УДК 026.06

А. В. Шувалов

Применение OCR в технологии каталогизации на примере разработки специализированного программного модуля для ИРБИС64

Представлен обзор средств, призванных облегчить процесс каталогизации и передать ЭВМ часть функций по заполнению полей при составлении записей.

Применяемый автором подход основан на использовании технологии OCR – оптического распознавания символов – и собственных программных решений.

Ключевые слова: автоматизация библиотек, библиографические записи, каталогизация, Система автоматизации библиотек ИРБИС64, оптическое распознавание символов (OCR), язык BIRMA, ввод оглавления.

До сих пор одна из наиболее трудоёмких задач в процессе автоматизации библиотек – это, пожалуй, создание библиографических записей, т.е. непосредственно каталогизация. Однако скорость работы библиографа-каталогизатора может существенно повыситься без заметного ухудшения качества за счёт оснащения библиотек сканерами и специализированным ПО.

Как известно, каталогизация подразумевает последовательное заполнение отдельных полей БЗ. Для журналов и сборников научных трудов, а также других изданий, содержание которых предполагает деление на отдельные смысловые единицы, различающиеся по тематике и имеющие разных авторов, необходимо частично или полностью делать роспись оглавления. До настоящего времени эта работа во многом представляет собой рутинный, слабо автоматизированный процесс.

Технология заимствования записей во многом способствовала облегчению труда библиографа, однако ещё не все библиотеки могут позволить себе участие в ИРБИС-корпорации и МАРС. К тому же сам принцип работы корпорации предполагает, что вы должны не только брать, но и отдавать что-то взамен, следовательно, какое-то количество записей вам всё равно придётся создавать самостоятельно, и чем быстрее они будут созданы, тем скорее другие участники корпорации смогут воспользоваться результатом этого труда. А значит, применение технологий сканирования и распознавания текста уместно и в рамках участия в ИРБИС-корпорации.

Рассмотрим процесс каталогизации на примере САБ ИРБИС64 и её модуля «Каталогизатор». Для начала проанализируем те средства автоматизации, которые нам доступны уже сейчас.

1. Раскрывающиеся списки, которые позволяют заполнять необходимые поля на основе уже имеющихся в БД записей, содержащих такие же или схожие данные для этих полей (например, то же издательство – тогда его не нужно вводить целиком, а достаточно выбрать из списка – после ввода первых букв названия).

Итог: облегчают работу, однако все новые данные приходится вводить вручную.

2. Применение буферной записи для создания новых БЗ на основе уже имеющихся. Безусловно, чем больше общих данных содержат записи, тем быстрее будет осуществляться ввод. Этот подход имеет свои подводные камни, служащие потенциальным источником ошибок: старая запись копируется целиком со всеми её полями, в том числе с теми, значение которых необходимо исправить. Здесь нужно учитывать человеческий фактор, вследствие которого оператор может просто забыть изменить значения некоторых нужных полей.

Итог: не автоматизирует ввод новых данных, приводит к возникновению ошибок.

3. Импорт готовых записей из других БД. Это могут быть БД, как созданные в самой САБ ИРБИС, так и в других АБИС или произвольных СУБД. Безусловно, эта технология эффективнее, чем создание новых записей «с нуля», и её стоит применять во всех случаях, когда нужные записи уже имеются в каком-либо

формате. Но зачастую она требует написания специализированных конверторов и, главное, – не решает проблему первоначального источника получения записей. Другими словами, человек, который прежде занимался созданием записей, сам должен был решить для себя этот вопрос. Сюда же можно отнести и случай импорта готовых записей вследствие участия в корпорации.

Итог: не решает проблему первоначального источника получения записей; зачастую требует создания специализированного ПО для конкретного случая.

4. Копирование содержимого отдельных полей записи через буфер обмена *Windows.* При этом пользователю самому приходится выбирать те ячейки таблицы рабочего листа записи в АРМ «Каталогизатор», куда необходимо вставить текст. К тому же, текстовые данные, вставляемые из буфера обмена, должны быть откуда-то уже перенесены. Если не рассматривать случай переноса БЗ по отдельным частям из какой-либо другой БД, как в третьем варианте (что оправдано лишь в случае небольшого количества таких записей, иначе целесообразнее применять конверторы) или, скажем, со страниц Интернета, то так или иначе придётся задействовать сканер или цифровой фотоаппарат и программы OCR (*Optical Character Recording* – оптическое распознавание символов), чтобы получать эти данные непосредственно со страниц издания. А для этого, как будет показано далее, эффективнее применять программные средства вроде тех, что описываются в данной работе.

Итог: требует применения сканера и программных средств распознавания текста, но не лишает оператора необходимости непосредственного участия в процессе ввода записи, поскольку именно ему приходится каждый раз выбирать нужные поля таблицы для заполнения.

Забегая вперёд, отметим, что наш собственный подход является в какой-то мере совмещением третьего и четвёртого вариантов. Текстовые данные для формируемой записи, взятые со страниц книжных изданий посредством сканера и OCR, обрабатываются так же, как если бы они брались из какой-либо БД, а используемые программные средства являются по сути конверторами из формата представления этих данных в книжных изданиях в формат САБ ИРБИС. В случае обработки сводного библиографического описания, обычно располагающегося на обороте титульного листа, такой конвертор может работать и без участия человека; в случае обработки оглавления его необходимо предварительно настраивать.

Итак, можно сделать вывод, что средства, имеющиеся в модуле «Каталогизатор» САБ ИРБИС64, не обеспечивают автоматизацию самого процесса формирования БЗ. Впрочем, насколько мне известно, и в других автоматизированных библиографических системах эта задача оставляется на усмотрение сторонних разработчиков.

Некоторые подобные средства, демонстрирующие применение сканера и технологий OCR, в частности при решении задачи ретроконверсии карточных каталогов, действительно были созданы, но, судя по всему, они так и не стали достоянием широкой общественности. Примерами могут служить разработки российских фирм «ГИПЕР» (http://www.gpntb.ru/win/inter-events/crimea2004/300.pdf), «ПроСофт-М» (https://eva.rsl.ru/old/2000/eva/200008/lavrionova-r.htm) и белорусской «Агат-Систем» (http://www.agat-system.com/direction/it/itsr/).

Отметим, что независимо от того, как обстоят дела с распознаванием каталожных карточек, которые могут иметь много вариантов оформления, сводное библиографическое описание, которым снабжается практически каждое современное издание, достаточно стандартизовано. И если не браться за разработку некоего всеобъемлющего безошибочного алгоритма на все случаи жизни, то выделение в нём отдельных полей представляет собой достаточно тривиальную задачу.

Причиной, по которой подобная технология до сих пор не встроена непосредственно в АРМ «Каталогизатор», может быть повышение стоимости конечного продукта вследствие включения в него лицензии на систему OCR, а также слишком узкий сектор потенциальных пользователей: известно, что в настоящее время не во многих библиотеках нашей страны используются сканеры. Но поскольку многие сканеры уже имеют в комплекте программу распознавания символов (чаще всего это *ABBYY FineReader*), то решение второй проблемы повлечёт за собой и автоматическое решение первой – ведь текстовые данные для библиографической обработки (представления в виде отдельных полей БЗ) необязательно должны формироваться самой программой путём внутреннего вызова функций OCR. Они могут переноситься посредством буфера обмена и из самого приложения *FineReader* или подобного ему. Дело лишь за тем,

чтобы оснастить библиотеки сканерами с входящими в комплект программами для OCR.

Сама библиографическая обработка распознанного текста может осуществляться небольшими программными модулями, подключаемыми в качестве плагинов к различным АРМам конкретной АБИС, таким как АРМ «Каталогизатор» САБ ИРБИС64, либо приложениями, оформленными в виде самостоятельных АРМов. Один из примеров последних – АРМ «АльтерВвод» (от слов «альтернативный ввод»), специально разработанный автором для ИРБИС64.

АРМ «АльтерВвод» представляет собой упрощённый клиент для каталогизации, «заточенный» на импорт записей из внешних источников. Для этого он обладает как возможностями подключения плагинов в привычном формате ИРБИС64, так и собственным встроенным библиографическим редактором. Для него было разработано внешнее приложение БАРТ (библиографический анализ распознанного текста) в формате обычного плагина ИРБИС64 в расчёте на то, что его можно будет вызывать и из АРМа «Каталогизатор». БАРТ позволяет импортировать практически готовую запись из предварительно отсканированных и распознанных текстовых данных непосредственно в систему.

Приложение БАРТ основано на том, что сводное библиографическое описание, помещаемое на обороте титульного листа или в конце всех современных изданий, имеет достаточно чёткую структуру и без особых проблем автоматически разбивается на отдельные поля, из совокупности которых можно сразу же составить требуемую запись. Следовательно, необходимо было только реализовать соответствующий алгоритм. Отмечу: мой собственный алгоритм, конечно, не идеален и не учитывает всех возможных вариантов структурирования элементов, но, безусловно, способствует ускорению работы по каталогизации – от оператора требуется только поправить некоторые возможные ошибки, возникшие в ходе восстановления структуры записи.

В дополнение к возможности подключения внешних плагинов в формате САБ ИРБИС64, АРМ «АльтерВвод» предлагает и собственный текстовый редактор для выделения в тексте требуемых элементов БЗ с последующим переносом их в соответствующие поля текущей записи – так называемый *библиографический редактор*.

Кроме простого переноса текстовых данных из программы OCR посредством буфера обмена *Windows,* реализована также возможность подключения внешних плагинов к библиографическому редактору, на выходе которых в программу передаются готовые текстовые данные. Таким образом, пользователь может создать свой собственный встраиваемый программный инструмент, реализующий OCR на базе любой из существующих для этой цели библиотек, и ценовая политика конечного продукта АРМ «АльтерВвод» не будет подразумевать жёсткую привязку к созданным разработчиками интегрированным средствам OCR (т.е. библиотекам не придётся платить за лицензирование встроенного движка, если они, к примеру, уже приобрели сканер или сканирующую ручку с входящим в его комплект приложением *FineReader*).

Для восстановления структуры БЗ из текстовых данных, поступающих на вход библиографического редактора, был создан специальный язык BIRMA (*Bibliographic Information Recognition MArkup Language*), или БИРМА (библиографический инструмент распознавания маркировки). Соответствующая программная библиотека *BIRMA.DLL* содержит ряд функций для библиографического распознавания полей записи и может подключаться к произвольным разрабатываемым программам, требующим поддержки этого языка.

Язык BIRMA по своей структуре похож на язык форматирования ISIS, хотя выполняет, можно сказать, противоположную задачу: не формирует текст из элементов БЗ, а, наоборот, служит для составления запросов, осуществляющих выделение этих отдельных элементов из исходного текста. Правила этого языка лучше всего рассмотреть на примере конкретной задачи, а именно – восстановления структуры оглавления.

Предположим, мы имеем такой текст оглавления (взят из сборника научных статей):

Восстановить структуру этого оглавления и перевести его в формат ИРБИС нам поможет следующий запрос на языке BIRMA:

(&call_func('partmarker', '<буква>.{ }<буква>.')+(v330^F)&call_func
('partmarker', '<Ввод>'), &call_func('partmarker', '{ }')v330^C'\t', &call_func
('partmarker', '<цифра>')+v330^4&call_func('partmarker', '<Ввод>'))

Как видим, запрос состоит из нескольких предложений – подзапросов, разделённых запятой и обрамлённых круглыми скобками. Внешние скобки означают, что все содержащиеся в них отдельные запросы должны выполняться циклически, пока не достигнут конца текста. Каждый запрос может включать в себя идентификатор искомого поля записи, за которым может следовать разделитель его конкретного подполя.

В нашем случае в каждом из запросов, содержащихся в скобках, имеется подобный элемент: v330^F, v330^C и v330^4. Это означает, что мы ищем в тексте элементы, соответствующие трём подполям поля оглавления 330: ФИО первого автора, заглавию и номеру страницы. Поскольку эта группа запросов заключена в круглые скобки, они выполняются последовательно до конца исходного текста с увеличением счётчика

повторений поля 330 на каждом шаге цикла.

Таким образом, мы последовательно просматриваем весь текст и выделяем из него нужные элементы для каждого из пунктов оглавления. Мы видим, что первая конструкция (v330^F) также заключена в скобки. Это означает, что элемент является необязательным, т.е. на каждом из повторений поле 330 может либо содержать либо не содержать подполе F (ФИО). Если в процессе анализа текста следующий обязательный элемент (который запишется в v330^C) найден раньше, чем очередной необязательный элемент (v330^F), считается, что в этом повторении поля его нет. В нашем случае мы видим, что текст оглавления начинается с заголовка раздела, т.е. на этом повторении мы сразу же выделяем заглавие и не вводим никаких данных в подполе ФИО.

Справа и слева от идентификаторов конкретного поля/подполя должны стоять текстовые фрагменты, содержащие последовательности символов, которые составляют границы элемента в исходном тексте. Другими словами, справа и слева от выделяемых элементов в исходном тексте должны содержаться эти последовательности. В качестве составляющих их символов могут выступать, например, символы пробела, табуляции, точки и др. Возможно использование регулярных выражений, в том числе в виде выражений на специально разработанном автором языке ПАРТИЗАН (парсер-анализатор регулярной текстовой информации с заложенной автоматизацией набора).

Встречающиеся слева и справа от идентификаторов поля выражения вида &**call_func**('partmarker', **expr**) говорят о том, что вызывается функция partmarker, аргумент которой **expr** как раз представляет собой выражение языка ПАРТИЗАН. В первом подзапросе выражение **<буква>.{ }<буква>.** обозначает две подряд идущие буквы русского языка с точками после каждой, между которыми может находиться любое количество пробелов (или ни одного). Стоящее справа в этом запросе выражение **<Ввод>** означает просто перевод строки (при анализе заменяется на последовательность управляющих символов **\r\n**). Таким образом, весь запрос можно расшифровать так: мы считаем значением для ФИО в каждом из повторений поля текстовый фрагмент, заключённый между инициалами (**<буква>.{ }<буква>.**) и символами перевода строки. Знак «+», помещённый после выражения, обозначающего левую границу, и перед идентификатором поля, означает, что последовательность символов, которой соответствует данное выражение (**<буква>.{˙} <буква>.**) на текущем повторении цикла в исходном тексте, включается в формируемый текстовый фрагмент (т.е. в данном случае инициалы служат левой границей для выделения значения ФИО, но с них же и должно начинаться формируемое значение). Знак «+», стоящий после идентификатора поля/подполя, будет означать точно такую же необходимость включения идущего следом за ним в запросе значения правой границы в формируемое значение соответствующего текстового фрагмента.

Второй подзапрос содержит такие выражения для границ соответствующего найденного фрагмента: **&call_func('partmarker', '{ }')** и **'\t'**. Это означает, что мы берём текстовый фрагмент, располагающийся после произвольного количества подряд идущих пробелов (подразумевается, что они стоят в начале строки) и до управляющего символа табуляции, и считаем его очередным значением подполя C (заглавия) на данном повторении его поля 330.

Сразу за ним следует запрос **&call_func('partmarker', '<цифра>')+v330^4&call_func('partmarker', '<Ввод>')**, означающий, что мы просматриваем исходный текст далее до появления в нём первой цифры, отсчитываем от неё (включая и саму эту цифру) текстовый фрагмент до появления управляющего символа перевода строки и считаем его очередным значением подполя 4 (номер страницы) на данном повторении поля 330. Далее вся последовательность запросов повторяется для формирования следующего пункта оглавления.

Отметим, что в большинстве видов текста оглавления, помещаемого в начале или в конце издания, заглавие очередного пункта оглавления обычно отделяется от следующего за ним номера страницы отточием, т.е. множеством точек, заполняющих строку до её правого края, по которому располагается числовое значение номера страницы. Такой последовательности символов примерно соответствует выражение языка ПАРТИЗАН .**{.}** (или, чтобы не путать с точками внутри самого заглавия, – ..**{.}**), но, как правило, существующие программы и программные библиотеки OCR заменяют эту последовательность на один или несколько символов табуляции. Так что запрос **&call_func('partmarker', '{.}')v330^C'\t'** подходит для множества случаев выделения заглавия/названия для статей или отдельных разделов сборников.

Безусловно, для текстов оглавлений современных изданий характерны разнообразные варианты разметки, но

практически все их случаи можно строго формализовать средствами языка BIRMA. Для сложных случаев предусмотрена команда ветвления, которая позволяет модифицировать в запросе значения границ для элемента на основании найденных значений границ предыдущих элементов или менять порядок следования элементов более сложным образом, что позволяют сделать круглые скобки, в которые заключён идентификатор поля.

Однако, как показывает моя личная практика, даже применение одних только описанных простейших средств уже позволяет значительно автоматизировать весь процесс ввода практически любого оглавления. Время, требуемое на последующую обработку с исправлением ошибок и дополнительным вводом информации (некоторые поля вследствие неполноты применённого запроса могут оказаться незаполненными или заполненными частично), не идёт ни в какое сравнение с последовательным копированием и вставкой текстовых фрагментов в нужные подполя рабочего листа вручную. Про перепечатывание вручную всех пунктов оглавления, т.е. клавиатурный ввод, и говорить не приходится.

Подчеркну ещё раз: при использовании традиционного подхода процесс росписи оглавления – ещё более трудоемкий, чем создание самой записи. Оглавление может состоять из нескольких страниц текста. Обработка даже распознанного текста, но без применения специализированного ПО, представляет собой чисто механический труд с многократным переносом нужных данных из буфера обмена в соответствующие поля таблицы рабочего листа.

Таким образом, использование АРМ «АльтерВвод» и языка BIRMA– это единственная на сегодня технология быстрого ввода оглавления, хотя, безусловно, одним только оглавлением область их применения не исчерпывается.