УДК 681.3

РАЗРАБОТКА УНИФИЦИРОВАННОГО МЕТОДА РАЗБОРА СТРУКТУР ЭЛЕКТРОННЫХ ДОКУМЕНТОВ, ПРЕДСТАВЛЕННЫХ ОПИСАТЕЛЬНЫМИ ЯЗЫКАМИ РАЗМЕТКИ

В.М. Левыкин, Е.А Моспан.
Харьковский национальный университет радиоэлектроники.

В статье исследована возможность использования описательных языков разметки в качестве формата документов-шаблонов в технологиях формирования электронных документов в информационных системах. Разработан метод разбора структур электронных документов, форматом которых является произвольный описательный язык разметки. Метод позволяет получить объектную иерархическую модель структуры документа на основании её формального представления, что обеспечивает возможность использования описательных языков разметки в качестве форматов документов-шаблонов в технологиях формирования электронных документов.

Ключевые слова: *модель, структура электронного документа, описательный язык разметки, тег, объект, метод разбора, парсер.*

Введение

Технологии формирования электронных документов применяют для представления пользователю результатов реализации функциональных задач в информационных системах (ИС) [1]. Для формирования электронных документов в этих технологиях применяют документы-шаблоны, форматом которых являются описательные языки разметки. Посредством таких языков структуру электронного документа можно представить двумя способами: в виде формальной записи в текстовом файле и объектной иерархической моделью [2]. При этом процесс перехода от формального представления к объектной иерархической модели называют разбором структуры электронного документа. Представление структуры документа-шаблона в виде такой объектной иерархической модели обеспечивает возможность ее изменения на основании данных функциональной задачи, в рамках которой формируется этот документ. В работе [3] рассмотрена технология формирования электронных документов, в которой в качестве формата документа-шаблона используется произвольный язык разметки. Однако на сегодняшний день не существует унифицированного метода разбора структур электронных документов, форматом которых являются описательные языки разметки, для получения объектной иерархической модели.

Разнообразие описательных языков разметки предопределяет разнородность методов разбора их структур, представленных формальным способом. Использование однотипных методов разбора описательных языков разметки возможно только при условии вхождения их в единое семейство. Так разметки структуры электронных документов, форматом которых являются XML-совместимые описательные языки (например, XHTML, OpenDocument и другие), могут быть разобраны, так называемыми, XML-парсерами, что позволяет получить объектную модель структуры документа, которая соответствует Document Object Model (DOM) [4]. Однако существуют такие описательные языки разметки, для которых XML-парсеры неприменимы, например, HTML, RTF, DOC и другие. Поэтому для разбора структур электронных документов, представленных в формате таких языков разметки, необходимо применять другие методы разбора. Это обуславливает сложность использования различных описательных языков разметки в качестве форматов документов-шаблонов в современных технологиях формирования электронных документов в информационных системах.

Представим некоторый электронный документ в формальном виде в формате RTF:

{\rtf1\ansi{\fonttbl\f0\fswiss Helvetica;}\f0\pard This is some {\b bold} text.\par}

В тестовом процессоре, поддерживающим данный описательный язык разметки, такой документ будет выглядеть следующим образом:

This is some **bold** text.

Структуру данного элеткронного документа представим в виде следующего набора тегов (свойств объектов):

("{", "\rtf1", "\ansi", "}", "\fonttbl", "\f0", "\fswiss", "Helvetica;", "}", "\f0", "\pard", "This is some ", "{", "\b", "bold", "text.", "\par", "}").

На основании анализа подобных наборов тегов существующие методы разбора структур электронных документов, форматом которых являются описательные языки разметки, формируют объектную иерархическую модель структуры документа на основе формального способа её записи. Объекты, которые входят в состав такой модели, а также их свойства, полностью соответствуют спецификации описательного языка разметки. В связи с различной формой записи формального способа представления структуры электронного документа в существующих языках разметки описательного типа для получения иерархической объектной модели используют узкоспециализированные парсеры. Так, для языков семейства XML используют так называемые XML-парсеры, основными реализациями которых являются SAX и Xerxes. Результатом работы этих парсеров является DOM-модель документа. Для разбора структур электронных документов, представленных описательными языками разметки, которые не являются реализацией стандарта XML, используют API текстовых процессоров, которые их поддерживают.

Постановка задачи

На основании проведенного анализа можно утверждать, что использование существующих методов разбора структур электронных документов, форматом которых являются описательные языки разметки, для формирования объектной иерархической модели их структуры невозможно в рамках технологии, рассмотренной в работе [3]. Существующие на сегодняшний день методы позволяют получать иерархическую объектную модель структуры электронного документа, форматом которого является ограниченный набор описательных языков разметки. Но эти ограничения противоречат возможности использования в рамках технологии формирования электронных документов произвольного описательного языка разметки в качестве формата документа-шаблона. В связи с этим разработка унифицированного метода разбора структур электронных документов, формат которых относится к описательным языкам разметки, является актуальной задачей. Метод должен удовлетворять следующим требованиям:

- в результате разбора структуры электронного документа, должна быть получена объектная иерархическая модель его структуры;
- метод должен обеспечивать разбор структуры электронного документа, форматом которого является описательный язык разметки;
- формирование объектной иерархической модели структуры электронного должно осуществляться на основании ее формального представления в контексте произвольного описательного языка разметки.

Основной материал и результаты

Унифицированный метод разбора структур электронных документов представим в виде следующих этапов:

Этап 1. Формирование метаинформации об описательном языке разметки S.

Этап 2. Формирование множества тегов (свойств объектов), определяющих структуру конкретного электронного документа, на основании её формализованного представления в контексте описательного языка разметки S.

Этап 3. Формирование метаинформации об объектах описательного языка разметки S, введенных на 1-ом этапе.

Этап 4. Формирование структуры электронного документа в контексте описательного языка разметки S.

Первый этап унифицированного метода разбора структур электронных документов включает в себя формирование метаинформации об описательном языке разметки S. Метаинформация представляется в виде множества объектов и их свойств, а также матрицы, отображающей соответствие между ними.

Множество объектов (E_S) состоит из объектов E_i, состав которых определяется спецификацией описательного языка разметки S:

$$E_S = < E_1, E_2, ..., E_i, ..., E_{n-1}, E_n >, i = \overline{1,n}, \quad (1)$$

где n – количество объектов в спецификации описательного языка разметки S.

Помимо множества объектов E_S спецификация описательного языка разметки также определяет набор возможных свойств, которые могут принадлежать данным объектам. Под свойством объекта следует понимать некоторую характеристику, которая определяет поведение рассматриваемого объекта. Одним и тем же свойством могут обладать разные объекты. Введем множество всех свойств объектов (P_S):

$$P_S =< P_1, P_2, ..., P_j, ..., P_{m-1}, P_m >, j = \overline{1,m}. \quad (2)$$

где m – количество свойств объектов в спецификации описательного языка разметки S.

Свойство P_j представляет собой упорядоченную пару ключ-значение:

$$P_j =< K, V >, \quad (3)$$

где P_j – некоторый элемент множества P_S; K – ключ (идентификатор) свойства; V – значение свойства.

Для того, чтобы описать соответствие принадлежности между объектами и свойствами описательного языка разметки, введем матрицу соответствия (EP), размерностью nxm. Элементы этой матрицы могут принимать значения {0,1}. 1 – свойство P_j принадлежит объекту E_i; 0 – не принадлежит:

$$EP = \begin{vmatrix} EP_{11} & EP_{12} & \dots & EP_{n1} \\ \dots & \dots & \dots & \dots \\ EP_{1j} & \dots & EP_{ij} & EP_{nj} \\ \dots & \dots & \dots & \dots \\ EP_{1m} & EP_{2m} & \dots & EP_{nm} \end{vmatrix}. \quad (4)$$

2-й этап предполагает формирование множества тегов, которые формируют структуру электронного документа, на основании её формального представления. В формальном виде структура электронного документа, форматом которого является описательный язык разметки, представляет собой набор тегов (свойств объектов), которые входят во множество P_S. Данная последовательность определяет иерархическую структуру объектов документа. Представим данный набор тегов в виде множества (T_S), которое является входным множеством для третьего этапа унифицированного метода разбора структур документов, представленных языками разметки описательного типа:

$$T_S =< T_1, T_2, ..., T_k, ..., T_{q-1}, T_q >, k = \overline{1,q}, \quad (5)$$

где q – количество тегов в наборе, который определяет структуру электронного документа.

Определим основные свойства множества T_k:

Свойство 1. Любой элемент T_k множества T_S принадлежит множеству P_S:

$$\forall T_k \in P_s. \quad (6)$$

Свойство 2. Любой элемент T_k множества T_S является свойством некоторого объект E_i и, исходя из свойства 1, элементом P_j множества P_S, следовательно:

$$\forall T_k \exists EP_{ij} = 1. \quad (7)$$

3-й этап предполагает формирование метаинформации об объектах множества E_S, а, именно, правил построения иерархической структуры объектов документа на основании полного перебора всех элементов множества T_S. Подобную структуру можно представить в виде следующей модели M:

$$M = (\{O_i\}, \{C_{ikj}\}), i = \overline{1,n}, j = \overline{1..m}, \; k = \overline{1,n-1}, \quad (8)$$

где O_i – объект структуры электронного документа; C_{ilj} – отношение между объектами структуры электронного документа O_i и O_k (k<n), n – количество объектов в иерархической структуре электронного документа, m - .количество связей между ними.

Определим возможные отношения между объектами структуры документа:

Отношение 1. Объект O_i является контейнером объекта O_k.

$$O_k \in CO_{O_i}, \tag{9}$$

где CO_{O_i} – множество всех объектов, для которых O_i является контейнером.

Отношение 2. Объект O_i является дочерним элементом объекта O_k. Данное отношение является обратным первому.

Отношение 3. Объект O_i следует за объектом O_{i-1} в рамках общего объекта-контейнера O_k.

$$O_{i-1} \in CO_{O_k}, O_i \in CO_{O_k}, \tag{10}$$

где CO_{O_k} – множество всех объектов, для которых O_k является контейнером.

В состав структуры документа могут входить несколько объектов одного типа (например, несколько абзацев, таблиц и прочее), а одно свойство может принадлежать нескольким типам объектов. Следовательно, в процессе анализа множества T_S необходимо определять, к какому объекту относится рассматриваемое свойство. С другой стороны, необходимо также обозначить момент окончания заполнения свойствами объекта и определения его отношений с другими объектами структуры документа. Исходя из выше сказанного, можно сделать вывод о том, что некоторое подмножество элементов T_S' множества T_S соответствует некоторому объекту E_i, который входит в модель структуры документа. Следовательно, структуру документа можно рассматривать как совокупность некоторых подмножеств T_s', определяющих состав её объектов и отношений между ними.

Введем правило, которое позволит определять границы подмножества T_S', соответствующему одному объекту структуры документа. В общем случае, границы подмножества T_S' можно определить на основании некоторого открывающего и соответствующему ему закрывающего тега. Такое утверждение актуально для описательных языков, входящих в семейство SGML (например, HTML), однако для некоторых языков (таких как RTF) подобного правила недостаточно. В связи с этим введем обобщенное правило определения границ подмножества T_s'. Открывающий тег будем далее называть идентификатором создания объекта (I_c), а закрывающий тег - идентификатором завершения инициализации объекта (I_d). В процессе перебора всех элементов множества T_S среди них будут встречаться элементы, которые соответствуют I_c и I_d некоторых объектов структуры документа. Между парой идентификаторов I_c и I_d одного объекта могут находиться идентификаторы I_c и I_d других объектов. Такая особенность обуславливает иерархию объектов в структуре электронного документа. Множество T_s' для объекта, созданного на основании I_c, определим следующим образом:

$$T_s' = \begin{cases} < T_{Ic} ... T_{Id} >, \omega_E = \varnothing \\ < T_{Ic}, ..., T_1'', T_2'', ... T_i'', ..., T_{p-1}'', T_p'', ..., T_{Id} > \notin T_{de}, \omega_E <> \varnothing, i = \overline{1, p} \end{cases} \tag{11}$$

где T_{Ic} – элемент множества T_S, который соответствует идентификатору I_c; T_{Id} – элемент множества T_S, который соответствует идентификатору I_d; T_{de} – элементы множества T_S, которые относятся к дочерним объектам элемента структуры электронного документа, соответствующего множеству T_s':

$$T_{de} = < T_1'', T_2'', ... T_i'', ..., T_{p-1}'', T_p'' >, i = \overline{1, p}, \tag{12}$$

p – количество тегов, которые к дочерним объектам; ω_E – множество всех объектов, для которых созданный объект является контейнером.

Объект, который создается на основании некоторого I_c, будем называть текущим и обозначим *CE* (current element). А объект, который выступает контейнером для *CE*, будем называть родительским объектом *PE* (parent element). Введем расширенную функцию, которая

позволяет определить какой объект должен быть создан на основании некоторого идентификатора I_c, а также CE и PE:

$$f_{I_c} = \varphi(T_k, CE, PE), \quad (13)$$

где f_{I_c} – функция, которая определяет условия создания объекта структуры документа, возможные значения 0 (объект не может быть создан) или 1 (объект должен быть создан); T_k – рассматриваемый элемент множества T_S.

Введем функцию f_{I_d}, которая определяет условия завершения инициализации текущего объекта:

$$f_{I_d} = \phi(T_k, CE, PE), \quad (14)$$

её возможные значения 0 (инициализация объекта продолжается) или 1 (инициализация объекта должна быть завершена); T_k – рассматриваемый элемент множества T_S.

Каждый объект множества E_S обладает множеством функций f_{I_c} и f_{I_d}, которые предопределяют условия его создания и завершения инициализации, а также множество элементов T_s', на основании которого определяются свойства объекта. Соответственно, объект E_i можно представить в виде следующего кортежа:

$$E_i = < P', F_{I_c}, F_{I_d} >, \quad (15)$$

где P' – множество всех свойств, которыми обладает объект E_i, данное множество можно определить на основании матрицы EP; F_{I_c} – множество функций f_{I_c}; F_{I_d} – множество функций f_{I_d}.

4-ый этап предполагает формирование структуры документа в контексте описательного языка разметки S. Сформулируем правила (П) формирования структуры электронного документа на основании множества T_S, зафиксировав некоторый элемент T_k, текущий объект CE и родительский объект PE:

П$_1$: Если $T_k \in P'$, тогда элемент T_k является свойством CE.

П$_2$: Если $T_k \notin P'$, тогда необходимо выполнить функции множества F_{I_c} всех объектов множества E_S. Если результат выполнения одной из функций будет равен 1, то это свидетельствует о создании соответствующего объекта. В этом случае текущий объект CE становится родительским PE, а вновь созданный объект текущим CE.

П$_3$: Если в процессе первых двух этапов не удалось определить, принадлежность свойства, то необходимо выполнить функции множества F_{I_d} текущего объекта CE. Если результат выполнения одной из функций будет равен 1, то это свидетельствует о завершении инициализации текущего объекта CE и добавлении его в родительский объект.

В общем виде эти правила представим следующим образом:

$$f(T_k) = \begin{cases} \varphi(CE, T_k), T_k \in P'_{CE} \\ PE = CE;\ CE = E_j, T_k \notin P'_{CE}\ и\ f_{I_c} = 1,\ причем\ f_{I_c} \in F_{I_c} \in E_j, \\ \phi(CE, PE), f_{I_c} = 1,\ причем\ f_{I_d} \in F_{I_d} \in CE \end{cases} \quad (16)$$

где $f(T_k)$ – действия, выполняемые над каждым элементом множества T_S; T_k – один из элементов множества T_S; $\varphi(CE, T_k)$ – функция, посредством которой осуществляется установка свойства T_k в объект CE; P'_{CE} – множества всех свойств объекта CE; f_{I_c} – функция, которая определяет условия создания объекта структуры документа; F_{I_c} – множество функций f_{I_c}, которые принадлежат объекту E_j; $\phi(CE, PE)$ – функция, посредством которой осуществляется добавление текущего объекта CE к родительскому объекту PE; f_{I_d} – функция, которая

определяет условия завершения инициализации объекта структуры документа; F_{I_d} – множество функций f_{I_d}, которые принадлежат объекту E_j.

Таким образом, этапы унифицированного метода разбора структур электронных документов, представленных описательными языками разметки, с учетом введенных обозначений имеют следующий вид:

Этап 1. Формирование множеств E_S, P_S, а также матрицы EP для описательного языка разметки S.

Этап 2. Формирование множества T_S на основании формализованного представления структуры электронного документа в контексте описательного языка разметки S.

Этап 3. Описание множеств F_{I_c} и F_{I_d} для всех объектов множества E_S.

Этап 4. Формирование структуры электронного документа в контексте описательного языка разметки S, на основании (11), (16).

Выводы

Разработанный унифицированный метод разбора структур документов, форматом которых являются языки разметки описательного типа, позволяет представлять структуры документов *в виде объектной иерархической модели, вне зависимости от их формата. Это обеспечивает* возможность использования произвольных языков разметки описательного типа в качестве формата документа-шаблона в технологиях формирования электронных документов. Полученная объектная иерархическая модель представления структуры электронного документа позволяет определить правила изменения исходного документа-шаблона с целью наполнения его актуальными данными функциональной задачи информационной системы, в рамках которой формируется выходной документ.

Литература:

1. Чалый, С. Ф. Разработка модели модифицированной технологии формирования электронных документов на основании шаблонов в WEB-ориентированных информационных системах / С. Ф. Чалый, Д. Л. Кравченко, Е. А. Моспан. // Сборник научных трудов Харьковского университета воздушных сил. – 2008. – Вып. 3 (18). – С. 135-138.

2. Markup Languages: Theory and Practice: Journal. – Cambridge: MIT Press, 1999. – 120 p.

3. Левыкин, В. М., Моспан, Е. А. Разработка модели формирования электронных документов в WEB-ориентированных информационных системах / В. М. Левыкин, Е. А. Моспан. // АСУ и приборы автоматики. – 2008. – Вып. 144. – С. 54-58.

4. Harold Elliotte Rusty XML bible. – Michigan: Hungry Minds, 2001. – 1565 p.

Поступила в редакцию 17.07.2009.

Левыкин Виктор Макарович, доктор технических наук, профессор. Харьковский национальный университет радиоэлектроники, заведующий кафедрой информационных управляющих систем. Тел.: (057) 702-14-51.

Моспан Евгений Александрович, начальник отдела Java ООО «ПрофИТсофт». Тел.: (097) 457 84 01; e-mail: eugene.mospan@mail.ru