Алгоритмы преобразования сложноструктурированных объектовдля синтеза речи по тексту

Л.И. Цирульник,
кандидат технических наук
Ю.С. Гецевич,
аспирант

Одним из путей расширения использования систем синтеза речи является обработка и озвучивание не только текстовой информации, но и сложноструктурированных объектов, таких как таблицы, рисунки, формулы и т.д. Преобразование подобных объектов в орфографический текст является частным случаем задачи анализа сцен, но требует создания специальных алгоритмов, учитывающих структуру обрабатываемых объектов. При этом критерием корректности созданных алгоритмов должна являться, по мнению авторов, не только достоверность полученной информации, но и адекватность смыслового восприятия сформированного орфографического текста.
В работе рассматриваются сложноструктурированные объекты MS Word, предлагается шкала оценок сложноструктурированных объектов по критерию их сложности для смыслового восприятия, приводятся алгоритмы преобразования таких объектов в орфографический текст, описываются особенности программной реализации разработанных алгоритмов.

Abctract
One of the ways to extend the application of text-to-speech synthesis systems is to process not only textual information, but also complex-structured objects such as tables, figures, formulas, etc. The conversion of complex-structured objects into orthographic text is a particular case of scene analysis problem. Such conversion requires the development of special algorithms which take into account the structure of processed objects. The criteria of correctness of created algorithms are, on the author's opinion, not only accuracy of obtained information, but also adequacy of sense understanding of generated orthographic text.

The complex-structured objects of MS Word are considered in the paper, the rating scale of complex-structured objects by criterion of sense understanding complexity is suggested, the algorithms of transformation of complex-structured objects into orthographic text are given, the specificities of software implementation of developed algorithms are described.

Введение

При разработке систем синтеза речи по тексту предполагается, как правило, что на вход системы подаётся текстовый файл, содержащий, впрочем, не только орфографический текст, но и аббревиатуры, числа, сокращения и т.д. [1].Однако практика показывает, что такое ограничение на формат входного файла затрудняет широкое использование системы синтеза речи по тексту. Входная информация во многих случаях может содержаться в файлах в pdf, doc и других широко распространённых форматах.

Наиболее удачное решение озвучивания информации в произвольном формате — это предварительное её преобразование к одному определённому формату и последующая обработка стандартными модулями системы синтеза речи. При этом, очевидно, блок преобразования формата должен входить в состав системы синтеза речи.

Преобразование форматов входных данных требует решения дополнительных задач. Многие широко распространённые форматы допускают наличие в тексте рисунков, формул, таблиц и других объектов, требующих особой обработки. Оценкой корректности производимой обработки должна быть, по мнению авторов, адекватность смыслового восприятия сформированного текста.

В данной работе описываются исследования сложноструктурированных объектов MS Word и алгоритмы их преобразования в орфографический текст, который подаётся на вход системы синтеза речи по тексту. Под *сложноструктурированным объектом* понимается любой фрагмент содержимого файла MS Word, который: а) не является текстом и требует преобразования к текстовому формату либо б) является фрагментом текста, требующим особого интонационного выделения.

Примером первого типа сложноструктурированных объектов являются формулы, примером второго типа — заголовки.

1. Типы сложноструктурированных объектов MS Word и их классификация

Для выявления различных типов сложноструктурированных объектов были проанализированы текстовые документы MS Word, относящиеся к научному и художественному стилям. Соответствующие doc-файлы содержат в сумме 700 страниц и включают 43 таблицы, 67 формул и 211 рисунков.

В результате анализа выделены следующие основные типы сложноструктурированных объектов: оглавление, заголовок, список, перекрёстная ссылка, таблица, рисунок, формула.

Необходимо отметить, что в общем случае сложноструктурированные объекты могут быть вложенными: например, формула или рисунок могут являться содержимым ячейки таблицы.

Выявленные сложноструктурированные объекты, а также их составляющие были классифицированы по степени их значимости для смыслового восприятия текста. Для такой классификации разработана шкала оценок, приведённая в ***таблице 1***.

Таблица 1

Шкала оценок значимости объектов для смыслового восприятия текста

Оценка	Значение
0	Объект является незначимым и не должен быть преобразован в орфографический текст
1	Объект является значимым, но сложным для понимания смысла при восприятии соответствующего текста на слух
2	Объект является значимым; без его преобразования к орфографическому тексту и последующего озвучивания будет утерян смысл фрагмента текста

Детальный анализ сложноструктурированных объектов позволил выявить их структуру и оценить значимость самих объектов и их составляющих для смыслового восприятия текста.

1.1. Структура и степень значимости объекта «Оглавление»

Объект «Оглавление» представляет собой перечень строк. Каждая строка содержит текст заголовка, заполнитель (последовательность одинаковых символов) и номер страницы. Заполнитель и номер страницы не являются обязательными составляющими строки оглавления.

Степень значимости составляющих объекта «Оглавление», а также соответствующие пояснения приведены в таблице 2.

Таблица 2

Степень значимости составляющих объекта «Оглавление»

Наименование составляющей объекта «Оглавление»	Степень значимости	Комментарии
Текст заголовка	2	При озвучивании позволяет слушателю получить общее представление о содержимом текста
Заполнитель	0	Не является информативным
Номер страницы	1	При озвучивании позволяет слушателю получить представление об объёме соответствующего раздела

1.2. Структура и степень значимости объекта «Заголовок»

Объект «Заголовок» представляет собой абзац, отличительной особенностью которого является наличие в соответствующей строке особого стиля: например, «Заголовок 1», «Заголовок 2» и т.д.

Степень значимости объекта «Заголовок» для смыслового восприятия текста равна 2.

1.3. Структура и степень значимости объекта «Список»

Объекты «Список» в MS Word делятся на две основные категории: маркированные (характеризующиеся наличием маркера перед каждым элементом списка) и нумерованные (характеризующиеся наличием порядкового номера перед каждым элементом списка).

Каждый элемент и маркированного, и нумерованного списков может состоять, в общем случае, из части предложения (при перечислении), целого предложения или нескольких предложений. Элементы маркированного списка отделяются, как правило, точкой с запятой («;»), а элементы нумерованного списка заканчиваются, как правило, точкой.

Степень значимости для смыслового восприятия составляющих объекта «Список» приведена в таблице 3.

Таблица 3

Степень значимости составляющих объекта «Список»

Наименование составляющей объекта «Список»	Степень значимости	Комментарии
Текст элемента списка	2	Отражает смысловое содержание
Маркер	0	Не является информативным
Порядковый номер	2	Является элементом, способствующим пониманию смысла текста

1.4. Структура и степень значимости объекта «Перекрёстная ссылка»

В документе MS Word могут содержаться перекрёстные ссылки следующих типов: абзац, заголовок, закладка, сноска, концевая сноска, рисунок, таблица, формула.

В качестве перекрёстной ссылки в документе может находиться:

- текст объекта (заголовка, закладки, абзаца) либо название объекта (рисунка, таблицы, формулы);
- порядковый номер соответствующего объекта (абзаца, закладки, (концевой) сноски, заголовка; номер абзаца, в котором находится закладка);
- номер страницы, на которой находится соответствующий объект;
- слово «выше» или «ниже».

Степень значимости каждой из категорий приведена в таблице 4.

Таблица 4

Степень значимости различных категорий объекта «Перекрёстная ссылка»

Тип перекрёстной ссылки	Степень значимости	Комментарии
Текст либо название объекта	2	Является пояснением сути перекрёстной ссылки
Порядковый номер объекта	0; 2	При всех типах перекрёстных ссылок, кроме (концевой) сноски, не является информативной и имеет степень значимости, равную 0; в случае перекрёстной ссылки на (концевую) сноску степень значимости равна 2
Номер страницы, на которой находится соответствующий объект	0	Не является информативным
Слово «выше» или «ниже»	2	Содержит пояснительную информацию

1.5. Структура и степень значимости объекта «Таблица»

Таблица состоит из двух основных составляющих: названия и непосредственно таблицы.

Название таблицы находится, как правило, перед таблицей и состоит из слова «таблица», за которым может следовать порядковый номер, и наименования таблицы.

Степень значимости названия таблицы равна 2.

Непосредственно таблица имеет следующие характеристики, важные для её преобразования в текстовый вид: наличие или отсутствие заголовков (подзаголовков) строк и столбцов; количество заголовков (подзаголовков) строк и столбцов; количество столбцов; количество строк.

Степень значимости непосредственно таблицы зависит от её сложности, которая может быть вычислена в соответствии с формулой:

$$S = (k_r + k_{pr}) n_r + (k_c + k_{pc}) n_c + k_m \sum_{i=1}^{l} S_i , \qquad (1)$$

где k_r, k_{pr}, k_c, k_{pc}, k_m — коэффициенты сложности, соответственно, строк таблицы; заголовков строк таблицы; столбцов таблицы; заголовков столбцов таблицы; сложноструктурированных объектов, являющихся содержимым таблицы; n_r, n_c, l — количество, соответственно, строк, столбцов и сложноструктурированных объектов в таблице; S_i — сложность i-го сложноструктурированного объекта, входящего в таблицу.

Используемые в формуле (1) коэффициенты удовлетворяют уравнению:

$$k_r + k_{pr} + k_c + k_{pc} + k_m = 1$$

— и вычисляются экспериментальным путём.

Соответствие сложности таблицы и её степени значимости также определяется экспериментально; увеличение сложности таблицы влечёт уменьшение степени её значимости для смыслового восприятия текста.

1.6. Структура и степень значимости объекта «Рисунок»

Рисунок состоит из двух основных составляющих: подрисуночной подписи и непосредственно рисунка.

Подрисуночная подпись — это отдельный абзац, состоящий, как правило, из слова «Рисунок» (или «Рис.»), за которым может следовать порядковый номер рисунка и его название.

Степень значимости подрисуночной подписи для смыслового восприятия текста равна 2.

Непосредственно рисунки делятся на следующие основные категории:

- растровые изображения;
- объекты внешних приложений, позволяющие создавать графики, диаграммы и т.д., например, объекты MS Visio или MS Excel;
- рисунки, выполненные с использованием средств рисования MS Word (Word-рисунки);
- «смешанные» рисунки, содержащие две или более составляющих, принадлежащих к различным категориям.

Степень значимости непосредственно рисунка зависит от его типа и сложности. Обзор степеней значимости рисунков для смыслового восприятия приведён в таблице 5.

Таблица 5

Степень значимости составляющих объекта «Рисунок»

Категория рисунка	Степень значимости	Комментарии
Растровое изображение	0	В общем случае не может быть преобразовано в текст, описывающий его содержимое
Объект внешнего приложения	0	Требует обработки фрагментов, не являющихся объектами MS Word
Word-рисунок	0–2	В зависимости от сложности объекта
«Смешанный» рисунок	0	Содержит фрагменты, которые в общем случае не могут быть преобразованы в текст

Сложность рисунка определяется в зависимости от количества и типа составляющих его фигур и может быть вычислена по формуле:

$$S = k_l n_l + k_b n_b + k_a n_a + k_{bs} n_{bs} + k_m n_m + k_s n_s \qquad (2)$$

где k_l, k_b, k_a, k_{bs}, k_m, k_s — коэффициенты сложности, соответственно, линии, основной фигуры, фигурной стрелки, элемента блок-схемы, выноски, звезды или ленты; n_l, n_b, n_a, n_{bs}, n_m, n_s — количество, соответственно, линий, основных фигур, фигурных стрелок, элементов блок-схемы, выносок и звёзд или лент на рисунке.

Используемые в формуле (2) коэффициенты удовлетворяют уравнению:

$$k_l + k_b + k_a + k_{bs} + k_m = 1$$

— и вычисляются экспериментальным путём.

Соответствие сложности рисунка и его степени значимости также должно быть определено экспериментальным путём; при этом при увеличении сложности рисунка его степень значимости будет уменьшаться.

1.7. Структура и степень значимости объекта «Формула»

Объект «Формула» может рекурсивно включать в себя следующие структуры: дроби; верхние и нижние индексы; радикалы; крупные операторы (например, сумма, произведение и т.п.); круглые, квадратные, фигурные, угловые скобки и их разновидности; скобки с разделителями; тригонометрические, обратные тригонометрические, гиперболические, обратные гиперболические функции; диакритические знаки; матрицы.

Кроме того, формула может содержать обычный (не являющийся математическим) текст, а также следующие типы символов: основные математические символы; греческие буквы; буквоподобные символы; операторы; стрелки; отношения с отрицанием; геометрические символы.

Значимость объекта «Формула» для смыслового восприятия текста, как и объектов «Рисунок» и «Таблица», зависит от его сложности. Очевидно, что если формула обладает большой сложностью, то преобразование её в текстовый вид и последующее озвучивание вызовет у слушателя затруднения при восприятии сути формулы. Следовательно, при увеличении сложности формулы степень её значимости для смыслового восприятия текста уменьшается. Сложность формулы может быть вычислена в соответствии с рекурсивным выражением:

$$S = k_{fr}S_{fr} + k_iS_i + k_rS_r + k_pS_p + k_bS_b + k_{fn}S_{fn} + k_dS_d + k_mS_m \qquad (3)$$

где k_{fr}, k_i, k_r, k_p, k_b, k_{fn}, k_d, k_m — коэффициенты сложности, соответственно, дроби, индекса, радикала, крупного оператора, скобки, функции, диакритического знака, матрицы; S_{fr}, S_i, S_r, S_p, S_b, S_{fn}, S_d, S_m — среднее значение сложности, соответственно, дробей, индексов, радикалов, крупных операторов, скобок, функций, диакритических знаков, матриц.

Коэффициенты формулы (3) удовлетворяют уравнению:

$$k_{fr} + k_i + k_r + k_p + k_b + k_{fn} + k_d + k_m = 1 \qquad (4)$$

Для составляющих объекта «Формула», не содержащих внутри себя дробей, индексов, радикалов, крупных операторов, скобок, функций, диакритических знаков и матриц, сложность может быть вычислена в соответствии с выражением:

$$S = k_sn_s + k_{gr}n_{gr} + k_ln_l + k_on_o + k_an_a + k_rn_r + k_{gm}n_{gm} \qquad (5)$$

где k_s, k_{gr}, k_l, k_o, k_a, k_r, k_{gm} — коэффициенты сложности, соответственно, основных математических символов, греческих букв, буквоподобных символов, операторов, стрелок, отношений с отрицанием, геометрических символов; n_s, n_{gr}, n_l, n_o, n_a, n_r, n_{gm} — количество, соответственно, основных математических символов, греческих букв, буквоподобных символов, операторов, стрелок, отношений с отрицанием и геометрических символов в формуле.

Коэффициенты формулы (5) удовлетворяют уравнению:

$$k_s + k_{gr} + k_l + k_o + k_a + k_r + k_{gm} = 1 \qquad (6)$$

Коэффициенты, входящие в состав формул (4) и (6), вычисляются экспериментальным путём.

2. Алгоритмы преобразования сложноструктурированных объектов MS Word к орфографическому тексту

Как уже указывалось ранее, под сложноструктурированным объектом понимается не только фрагмент текста, который требует преобразования к текстовому формату, но и фрагмент, требующий особого интонационного выделения. Для того, чтобы распознанный в Word-документе объект второго типа мог быть идентифицирован на последующих этапах преобразования текста и соответствующим образом обработан, были введены специальные теги, добавляемые в выходной документ, которые являются индикаторами наличия того или иного объекта: например, строки оглавления, заголовка и т.д.

2.1. Алгоритмы преобразования объектов, требующих особого интонационного выделения

К таким объектам относятся: оглавление и его составляющие; заголовок; элементы списка; перекрёстная ссылка.

При обработке входного документа распознаётся наличие сложноструктурированного объекта, он разбивается на составляющие, и затем каждая из составляющих определённым образом обрабатывается. Составляющие, степень значимости которых для смыслового восприятия текста равна 0, удаляются из документа.

Правила преобразования этих объектов в процессе обработки документа представлены в таблице 6.

Таблица 6

Правила преобразования некоторых сложноструктурированных объектов в текст

Входной объект или его составляющая	Выходной текст
Текст заголовка оглавления	<content_a> Текст заголовка оглавления </content_a>
Номер страницы в заголовке оглавления	<content_p> Номер страницы в заголовке оглавления </content_p>
Заголовок	<paragraph_a > Заголовок </ paragraph_a >
Номер элемента списка	<list_n> Номер элемента списка</list_n>
Текст элемента списка	<list_t> Текст элемента списка </list_t>
Слова «Выше» («Ниже») как содержимое перекрёстной ссылки	<cross_reference_ab> «Выше» («Ниже») </cross_reference_ab>
Текст как фрагмент перекрёстной ссылки	<cross_reference_t> Текст </cross_reference_t>
Номер как фрагмент перекрёстной ссылки	<cross_reference_n> Номер </cross_reference_n>

2.2. Алгоритмы обработки объектов, требующих преобразования к текстовому формату

К объектам, требующим преобразования к текстовому формату, относятся таблицы, рисунки и формулы. В общем случае можно определить три варианта преобразования к текстовому формату таких объектов:

- **краткий,** когда для таблиц преобразуются к текстовому виду только их названия, для рисунков — только подрисуночные подписи, формулы заменяются словом «формула»;

- **средний,** при котором преобразуются к текстовому виду таблицы, рисунки и формулы, сложность которых не превышает некоторого заданного порога (причём значение порога будет различным для разных типов объектов); для объектов, сложность которых превышает порог, алгоритм преобразования будет таким же, как и в первом случае;

- **подробный,** при котором преобразуются к текстовому виду все объекты, вне зависимости от их сложности.

В последнем случае смысловое восприятие текста будет, вероятно, затруднено из-за большой сложности объектов. Тем не менее, целесообразно предоставить пользователю системы возможность выбирать наиболее подходящий для него режим.

Очевидно, что в любом случае не будут преобразованы к текстовому виду рисунки, которые являются растровыми изображениями или объектами внешних приложений.

2.2.1. Алгоритм преобразования таблиц в орфографический текст

На вход алгоритма подаётся таблица MS Word. В результате преобразования формируется орфографический текст, включающий заголовки и подзаголовки строк таблицы, столбцов таблицы, содержимое ячеек таблицы, а также специальные теги для заголовков и подзаголовков строк, столбцов и содержимого ячеек.

Алгоритм состоит из следующих шагов.

Шаг 1. Вычисляется общее количество строк таблицы n, количество заголовков и подзаголовков строк nc, общее количество столбцов таблицы m, количество заголовков и подзаголовков столбцов mc.

Шаг 2. Текущий номер строки i принимает значение $(nc+1)$.

Шаг 3. Текущий номер столбца j принимает значение (mc+1).

Шаг 4. Для всех k от 1 до nc формируется текст: <table_header_r> $T[i,k]$ </table_header_r>, где i — номер строки таблицы, k — номер столбца таблицы, $T[i, k]$ — содержимое ячейки i, k таблицы T.

Шаг 5. Для всех l от 1 до mc формируется текст: <table_header_c> $T[l,j]$ </table_header_c>.

Шаг 6. Формируется текст: <table_cell> $T[l,j]$ </table_cell>.

Шаг 7. Значение j увеличивается на 1.

Шаг 8. Если $j \le m$, переход к шагу 4. Иначе — переход к шагу 9.

Шаг 9. Значение i увеличивается на 1.

Шаг 10. Если $i \le n$, переход к шагу 3. Иначе — конец алгоритма.

Отличительными особенностями данного алгоритма являются следующие:

а) перечисление заголовков и подзаголовков (при их наличии) строки и столбца таблицы перед содержимым ячейки, которая находится на пересечении данных строки и столбца (благодаря такому перечислению из сформированного текста понятно, что именно отражает содержимое каждой ячейки; это особенно важно при «озвучивании» таблиц, содержащих большое количество строк и столбцов или несколько подзаголовков строк и столбцов);

б) вставка в текст специальных тегов, указывающих начало/конец заголовков (подзаголовков) строк и столбцов, а также начало/конец содержимого ячейки таблицы (благодаря вставке специальных тегов появляется возможность особого интонационного выделения заголовков и подзаголовков строк и столбцов, а также содержимого ячеек таблицы на последующих этапах обработки текста).

2.2.2. Алгоритм преобразования рисунков к орфографическому тексту

На вход алгоритма подаётся рисунок, включающий только фигуры MS Word. В результате преобразования формируется орфографический текст, содержащий названия фигур, тексты фигур, а также специальные теги для названий и текстов фигур.

Алгоритм включает следующие шаги.

Шаг 1. «Считывается» первая фигура от верхнего левого угла рисунка (определяется по координатам фигур).

Шаг 2. Если фигура не принадлежит к одному из классов «Основные фигуры», «Элементы блок-схемы», «Выноски», «Звёзды и ленты», то переход к шагу 5.

Шаг 3. Формируется текст: <figure_shape> <Название фигуры> </figure_shape>, где <Название фигуры> — текстовое название конкретной фигуры, например, «прямоугольник», «ромб», «куб» и т.д.

Шаг 4. Если внутри фигуры содержится текст, то формируется текстовая последовательность: <figure_text> <Текст фигуры> </figure_text>, где <Текст фигуры> — текстовое содержимое фигуры.

Шаг 5. Если «считаны» все фигуры, то переход к шагу 6, иначе — считывается очередная фигура по принципу «сверху вниз, слева направо» (определяется по координатам фигур) и осуществляется переход к шагу 2.

Шаг 6. Конец алгоритма.

Особенности данного алгоритма следующие:

а) к текстовому формату не преобразуются стрелки и соединительные линии (эта особенность связана с тем, что стрелка может быть направлена из объекта в тот же объект; стрелки могут циклически соединять несколько объектов, причём данные объекты не охватывают всего рисунка; из одного объекта может «выходить» несколько стрелок, направленных в различные объекты);

б) если одна и та же фигура входит в классы «Основные фигуры» и «Элементы блок-схемы», то при её преобразовании к тексту формируется название, взятое из класса «Основные фигуры» (к таким фигурам относится, например, ромб, название которого в классе «Основные фигуры» — «ромб», а в классе «Элементы блок-схемы» — «Блок-схема: решение»).

Как показали результаты экспертного тестирования разработанного алгоритма, выбор названия фигуры из класса «Основные фигуры» способствует лучшему смысловому восприятию сформированного текста.

2.2.3. Алгоритм преобразования формул к орфографическому тексту

На вход алгоритма подаётся формула MS Word. В результате преобразования формируется орфографический текст, содержащий названия элементов формулы и специальные теги для различных типов элементов.

Шаги алгоритма следующие.

Шаг 1. Переменной $Formula$ присваивается значение исходной формулы.

Шаг 2. Переменной F_i присваивается значение очередного элемента формулы.

Шаг 3. Если F_i не принадлежит ни одному из классов «дробь», «индекс», «радикал», «крупный оператор», «скобка», «функция», «диакритический знак», «матрица», «интеграл», «логарифм», то переход к шагу 7.

Шаг 4. В зависимости от класса F_i формируется определённый открывающий тег: например, для класса «дробь» — <formula_fr>, для класса «индекс» — <formula_i>, для класса «радикал» — <formula_r> и т.д.

Шаг 5. Переменной $Formula$ присваивается значение F_i; осуществляется рекурсивный переход к шагу 2.

Шаг 6. В зависимости от класса F_i формируется соответствующий закрывающий тег: например, для класса «дробь» — </formula_fr>, для класса «индекс» — </formula_i>, для класса «радикал» — </formula_r> и т.д. Затем осуществляется переход к шагу 8.

Шаг 7. Формируется текст, соответствующий названию элемента F_i.

Шаг 8. Если достигнут конец формулы, переход к шагу 9, иначе — переход к шагу 2.

Шаг 9. Конец алгоритма.

Особенностью алгоритма является то, что он учитывает наличие «сложных» элементов, таких как дробь, радикал, крупный оператор и т.д., которые, в свою очередь, обрабатываются так же, как исходная формула, благодаря рекурсивной структуре алгоритма. Кроме того, при реализации алгоритма была составлена таблица соответствия символов, используемых в формулах, и их текстовых эквивалентов, например, ε — «эпсилон», sin — «синус» и т.д.

3. Особенности программной реализации алгоритмов

Обработка файлов MS Word осуществляется в два этапа. На первом этапе входной файл в формате doc преобразовывается к валидному xml-документу. На втором этапе в xml-документе на основе тегов распознаются сложноструктурированные объекты, в результате преобразования которых формируется последовательный орфографический текст, включающий специальные теги, маркирующие сложноструктурированные объекты.

3.1. Особенности преобразования документа MS Word к xml-документу

Входными данными обработчика является произвольный документ MS Word. В процессе обработки формируется временный html-документ, который затем преобразовывается в корректный по оформлению и наполнению

xml-документ (***рис. 1***). Для обработки файлов в форматах html и xml используются стандартные библиотеки классов Microsoft (Microsoft.Office. Interop.Word.dll, SgmlReaderDll.dll).

Основным действием преобразования к xml-документу является фильтрация тегов и их атрибутов, после которой в результирующем документе остаются только теги и атрибуты, необходимые для идентификации и обработки сложноструктурированных объектов.

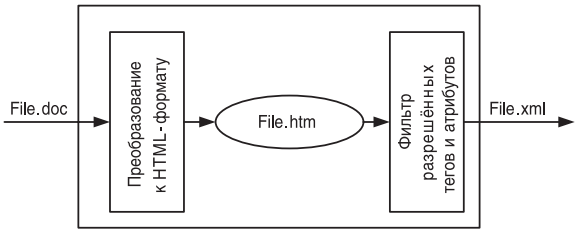

Рис. 1. Процесс преобразования документа MS Word в xml-документ

3.2. Особенности преобразования сложноструктурированных объектов в орфографический текст

Существует два подхода к обработке xml-документов, принципиальная разница между которыми заключается в способе загрузки документа и доступа к его содержимому. При первом подходе, реализованном в рамках DOM-технологии (Document Object Model — объектная модель документа) [2] xml-документ полностью загружается в оперативную память и после этого обрабатывается. При втором подходе, реализованном в рамках SAX-технологии (Simple API for XML) [3], обработка документа происходит последовательно, с использованием относительно небольшого буфера выделенной памяти, в котором хранится минимально необходимая для обработки документа информация.

Преимуществом DOM-технологии[1] является возможность произвольного доступа к содержимому документа. Специальные языки программирования, разработанные для DOM-технологии, позволяют очень удобно записывать правила обработки для целых групп тегов или конкретных атрибутов тегов. Основным недостатком DOM-технологии является использование большого объёма оперативной памяти, так как для обработки документа необходимо загрузить в память всё дерево тегов.

Преимуществом SAX-технологии является, наоборот, использование относительно небольшого объёма оперативной памяти. Платой за это являются определённые неудобства при программной реализации правил обработки xml-документов. Решение по обработке тега и его содержимого необходимо принимать почти вслепую, поскольку неизвестно, что было вложено в тег ранее и будет вложено дальше. Невозможно также обобщать правила обработки xml-документов; кроме того, такие правила программируются на языке, не приспособленном для обработки тегов с их атрибутами.

Реализация разработанных алгоритмов была осуществлена с использованием как DOM-обработчика, так и SAX- обработчика.

Для реализации в рамках DOM-технологии использовался язык преобразования xml-документов XSLT; правила преобразования записывались в файл стилей styles.xsl (***рис. 2***).

[1] Под DOM понимается технология, при которой для обработки документа требуется его полная загрузка в оперативную память.

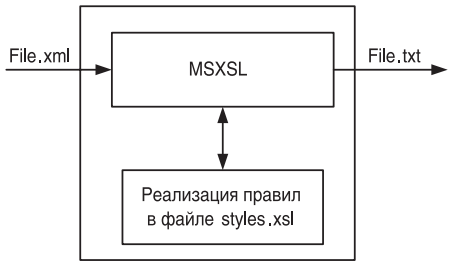

Рис. 2. Схема обработки xml-документа с использованием XSLT

Для выбора данных из исходного дерева тегов формировались запросы на языке запросов XPath. Применение языка XSLT осуществлялось с использованием процессора Microsoft XML Parser 4.0 [4]. Он принимает на вход исходное дерево тегов (xml-файл), выходными данными является файл как результат применения правил-стилей XSLT к xml-файлу. Обработка каждого узла данным процессором осуществляется применением к нему всех правил, шаблону условия которых удовлетворяет данный узел. Вычислительная сложность обработки составляет $O(n*m)$, где n — количество узлов, m — количество правил [5].

Реализация разработанных алгоритмов с использованием SAX-технологии заключается фактически в написании кода, составляющего методы StartTag, StopTag, TagContent. Такой код был реализован путём создания классов на C++, содержащих методы обработки различных сложноструктурированных объектов. Программа читает входной xml-документ порциями около 200 байт и посылает информацию на обработку (***рис. 3***). Обработчик идентифицирует в каждой поступившей порции данных открывающие и закрывающие теги. Для распознанных тегов в блоке идентификации объектов осуществляется поиск начала и завершения конкретного сложноструктурированного объекта. Каждый распознанный объект записывается в кеш-память, после чего вызывается обработчик данного объекта; результат преобразования объекта подаётся на выход.

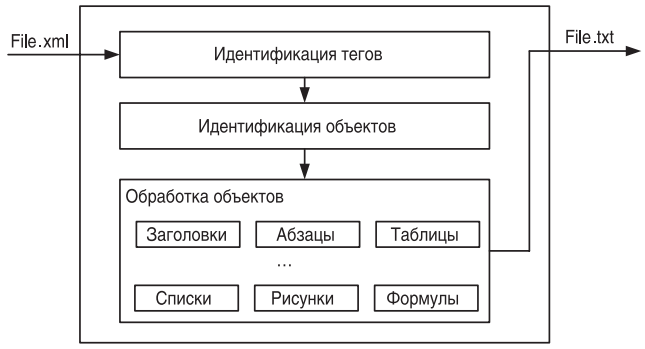

Рис. 3. Схема обработки xml-документа с использованием SAX

Оба обработчика тестировались на нескольких xml-документах. Результаты тестирования представлены в таблице 7, где отражено количество сложноструктурированных объектов, обрабатываемых в соответствии с описанными ранее алгоритмами, а также используемый объём памяти и скорость обработки входных документов на компьютере, имеющем процессор Intel® Core(TM)2 с тактовой частотой 2x1.80 гГц.

Таблица 7

Результаты тестирования программных модулей обработки xml-документов с использованием XSLT- и SAX-технологий

Наименование xml-документа	Объём документа, кБ	Кол-во сложностр. объектов	Объём памяти, мБ (XSLT/ SAX)	Скорость обработки, с (XSLT/ SAX)
Компьютерный синтез и клонирование.xml	1 394	27 131	8 / 1,5	0,6 / 1,33
Финансовые таблицы.xml	20 498	382 541	70 / 1,5	600 / 15

Как видно из таблицы, при применении SAX-технологии требуется значительно меньший объём памяти для обработки xml-документов, чем при применении XSLT-технологии. Этот факт объясняется необходимостью загрузки в память всего документа в программном модуле, использующем XSLT-технологию; при использовании SAX-технологии требуемый объём памяти определяется размером максимального сложноструктурированного объекта, входящего в обрабатываемый документ.

Скорость обработки документов при использовании XSLT-технологии резко уменьшается при увеличении объёма документа и количества сложноструктурированных объектов в нём. При использовании SAX-технологии обработка xml-документов, бо̀льших по объёму и количеству сложноструктурированных объектов, влечёт незначительное увеличение скорости обработки.

Заключение

Предлагаемые в работе алгоритмы программно реализованы и используются в составе системы синтеза речи MultiPhone.

Достоинством алгоритмов является то, что они направлены на преобразование сложноструктурированных объектов в орфографический текст с учётом адекватности смыслового восприятия полученного текста. Предложены несколько режимов обработки сложноструктурированных объектов, классифицированные по степени подробности преобразования.

Программная реализация разработанных алгоритмов осуществлена с использованием двух различных технологий, что позволит использовать программные модули как для мобильных устройств, характеризующихся малым объёмом памяти и низким быстродействием, так и для приложений, существенной характеристикой которых является расширяемость и масштабируемость.

Двухэтапная обработка сложноструктурированных объектов, включающая преобразование к xml-документу на первом этапе и обработку сложноструктурированных объектов в составе xml-документа на втором этапе, в котором реализованы разработанные алгоритмы, позволяет осуществить преобразование сложноструктурированных объектов в других форматах путём добавления модулей формирования xml-документа на основе, например, pdf-файлов.

За рамками данного исследования осталась разработка тестов для оценки адекватности смыслового восприятия сформированного текста и соответствующее тестирование предлагаемых алгоритмов. На решение данных задач будут направлены дальнейшие усилия авторов.

Литература

1. *Лобанов Б.М., Цирульник Л.И.* «Компьютерный синтез и клонирование речи», Минск: Белорусская наука, 2008. — 342 с.
2. http://ru.wikipedia.org/wiki/DOM
3. http://ru.wikipedia.org/wiki/SAX
4. http://www.w3.org/TR/xslt
5. http://www.zdnet.de/security/news/0,39029460,39198860,00.htm

Цирульник Лилия Исааковна —

Окончила факультет прикладной математики и информатики Белорусского государственного университета. Кандидат технических наук, старший научный сотрудник лаборатории распознавания и синтеза речи Объединённого института проблем информатики Национальной академии наук Беларуси, автор более 40 научных работ по проблемам компьютерного синтеза и клонирования речи. Область научных интересов — методы автоматического анализа и синтеза речевых сигналов, человеко-машинные системы речевого общения, речевые компьютерные технологии.

Гецевич Юрий Станиславович —

Окончил факультет прикладной математики и информатики Белорусского государственного университета, факультет математики и информатики университета в Мангейме (Германия). Аспирант, младший научный сотрудник лаборатории распознавания и синтеза речи Объединённого института проблем информатики Национальной академии наук Беларуси. Область научных интересов — методы синтеза белорусской и русской речи по тексту, человеко-машинные системы речевого общения, речевые компьютерные технологии.