Идеология и техника разметки в Технологии смешанного набора

А. В. Коваленин

Институт систем информатики имени А.П.Ершова СО РАН Новосибирск, Россия

Технология смешанного набора (ТСН) есть способ работы с текстами, содержащими фрагменты различных письменных систем. С технической точки зрения можно говорить просто о фрагментах, требующих разной интерпретации, что сразу делает формулировку проблемы более общей и актуальной и для текстов, однородных по используемой письменности. Для работы с такими текстами требуется решение трёх взаимосвязанных задач: 1) создание единой платформы для представления в одном файле разного рода фрагментов; 2) нахождение способа разбиения текста на фрагменты; 3) предоставление пользователю способа описания интерпретации фрагмента.

Цель данной работы – сосредоточиться на вопросах разметки и её интерпретации, рассмотрев их общие основания, и показать возможности для повседневной работы исследователя принятого в ТСН подхода в сравнении с другими имеющимися подходами к разметке, прежде всего XML.

1. *Термины. Текст* мы понимаем как зафиксированную последовательность знаков. Текст подвергается *интерпретации* человеком или автоматом с конкретной практической или исследовательской целью. Каждое исследование опирается на информацию, связанную с конкретными местами в тексте, и всякая смена способа интерпретации нуждается в указании места такой смены. Совокупность указаний на места в тексте, связанных с конкретной интерпретацией, мы называем *разметкой*. Таким образом, с текстом может быть связано много разметок, отвечающих разным целям.

Важными примерами разметки являются *разбиение* текста на составляющие его фрагменты и *отражение* структуры текста в виде *иерархии*. Связь иерархии с текстом обусловлена взглядом на текст (целью интерпретации), так что разные разметки одного текста могут поддерживать разные иерархии.

Но в общем случае разметка – это не разбиение и не иерархия, а только набор указаний на места в тексте.

По способу связи с текстом разметки делятся на два класса: внешние и внутренние.

Внешняя разметка – это набор ссылок на элементы текста, хранимый отдельно от текста. Такая разметка наилучшим образом соответствует независимости объекта (текста) от его описания, обеспечивая независимость разметок. Примером внешней разметки может быть перечень мест в последовательности знаков, выраженный номерами слов текста, букв слова.

Внутренняя разметка – такой способ разметки, при котором место в тексте отмечается внесением прямо в последовательность знаков текста специальных *меток* – элементов языка разметки. При этом текст уже не является неизменным объектом, а изменяется с добавлением каждой новой разметки. Вместе с внутренней разметкой удобно рассматривать и *неявную разметку* – видимые и невидимые элементы исходного текста, которые тоже могут служить исследователям опорой при интерпретации.

Внешняя разметка имеет два неудобства: 1) для её редактирования необходимо специальное программное обеспечение (СПО), 2) редактирование самого текста влияет на все имеющиеся разметки, в том числе созданные независимыми исследователями и вне СПО. Тем не менее, внешняя разметка используется в практике, так как её важным достоинством являются поддержка множества независимых разметок и неизменность (защищённость) самих текстов корпуса. Известен и подход, при котором внешняя и внутренняя разметки переводятся друг в друга – для редактирования все внешние разметки вносятся в текст, становясь внутренними, а внутренние разметки для хранения и интерпретации снова переводятся во внешнюю форму. В корпусе могут использоваться и разметки обоих классов, тем более что внешняя разметка может с успехом опираться на явную и неявную внутреннюю разметку.

Свои неудобства имеют и внутренние разметки. Во-первых, в тексте они присутствуют все сразу, в том числе не нужные конкретному исследователю, что делает острым вопрос о наглядности синтаксиса разметки. Во-вторых, встаёт вопрос о защищённости исходного текста, который может сниматься

также посредством использования специальных редакторов разметки. Кроме того, конкретные виды систем внутренней разметки накладывают свои ограничения на возможности разметки. Так, среди языков разметки на основе SGML, благодаря наличию возможности отдельно задавать интерпретацию элементов разметки, наибольшее распространение получил XML. Он, однако, в своей основе предъявляет к самой разметке жёсткие требования, практически исключающие использование множества независимых разметок. В итоге для разных исследовательских и практических задач разметка должна делаться на разных экземплярах исходного текста, что снижает надёжность работы.

2. *Разметка в ТСН* – внутренняя. Её принципы складывались под влиянием стремления: а) к *лаконичности* разметки (исходный текст должен оставаться как можно лучше читаемым); б) ко *множественности* разметок в одном тексте (рабочая форма текста должна остаться единственной для разных задач); в) к *независимости* этих разметок (новая вносимая разметка не должна влиять на интерпретацию ранее внесённых); г) к *простоте* описания интерпретации разметки (описание должно быть доступным для непрограммистов).

А. *Лаконичность* обеспечивается следующими средствами.

Во-первых, интерпретация текста опирается не только на явную разметку стандартного формата, но и на неявную, как и на любые дополнительные символы или комбинации символов. Не нуждаются в специальных метках и особые случаи интерпретации, идентифицируемые по сочетанию метки с элементами исходного текста, так как такому сочетанию можно придать особую интерпретацию тем же механизмом, что и самой метке.

Во-вторых, хотя элементы разметки открывают собственные фрагменты текста, к ним не предъявляется строгое требование иметь парный (“закрывающий”) элемент (что не мешает использовать парность при необходимости, поддерживая её при описании интерпретации).

Эти решения делают разметки в тексте теоретически минимальными: метка ставится там и только там, где возникает необходимость в смене интерпретации, не усматриваемая из других маркеров. В результате некоторые

задачи могут решаться без внесения дополнительной разметки в уже имеющиеся корпуса.

Б. *Множественность* разметок обеспечивается возможностью перед каждым помеченным фрагментом вставить новый, служебный фрагмент со своей меткой. Такие фрагменты синтаксически не отличаются от фрагментов исходного текста (и могут нести любое количество информации, содержательной для поддерживаемого данной разметкой аспекта). При этом, если метке фрагмента не придано никакого значения при описании интерпретации, то весь фрагмент (а не только метка) игнорируется. Таким образом, добавление в текст новых меток не влияет на уже существующие интерпретации, в описании которых про эти метки ничего не сказано.

В. *Независимость* разметок реализуется, кроме указанной способности механизма интерпретации игнорировать “чужие” фрагменты, тем, что он (механизм) допускает и произвольное задавание трактовки самого разбиения на фрагменты, позволяя считать существенными для разбиения только нужные метки.

Г. *Простота описания интерпретации* разметки обеспечивается тем, что с точки зрения редактора корпуса она состоит в выписывании в простом текстовом файле (“стилевом файле”) для каждой метки сопоставления ей цепочки символов и *преобразований*, которые следует применить к фрагменту. Преобразования, о которых идёт речь, указываются своим именем, за которым может стоять либо программная функция, либо непрограммное преобразование, задаваемое перечнем простых или рекурсивных замен в специальной текстовой форме. Синтаксически имена тех или других преобразований не различаются, так что редактор корпуса может по-своему переопределять какие-то из системных функций.

Примерами непрограммных преобразований могут служить: перевод представления символов в ту или иную шрифтовую кодировку, удаление ударений, расстановка переносов и/или морфемных границ, лемматизация, снятие/расстановка титл и другие операции, специфичные для орфографической традиции и представляющие сложность не столько алгоритмическую, сколько предметно-специальную. Несмотря на техническую простоту, сами преобра-

зования, получаемые таким образом, могут быть довольно сложными, во всяком случае за ними обычно стоит большой труд специалиста, который представляет собой законченный результат. Предлагаемая форма даёт возможность специалистам обмениваться и такими результатами.

3. *Процесс интерпретации разметки.* Стилевой файл применяется на втором этапе интерпретации. Первый и третий этап задают условия и цель работы разработчика разметки.

Метки стандартного вида начинаются специальным маркером “начало” (часто это знак “%”). Первый этап интерпретации ставит перед ними дополнительный маркер “конец”, так что на границах фрагментов оказывается контаминация этих маркеров. На третьем этапе удаляются все участки текста между этой контаминацией и ближайшим к ней маркером “конец”. Поэтому задача второго этапа, который определяет составитель стилевого файла, – прописать в нём такие замены, в результате которых нужные фрагменты не только не удалились бы на третьем этапе, но и содержали бы в себе предписания нужных преобразований. Используя в таких заменах и сами маркеры “начало” и “конец”, можно гибко влиять на сложившееся в результате пересечения разметок первоначальное “разбиение”.

ТСН реализована на платформе PHP, что было вызвано её первоначальным назначением – представлять в интернете в разных видах древнерусские песнопения. Спецификации, необходимые для переноса Технологии на другие платформы, представлены на сайте проекта “Фонд знаменных песнопений” (URL: http://znamen.ru/tsn/tsnp-zam.php).

Ideology and technique for encoding using the mixed scripts technique

Alexander V. Kovalenin

A. P. Ershov Institute of Informatics Systems of the Siberian Branch of the Russian Academy of Science, Novosibirsk, Russia

This paper discusses common problems in text encoding and proposes their solution using a “mixed scripts technique” (MST), a way of working with texts containing parts which need different interpretation. MST’s way of text markup, together with a procedure of markup interpretation, is given in comparison with other

solutions like XML. A special form of non-programmed user-defined transformations is shown, which can become common for carrying out certain kinds of scientific research.