следующем примере восклицательный знак должен располагаться в конце арабского текста, т. е. слева от него:

Введение в تكييف واجهات المعلومات!
на арабском языке. «✓»

Однако если использовать только бидиалгоритм, то в результате получится:

Введение в تكييف واجهات المعلومات!
на арабском языке. «✗»

Поскольку восклицательный знак был напечатан между последним символом RTL 'ت' (слева) и знаком «н» LTR (от предлога «на»), его направленность определяется базовой направленностью фразы, в данном случае LTR. Так как восклицательный знак рассматривается как LTR, то он присоединяется к фразе, которая включает в себя арабский текст [3], [10].

Для решения данной проблемы достаточно поставить во встречных направлениях фразы тег с «dir»-атрибутом. Если дополнительного тега нет, необходимо использовать «span»:

```
<p>Введение в"<cite dir="rtl" lang="ar">تكييف
واجهات المعلومات!</cite>" на арабском языке.</p>
```

Введение в "*تكييف واجهات المعلومات!*"
на арабском языке. «✓»

В данном случае верным решением является использование тега «bdi» при отсутствии дополнительного тега, в противном случае необходимо использовать «dir="auto"»:

```
<p>Введение в "<bdi lang="ar">تكييف واجهات
المعلومات!</bdi>" на арабском языке.</p>
<p>Введение в "<cite dir="auto" lang="ar">تكييف
واجهات المعلومات!</cite>" на арабском языке.</p>
<p>Введение в "<cite lang="ar"><bdi
dir="rtl">تكييف واجهات المعلومات!</bdi></cite>"
на арабском языке.</p>
```

Таким образом, локализация гораздо больше, чем просто перевод с одного языка на другой. При локализации в двунаправленной языке необходимо настроить пользовательский интерфейс, чтобы сделать его пригодным для использования. Предложенной формой интерфейса могут пользоваться не только страны Арабского мира, но и другие страны, составляющие 42 % населения земного шара.

СПИСОК ЛИТЕРАТУРЫ

1. Ходаков В. Е. Пользовательский адаптивный интерфейс: задачи исследования и построения // Восточно-Европейский журн. передовых технологий. 2004. № 2. С. 20–29.

2. Findlater L. Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces // AI Magazine. 2009. № 30 (4). P. 68–73.

3. Назаренко Н. А., Падерно П. И., Аль-Нами Б. А. Автоматизация процедуры оценки качества пользовательских интерфейсов // Седьмой междунар. аэрокосмический конгресс IAC-12: тез. докл. М., 2012. С. 310–311.

4. Moukas A. Amalthaea: An evolving multi-agent information filtering and discovery system for the www // Autonomous Agents and Multi-Agent Systems. 1998. № 1. P. 59–88.

5. Тидвелл Д., Разработка пользовательских интерфейсов. СПб.: Питер, 2008.

6. Денинг В., Эссиг Г., Маас С. Диалоговая система «человек-ЭВМ». Адаптация к требованиям пользователя. М.: Мир, 1984.

7. Деревицкий Д. П., Фрадков А. Л. Прикладная теория дискретных адаптивных систем управления. М.: Наука, 1981.

8. Адаптивные системы автоматического управления / под ред. В. Б. Яковлева. Л.: Изд-во Ленингр. ун-та, 1984.

9. Trapeznikov S., Dinenberg F., Kuchin S. InterBase: A Natural Language Interface system for popular commercial DBMSs // Proc. of the EAST-WEST conf. on artificial intelligence, Moscow, 1993. P. 189–193.

10. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем. СПб.: Символ-Плюс, 2005.

B. A. Al-Nami
Saint-Petersburg state electrotechnical university «LETI»

ADAPTATION OF TEXTS THE OPPOSITE DIRECTION (RIGHT TO LEFT) IN INFORMATION PRODUCTS

This article discusses the basic rules of writing the HTML code for text blocks phase content or line, that means blocks which are mixed within a paragraph of text phrases with different directions of the letter. Article is devoted to the use of markup in HTML, but most of the concepts can also be used for other languages.

Information model, adaptation, users, especially writing and perceptions, interfaces, internet-browsers

УДК 20.53.19, 28.23.13

А. В. Смирнов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Мутационное тестирование программного обеспечения

Рассматриваются преимущества и недостатки набирающего в последнее время популярность метода тестирования программного обеспечения – мутационного тестирования и возможность его применения в качестве одной из основных характеристик при оценке качества программного обеспечения.

Тестирование, разработка через тестирование, программное обеспечение, мутационное тестирование, покрытие кода, качество программного обеспечения

Многие современные методики разработки программного обеспечения (ПО) не только уделяют большое внимание автоматическому тестированию, но и основываются на нем.

Главным примером сложившейся ситуации является ответвившаяся от «экстремального» программирования *разработка через тестирование* (англ. test-driven development, TDD) – техника, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

Одной из основных метрик, которые отражают качество созданного программного обеспечения, является *покрытие кода* (англ. code coverage), показывающее процент тестирования исходного кода программы.

Однако сам по себе даже высокий процент покрытия кода тестами не может свидетельствовать о качестве кода, если не существует метода, позволяющего оценить достоверность и качество самих тестов. Один из таких методов и описывается в данной статье.

Мутационное тестирование (мутационный анализ) – метод тестирования программного обеспечения, который включает небольшие изменения