ПРЕДСТАВЛЕНИЕ И ИНТЕРПРЕТАЦИЯ ТЕКСТА В ТЕХНОЛОГИИ СМЕШАННОГО НАБОРА

TEXT REPRESENTATION AND INTERPRETATION IN A «MIXED SCRIPTS TECHNIQUE»

А. В. Коваленин

Издательство «Книжица», Новосибирск
kovalenin@mail.ru

Для создания корпуса старинных певческих рукописей потребовалось найти способ работы с текстами, содержащими знаки крюкового письма, нот и церковнославянского языка. Найденный подход оказался применимым и для других письменностей, а его ключевой автомат — для некоторых задач прикладной лингвистики.

1. Создание электронного фонда старинных певческих рукописей («Фонда знаменных песнопений»[1]) стало возможным благодаря их набору в формате создаваемой «Технологии смешанного набора», поскольку этот формат позволяет:

– разделить содержание текста и детали для его интерпретации;

– отразить в тексте разные взгляды на то, что является его содержанием, то есть позволить каждой группе исследователей помещать в единицу хранения информацию, не нужную другим;

– использовать широкое понимание интерпретации, включающее многоступенчатые преобразования самого текста, опирающиеся на контекстно-обусловленные описания специального вида.

Последнее свойство не мешает Технологии смешанного набора быть открытой для подключения к процессу интерпретации любых внешних преобразователей. Однако пока удалось обойтись одним автоматом, описав необходимые для нашей практики преобразования в нужном ему виде, что позволило реализовать компактную рабочую версию инструментария (70 К в zip-архиве).

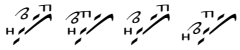
2. Недостатки непосредственного набора

Технология смешанного набора первоначально создавалась для работы с текстами древнерусских певческих рукописей, после того как стандартный подход (набор непосредственно шрифтами в привычном текстовом процессоре) оказался неудобным. Это неудобство объективно, сложность крюкового письма только помогла осознать необходимость альтернативы. Перечислю эти трудности.

1. *При наборе.* а). Знак в крюковом письме («знамя») — составной объект, в котором на основной элемент накладываются до 5 дополнительных. Реализация в шрифте всех комбинаций отдельными символами громоздка и затрудняет выбор символа, а реализация накладных элементов символами, не сдвигающими курсор, затрудняет редактирование. Последняя трудность становится критической, так как некоторые элементы традиционно отображаются другим цветом.

б). Многие накладные элементы нужно иметь в шрифте в нескольких вариантах. Это видно уже на простом тексте с ударениями — например, ударение, хорошо стоящее над ѡ, плохо встаёт над а. В крюковом письме помета свободно ставится рядом со знаменем; в нашем шрифте, например, помета т сочетается с крюком восемью способами:

Когда накладных элементов несколько, подбор взаимно подходящих сочетаний становится головоломным. Например, знамя *стрела мрачная борзая* с двумя пометами н и т может набираться разными способами, не все из которых удачны:

 (1).

Возникает парадокс — чем лучше (с точки зрения полиграфической гибкости) продуман шрифт, тем труднее с ним работать.

в). Работа с разными шрифтами требует овладения их кодировками, различие которых даже для одной письменности неизбежно. Практически это означает, что интересные во многих отношениях шрифты остаются невостребованными.

2. *Работа с набранным текстом* попадает в зависимость от кодировки выбранных шрифтов. Сменить шрифт на шрифт другой кодировки невозможно. При использовании нескольких шрифтов даже простой контекстный поиск оказывается затруднён. При использовании в тексте нескольких письменностей все эти трудности умножаются, и добавляются проблемы смены раскладки клавиатуры.

Причина всех этих сложностей — концептуальное смешение процессов ввода, хранения и отображения информации, нарушение принципа «отдельные вещи делаются отдельно». Текст, по сути, неизбежно набирается в расчете на определенный внешний вид, хотя исследователю это может быть вовсе не нужно или нужно в другом виде.

[1] Фонд знаменных песнопений — http://znamen.ru.

В итоге, хотя наш первый опыт крюкового набора был также связан с разработкой специального шрифта, мы теперь не набираем и не рекомендуем другим набирать непосредственно шрифтом.

3. Примеры набора

Далее *текстом* будем называть просто последовательность знаков, предназначенную для последующей интерпретации читающим (человеком или автоматом). Примеры показывают возможный результат наиболее распространенного вида интерпретации текста — получения внешнего вида.

В примере 1 используются четыре разных письменности — знамена, нотно-линейное письмо, церковнославянские буквы, современные русские буквы; в примере 2 — буквы старославянского, русского (до 1918 г.), латинского и греческого алфавитов.

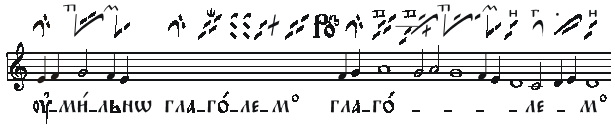
Пример 1 (фрагмент крюковой азбуки):

7. ѿ Ча́шка  ѿ по́лнаѧ  . Чашка полная встречается только в восьмом гласе.

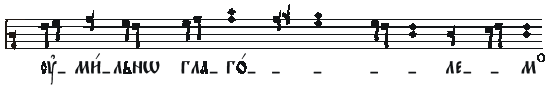
Пример 2 (фрагмент словаря Срезневского):

въıль — *трава, herba:* — ꙁемѧ… раждажшти **въıль** влагж. *Панд. Антт. XI в. 49.* **свєрѣна** во **єсть** похоть **акы** дикам **выль** о **совѣ** **въꙁникши** на **недѣланѣи** ꙁемли. *Іак. Посл. Дм. п. 1078 г.* — *Ср.* **въıти**, *Гр.* φύλλον, *Лат. folium.*

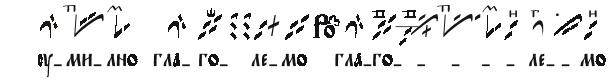
Пример 3 (для современного издания):



Пример 4 (как синодальное издание):



Пример 5 (приближенно к оригиналу):



Примеры 3–5 — результаты различной интерпретации одного и того же *текста,* набранного из тропаря празднику Покрова. Можно видеть, что по выбору пользователя:

а) текст песнопения представлен шрифтами *Irmologion Ucs* (пример 3) или *knijitsa* (4, 5), имеющими разные кодировки;

б) оставлена крюковая запись мелодии (5), нотная (4) или обе (3), во втором случае удалён и потерявший значение фрагмент текста;

в) формат нотной записи выбран современный (3) или старый (4);

г–д) раздельноречие и ударения показаны (3, 4) или нет (5);

е) подтекстовка дана в орфографии оригинала (5) или в нормативной орфографии «позднего извода» (3, 4).

4. Технология смешанного набора

§1. Текст набирается в простейшем текстовом редакторе; фрагменты каждой письменности изображаются посредством разработанного для неё представления, основанного на набираемых с клавиатуры символах. Например, в церковнославянском тексте буква ѧ изображается «`я'`», буква ѩ — «`ja=`», ѣ — «`jь`»; в крюковом письме упомянутая стрела борзая (1) изображается «`нпб_./`». Такое представление обычно легко читается, не заставляя исследователя разбираться в шрифтах и их поддержке. Для церковнославянского языка оно сделано на основе стандарта HIP [2], для нот и крюков разработано самостоятельно.

§2. Каждый фрагмент, требующий особой интерпретации, помечается соответствующей этой особенности меткой. Метка — это только сигнал о наличии в *её фрагменте* особого содержания, а конкретизация этой особенности — способ интерпретации этого фрагмента задается отдельно в *стилевом файле*. Пример 1 — результат интерпретации следующего текста (подчёркнуты метки):

```
%ном 7. <з> сЧ <род>Ча'шка <н>с1н1 <з>
стЧ. <т>по'лная <н>с2н1г1 %объ Чашка
полная встречается только в восьмом гласе.
```

То есть само понятие метки при этом не отличается от привычного понятия *стиля* (*tag*) известных текстовых процессоров. Отличие состоит в двух особенностях интерпретации меток (§§ 3, 4).

§3. Интерпретация по умолчанию (если она не задана в стилевом файле) — не игнорирование метки, как в HTML, а уничтожение следующего за ней фрагмента. Это позволяет вносить в текст любое количество независимых друг от друга разметок. Например, можно перед любой имеющейся в тексте меткой добавить свой фрагмент: «`%кстати, к этому месту надо ещё вернуться`». Если в стилевых файлах пользователей метка «`%кстати`» не предусмотрена, для них этот фрагмент будет несуществующим. Так, исключение нот или крюков в примерах 4 и 5 произведены просто удалением из стилевого файла описания интерпретации соответствующих меток.

Таким образом, в тексте может содержаться много слоёв информации, соответствующей различным аспектам исследования. Стилевой файл становится держателем *взгляда на текст* — представления о том, что в нём для пользователя значимо, а что будет игнорироваться.

Поскольку не возникает необходимости ограничить сосуществование различных разметок их взаимной вложенностью, постольку эти взгляды могут быть совершенно независимы.

§4. Интерпретация фрагмента состоит в задании необходимых *преобразований* помеченного фрагмента. Если, например, предполагается распечатка текста, то необходимы преобразователи пред-

[2] Стандарт HIP, разработка М.Гринчука. – http://orthlib.ru/hip/

ставлений используемых в нём письменностей в кодировку какого-нибудь шрифта. В инструменте Фонда знаменных песнопений используются и такие преобразования, как устранение надстрочных знаков, выбор варианта орфографии (между оригиналом и нормой), преобразование арабских чисел в славянские, применение буквицы, выключение двухцветности крюкового фрагмента.

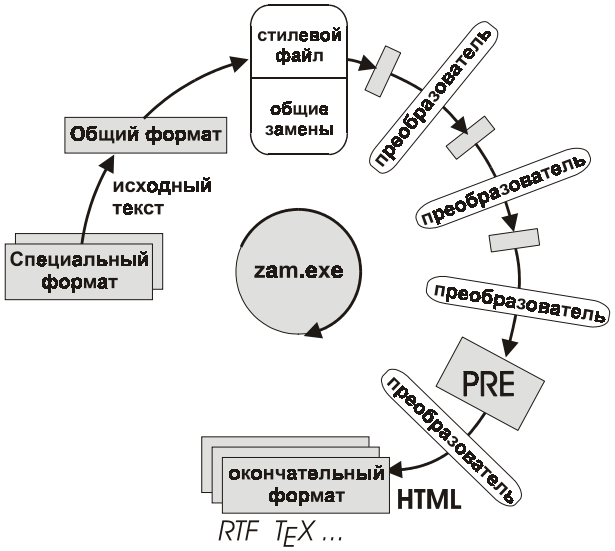


Рис. 1. Схема интерпретации текста

Интерпретация текста состоит в следующем: согласно стилевому файлу метки заменяются символьными цепочками, содержащими сигналы для включения других преобразователей. Преобразователи прочитывают текст и при обнаружении «своего сигнала» обрабатывают начинаемый им фрагмент. В результате этот фрагмент или остаётся в формате представления своей письменности (той же или другой), или превращается в цепочку кодов назначенного шрифта и управляющих символов выходного промежуточного языка. Если интерпретация производится ради получения внешнего вида, то производится преобразование из этого языка в выходной формат. Для действующей системы крюкового набора в качестве выходного формата пока реализован только HTML

5. Заменитель

Рассмотрим устройство основного автомата, реализующего цепочку имеющихся преобразований — как системных (начиная со стилевого файла, и кончая получением окончательного вида), так и перечисленных прикладных.

Объяснение его работы удобно начать с такой подзадачи морфологического разбора, как **выделение приставок**. Задача была решена автором в 1988 году в ВЦ СО АН СССР [1] в полной постановке. Здесь упрощенно рассмотрим её как задачу вставки в слово символа морфемной границы «*». Анализ словаря показал, что для определения, например, приставки «анти» не обязательно держать в памяти все словоформы на «анти-», а достаточно проверить три возможных продолжения слова. Это можно записать в виде четырёх строк:

```
|анти|анти*|
|античн|античн|
|антиквар|антиквар|
|антикварк|анти*кварк|          (2)
```

Каждая из них представляет собой **замену**, так как свою левую часть («*заменяемое*») предлагает заменить своей правой. Наш автомат («*заменитель*»), пробегая по тексту, сравнивает продолжение текста от текущей позиции на совпадение с левыми частями замен, и выполняет ту замену, для которой это совпадение наибольшее.

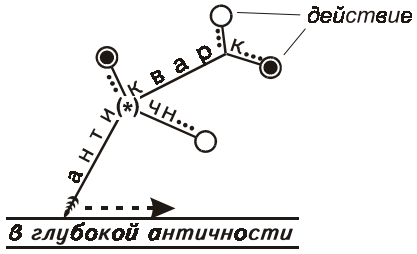


Рис. 2. Дерево замен в поиске места своего применения к тексту

На рисунке 2 *заменяемые* собраны в древовидную структуру, а возможные вставки на их место правых частей обозначены обобщающим словом «действие».

Эффективность этой операции зависит не от количества заменяемых (объёма полезной части словаря), а от числа ветвлений дерева на пути к «наиболее подходящей» замене, то есть при самом развесистом дереве ограничивается величиной, пропорциональной длине заменяемых. Это выгодно отличает такой механизм замен от привычного последовательного применения каждой отдельной замены ко всему тексту.

Вторым, решающим шагом в развитии заменителя явилось небольшое усложнение действия, выполняемого после замены — введение *сдвига*. Рассмотренной системе замен эквивалентна такая:

```
|анти|анти(*)|-3
|(*)|*|
|(*)чн|чн|
|(*)квар|квар|
|(*)кварк|*кварк|          (3)
```

Число **-3** (*«сдвиг»*) указывает, что после выполнения первой замены (если заменитель находился в позиции перед «анти») автомат сдвинется на три позиции назад, то есть встанет не после появившегося в тексте «(*)», как если бы сдвига не было, а перед ним. Можно сказать, что автомат находится в состоянии «разбираем приставку *анти*», в котором уже будут учитываться только необходимые в этой ситуации замены.

Более простое применение сдвига — замена «|'20'20|'20|-1», заменяющая любое количество стоящих подряд пробелов ('20) на один пробел.

Введение сдвига дало возможность такими заменами описывать сложные преобразования с движением по тексту не только вперёд, но и назад.

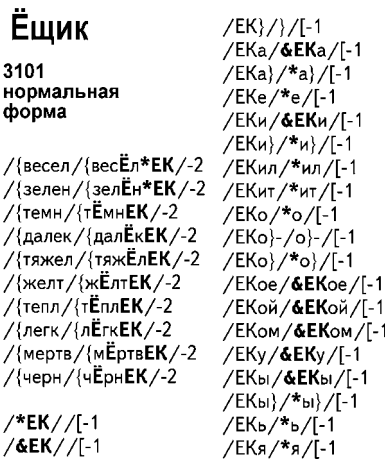


Рис.3. Фрагмент системы замен для расстановки точек над «е». Фигурные скобки обозначают начало и конец слова, а «`[-1`» — переход после замены назад, на одну позицию раньше прежнего положения автомата.

В качестве примера можно рассмотреть систему замен для расстановки в тексте «ё». Целый ряд основ имеет общую, так сказать, «ё-парадигму» — набор продолжений слова, говорящих о наличии или отсутствии в этой основе «ё». Согласно заменам на рисунке 3 слева, заменитель вставляет в эти основы временную (прописную) **Ё** и сочетание **ЕК**, с которым он затем работает согласно заменам справа. Эти замены, в свою очередь, вставляют в слово знак «**&**», подтверждающий наличие «ё», или знак «*****», отрицающий его. После этого ещё одна группа замен перемещает этот знак назад к ранее вставленной **Ё,** чтобы сделать окончательный выбор.

6. Задачи о заменителе

Поиск по словарю эффективно реализуется через его представление в виде изображенного на рисунке 1 дерева. Тем самым словарь с уже расставленными нужными признаками (морфемными границами, диакритическими знаками, кодами грамматических значений…) даёт тривиальное решение задачи их расстановки. Но, как видно из систем замен (2), на 118 словарных слов [2] достаточно иметь только 4 замены, и это даёт не только сокращение перебора, но и решение задачи для большого множества несловарных слов. В то же время, на некоторых из них это решение может быть неправильным, что оставляет актуальным поддержание эталонного словаря и ставит задачу *автоматического построения минимально достаточного дерева замен*.

Например, для корпуса текстов на церковнославянском языке[3] интернет-сообществом «Славянская типографика»[4] созданы словники, которые пополняются по мере пополнения корпуса. Решение этой задачи позволило бы справиться с проблемой расстановки надстрочных знаков. При этом проблема разноударяемых слов решалась бы так же, только вместо перечня слов, использовался бы перечень более длинных контекстов, составление которого по готовому корпусу текстов не представляет технической сложности.

Вторая задача — обнаружение в построенном дереве одинаковых поддеревьев и перестройка дерева, аналогичная переходу от (2) к (3). Вставляемую при этом метку «(*)» можно считать идентификатором некоторого словоизменительного класса: основы для изменения выписаны в строчках, ею заканчиваемых, а варианты изменения — в строчках, ею начинаемых.

За этими классами могут обнаруживаться реальные языковые явления. Но поскольку, как правило, конкретные основы представлены в словнике неполным набором возможных продолжений, интересно рассмотреть *более сложный вариант второй задачи* — когда для обнаруживаемых поддеревьев требуется не полное совпадение, но вложение друг в друга, а ещё лучше — когда поддерево, в которое могут вкладываться другие, реконструируется (восстановление полной парадигмы по неполным). Постановка такой задачи ещё нуждается в уточнении.

7. Добавление новой письменности

Показанный в системе замен (3) приём введения в текст «метки ситуации» позволяет так организовать систему замен, чтобы она работала только с фрагментами текста, начинающимися особым сигналом. Так создаются преобразователи для отдельных письменностей.

Допустим, мы хотим набрать подтекстовку песнопения японской слоговой азбукой хираганой. Покажем, как включить поддержку этой азбуки в систему. Сам японский текст напишем в удобной для нас транслитерации, предварив меткой-стилем:

```
<подтекст>шу авареме ё
```

В стилевом файле этой метке сопоставим выбор шрифта и сигнал для преобразователя хираганы:

```
|<подтекст>|<кг 16><гарн "Code2000">
                    <::япон hiragana-uni>|
```

Для составления преобразователя (системы замен) воспользуемся таблицей кодов хираганы в стандарте Unicode как показано на рисунке 4.

8. Состояние и перспективы проекта

Изложенная технология реализована в качестве рабочего инструмента для набора Фонда знаменных песнопений. Разработаны форматы для представления церковнославянских текстов, нот, знамен столпового и демественного распевов, реализованы преобразователи из них в отдельные шрифты, энтузиастами ведётся набор единиц хранения фонда.

[3] «Библиотека святоотеческой литературы». — http://orthlib.ru
[4] «Славянская типографика». —http://fonts.improvement.ru/

Для этого был сформулирован специальный формат песнопения (рис.1), свободный от оформительских деталей.

В качестве заменителя в инструменте Фонда знаменных песнопений используется программа zam.exe, придуманная автором и написанная в 1994 году А.Такмаковым в виде DOS-приложения. Это обстоятельство позволяет пока говорить только о «скелетной» версии инструмента, в которой средства набора и интерпретации разделены.

Технология смешанного набора позволяет преодолеть эту разделённость и даже вернуться к удобствам режима немедленной визуализации набираемого (wysiwyg), но без описанных в п.2 недостатков. Для этого структура текста внутри текстового процессора может оставаться в формате этой технологии, а сама технология — использоваться для визуализации текста на экране. При этом отражение движения по тексту курсора будет производиться через описанный механизм замен.

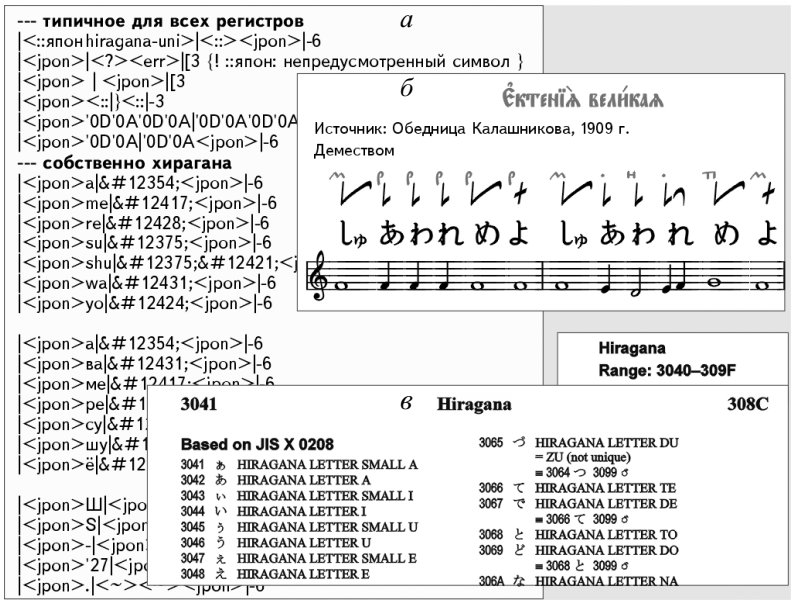


Рис.4. Подключение японского языка.
а — фрагмент преобразователя русской транслитерации японского текста в знаки хираганы (числа после «&#» — десятичные коды символов согласно Unicode); б — внешний вид песнопения с крюками, нотами и японской подтекстовкой; в — таблица хираганы в стандарте Unicode.

СПИСОК ЛИТЕРАТУРЫ

1. Коваленин А.В., Несговорова Г.П.. Несловарное описание исключений и автоматизация переноса слов // Третья всесоюзная конференция по созданию Машинного фонда русского языка. Тезисы докладов. Москва, 1989. Ч. I, С.164–166.
2. Зализняк А.А. Грамматический словарь русского языка. М.: Русский язык, 1977.