УДК 681.39

В.А. Антипов, Р.Е. Гузенко

ТРЁХУРОВНЕВАЯ МЕТАМОДЕЛЬ ОТОБРАЖЕНИЯ СЕМАНТИКИ ПРЕДМЕТНОЙ ОБЛАСТИ НА СТРУКТУРУ XML СООБЩЕНИЙ

Рассматриваются теоретические основы моделирования XML структур и формальный метод отображения на них семантики предметной области.

Ключевые слова: *XML структуры, моделирование, типы как средство отображения понятий предметной области, математический формализм XML структур.*

Введение. Моделирование семантики домена представляет очень важную задачу, так как сосредоточивается на восприятии данных в отличие от физического представления. Область исследований, связанная с семантическими моделями данных, в последние годы развивалась большими темпами [1, 2, 3, 4, 5]. С появлением и распространением XML схем как широко используемого и стандартизованного языка описания данных эта область приобретает ещё большее значение, поскольку увеличивается число сфер, в которых используются XML схемы. Медико-биологические науки – одна из таких сфер. В связи с этим всё более важным становится обеспечение формального определения XML структур с хорошо разработанным способом отображения на них семантики предметной области. С применением XML инфраструктуры для взаимодействия между объектами домена возникают новые требования к процессу разработки сообщений.

Чтобы гарантировать надёжность XML данных, которые служат как входные данные приложений, должна быть доказана правильность схем. Алгебраические спецификации, основанные на алгебраических уравнениях, являются формальным средством определения действительности (верности) схем. Обмен данными между различными системами требует преобразования XML структур. В свою очередь должны быть доказана эквивалентность и корректность этих преобразований. Эти преобразования также необходимы как строительные блоки для методов проектирования в процессе разработки основанных на XML моделях данных для различных предметных областей. С другой стороны, это требует соответствующей алгебры, подобной алгебре реляционных баз данных (РБД), чтобы доказать правильность и законченность проведённых преобразований и проекта в целом. Подобно РБД, где реляционная алгебра служит основанием для методов проектирования и формального доказательства, соответствующая алгебра для XML позволит формально специфицировать схему и осуществить проверку (синтаксический анализ). В рамках этого подхода решаются следующие задачи:

- разработка теоретических основ для моделирования XML структур;
- разработка математического формализма для доказательства правильности XML структур;
- разработка формального метода отображения семантики предметной области на XML структуры;
- определение того, как полученные результаты удовлетворяют построению XML алгебры и какой вклад они вносят в область формализации разработки XML приложений.

Цель работы разработать метамодель отображения семантики на структуру XML сообщений.

Ключевые концепции

1. Моделирование XML структур как направленных графов. XML моделирует документы и структуры данных как композицию элементов. Базовая структура XML документа представляется иерархически вложенными документами, поэтому естественным подходом к моделированию XML документа является древовидный граф. Это есть подход W3C (World Wide Web Consortium) в его спецификации DOM (Document Object Model). DOM определяет набор интерфейсов, использующих OMG CORBA IDL [6] для моделирования XML документа как граф-дерева, где рёбра графа отражают отношение предок - потомок между элементами. Несколько механизмов связывания, таких как Идентификатор/Ссылка (ID/IDREF), язык связывания XML (XML Linking Language), расширяют

базовую структуру дерева XML документа для выражения более сложных отношений между XML элементами. Для того чтобы обеспечить интеграцию механизмов связывания, предлагается моделировать XML структуры как направленные графы, а не как графы дерева.

2. Типы как средство отображения понятий предметной области. Применение понятий предметной области к XML структурам является ключевой проблемой. Исследовательские работы в области семантических сетей определяют некоторые идеи решения этой проблемы. Гипертекстовые системы, которые фактически являются порождением XML , могут рассматриваться как семантические сети [7, 8]. Семантическая сеть – это направленный граф, в котором понятия представлены как узлы, а отношения между понятиями представлены как связи. Используя этот подход, введем типы на узлах и связях направленного графа, соответственно отображая понятия предметной области и отношения между понятиями предметной области на типах узлов и связей.

3. Иерархии типов как средство абстракции. Процесс моделирования предметной области требует возможности представления и описания модели домена на различных уровнях абстракции. Чтобы обеспечить эту возможность, вводим иерархии на типы узлов и типы связей.

4. Выражение знаний предметной области через структурные ограничения. Понятия предметной области и отношения между ними отображаются на типы узлов и типы связей. Чтобы выразить знания предметной области и правила, определим структурные ограничения на эти типы.

5. Трёхуровневая метамодель. Последовательное разделение структуры и содержания вместе с определением типа структурных ограничений предписывает введение трёхуровневой метамодели.

Определение графа типа, основанное на теории множеств. Введём определение направленного графа с типами узлов и типами связей, основанное на теории множеств. Граф называется графом типа и обозначается как TG.

Ориентированные графы и составные связи. Направленный граф DG состоит из множества вершин (узлов) и из множества рёбер, которые соединяют узлы: $DG = (N, E)$ с N конечным и непустым множеством узлов $N = (n_1, n_2, n_3, \ldots, n_n)$ и $E \subseteq N \times N$ подмножеством Декартова произведения $N \times N$, содержащего рёбра графа. Направленный граф имеет только ориентированные рёбра, соединяющие начальный узел (или узел-источник) и конечный узел (или узел цели). Путь через граф определяется как связанная, конечная последовательность рёбер.

Введём термин *связь* для абстрактного выражения соединения между двумя узлами направленного графа (рисунок 1). В общем случае такая связь выражает путь через граф (в отличие от других публикаций, где *связь* используют как синоним ребра).

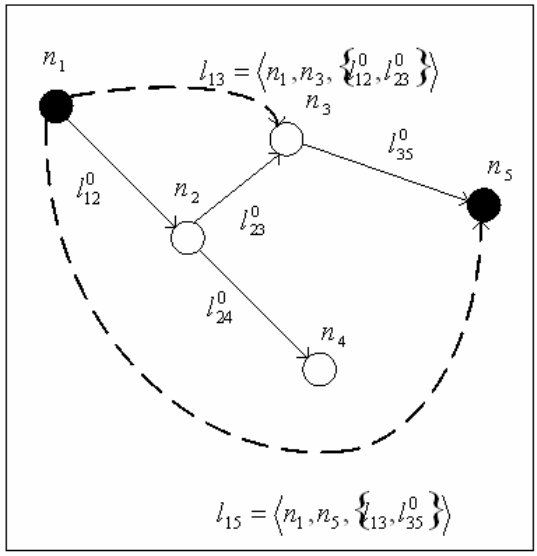

Рисунок 1

Связь определяется в терминах её начальной вершины, её конечной вершины и возможно пустой последовательности соединений промежуточных связей. В случае пустой последовательности промежуточных связей связь представляет собой ребро и называется *прямой связью*. Это позволяет рекурсивно определить связь через *прямые связи*. В противопоставлении общему двухуровневому подходу, который делает различие между дугами и путями, предлагаемый подход составных связей является более мощным и служит важным средством абстракции.

Переопределим направленный граф, который будет выражаться в терминах узлов и связей: $DG = (N, L)$, где $L \subseteq N_S \times N_E \times LS$ является множеством связей l с $N_S \subseteq N$ множеством начальных узлов; $N_E \subseteq N$ – множеством конечных узлов и LS – множеством последовательностей соединяющих связей.

Множество последовательностей соединяющих связей определяется следующим образом: $LS \subseteq \prod_{n \in \underline{N}} L_n$, при этом

$$LS = \{(l_1, l_2, l_3, \ldots, l_k) \mid k \geq 0 \wedge \forall i, l \leq i(k\text{-}1): t(l_i) = s(l_{i+1})\}.$$

Для каждых двух последующих связей в последовательности конечный узел первой связи является начальным узлом второй. $t(l)$ и $s(l)$ – две функции, определённые на связях l так, что $s(l) = n_s$ и $t(l)= n_e$, где s-отображение связи на её начальный узел n_s и t-отображение связи на её конечный узел n_e. Последовательность соединяющих связей (элемент LS) обозначается ls и ls^0 для специального случая пустой последовательности связей. Тогда полное рекурсивное определение связей $L \subseteq N_S \times N_E \times LS$

$$L=\{\langle n_s, n_e, ls\rangle \mid ls = ls^0 \vee \forall ls \neq ls^0 : s(ls)= n_s \wedge \wedge t(ls)= n_e\},$$

где $s(ls)$, $t(ls)$ - две функции последовательности связи ls такие, что $s(ls)$ отображает последовательность связи на её начальную вершину, а $t(ls)$ отображает последовательность связи на её конечную вершину. Всякий раз, когда необходимо сделать различие между связью, представляющей путь через граф, и связью, представляющей ребро графа, обозначаем последнее как l^0, что является специальным случаем связи $\langle n_s, n_e, ls^0\rangle$. Связь l^0 называется *прямой связью.*

Введение типов узлов и типов связей. В качестве цели было заявлено применение семантики домена предметной области к XML структурам. Мотивация моделирования XML структур направленным графом связана с тем, что это наиболее удачный подход для объединения различных механизмов связи. Интерпретация XML непосредственно как домена предметной области требует механизма отображения предметной области на направленный граф. Предлагаемый подход заключается в том, что вводятся типы на узлы и связи направленного графа таким образом, чтобы понятия предметной области отобразились на типах узлов, а отношения между понятиями домена отобразились на типах связей.

Вводятся множество типов узлов NT и множество типов связей LT. Элементы NT обозначаются t_{node}, а элементы LT обозначаются t_{link}. Вводятся обозначения: tn – узел типа и tl – связь типа. Множество узлов типов записывается как TN и определяется как отображение узлов на типы узлов $TN: N \rightarrow NT$. Множество связей типов TL определяется как отображение связей на типы связей TL: $L \rightarrow LT$.

Элемент из TN определяется как

$$tn = (n, t_{node})\ n \in N \wedge t_{node} \in NT.$$

Элемент из TL определяется как

$$tl = (tn_i, tn_j, tls, t_{link})\ tn_i, tn_j \in TN \wedge t_{link} \in LT.$$

Типы связей соединяют тип узла как начальный узел с другим типом узла как конечный узел через, возможно, пустую последовательность промежуточных типов связей. Тип элемента (узел-тип или связь-тип) называется *примером* его типа. Дополнительно определяются функция $value(tn)$ на узлах типов, которая отображает узел-тип на его значение узла $n{:}value(\langle n, t_{node}\rangle) = n$ и функция $type(tn)$ на узлах типов, которая отображает узел типа на его тип узла t_{node}: $type(\langle n, t_{node}\rangle) = t_{node}$.

Функция $type(tl)$ отображает связь типа на её тип связи t_{link}:

$$type(\langle tn_i, tn_j, tls, t_{link}\rangle) = t_{link}.$$

Введём понятие домена как множества возможных значений узлов типов. Домен определён на основе типа и обозначается $dom(type(tn))$ – домен типа узла такой, что если $tn = (n, t_{node})$ является типом узла, то $n \in dom(type(tn))$ или $n \in dom(t_{node})$. Тогда формально можно выразить

$$value(tn) \in dom(type(tn)).$$

Кортеж $GT=(NT, LT)$ называется графом типа.

Введём нотацию типового графа как $TG = (TN, TL, dom(TG))$, тогда домен TG обозначается как

$$dom(TG) = \bigcup_{tn \in TN} dom(type(tn)).$$

TG состоит из 3-х элементов: множества узлов типа, множества связей типа и множества $dom(TG)$, которое называется доменом типового графа.

Разработка трёхуровневой метамодели. Возможность выражения знаний предметной области является критичной для качества модели предметной области. Предлагаемый подход к выражению знаний о предметной области основан на определении ограничений на типах TG.

Проведём последовательное разделение типа и примера с целью разделения структуры и содержания, что, в свою очередь, приведёт к построению трёхуровневой метамодели.

Разделение типа и примера. Определим два уровня, на которых специфицируем элементы, множества и отношения. Один уровень назовём *тип-уровень*, другой — *пример-уровень*. Элементы, определённые на *тип-уровне,* называются типами, которые в нашем случае являются типами связей, типами узлов и графами типов. Элементы, определённые на *пример-уровне,* называются примерами (или типами элементов) и в нашем случае являются связями типа, узлами типа и типовыми графами. На рисунке 2 показан рассматриваемый двухуровневый подход.

Согласно свойствам отображения для одного типа может существовать множество примеров, но каждый пример должен иметь один и только один тип (для обозначения этого свойства будем использовать термин «*есть_частный случай*»).

Определение иерархий типа. Средством отображения предметной области на ориентированный граф являются типы (понятия домена отображаются на типы узлов, отношения между понятиями домена отображаются на типы связей).

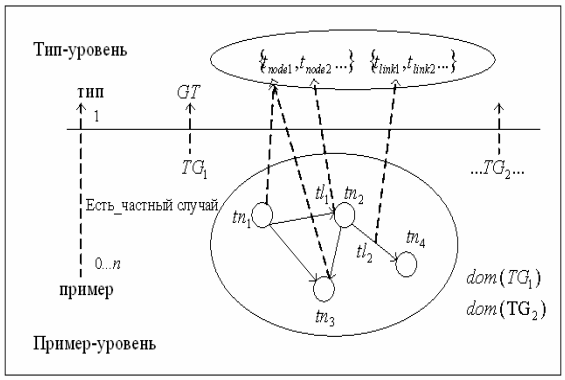

Рисунок 2

Процесс моделирования домена требует возможности описывать и представлять эти домены различными уровнями абстракции. Чтобы обеспечить различные уровни абстракции, введём иерархию типов.

Определим направленный граф, назовём его графом иерархии типов и обозначим как T_H:

$$T_H = (T, L_T),$$

где T является множеством типов, L_T является множеством *прямых связей* (рёбер), определяющих отношения между типами

$$L_T = \{\langle t_i, t_j, l_T^s\rangle \mid t_i, t_j \in T \wedge t_i \neq t_j \wedge l_T^s = l^0{}_T^s\}.$$

Для того чтобы сделать T_H иерархическим графом, необходимо применить следующие ограничения

$$|\Gamma^{-}(t_i)| = 0\ldots l, \text{ где } \Gamma^{-}(t_i) = \{m \mid \exists \langle m, t_i, l^0{}_T^s\rangle \in L_T\},$$

которые формально определяют то, что тип может иметь больше одного супертипа. Тип, не имеющий супертипа, называется корневым типом. Кроме того, определим

$$\forall t \in T : t \notin D(t),\quad D(\,) = \{m \mid \exists l_T^s : s(l_T^s) = t \wedge t: :(l_T^s) = m\},$$

где l_T^s –последовательность соединяющих связей $l_T \in L_T$. $D(t)$ является набором потомков t и следовательно определяет множество всех подтипов. Это означает, что граф иерархии типов является ациклическим.

Иерархия типов позволяет характеризовать зависимости типов и описывать структурные ограничения, представлять и трактовать различные уровни абстракции через средства специализации и обобщения. На *тип-уровне* тип будет специализацией своего супертипа и обобщением своего субтипа (подтипа). На *пример-уровне* в любом месте, где пример определённого типа может быть использован, пример любого подтипа этого типа может быть также использован. Это в подобной форме известно как полифизм в области объектно-ориентированного проектирования.

Эти понятия имеют важный смысл. На *тип-уровне* типы определяют особенности и структурные ограничения, которые будут применены к примерам этого типа. Примеры (типы элементов) должны удовлетворять не только ограничениям, определённым посредством их специфического типа, но также и ограничениям, определённым всеми супертипами этого типа в иерархии типов. Или наоборот, ограничения и особенности, специфицированные определённым типом, должны быть определены на примерах этого типа так же хорошо, как и на примерах всех субтипов (подтипов) этого определённого типа.

Для графа типа расширим определение типов узлов и типов связей для того, чтобы ввести иерархию типов:

$NT_H = (NT, L_{NT})$ – иерархия типа узла,
$LT_H = (LT, L_{LH})$ – иерархия типа связи,
$GT = (NT_H, LT_H)$ – граф типа.

Введение структурных ограничений. Структурные ограничения являются средством описания знаний и правил для определённой предметной области.

Ранее отмечалось, что понятия домена, отношения между этими понятиями отображаются на типы узлов и типы связей. Граф типа определяет структуру домена, и, следовательно, структурные ограничения должны быть определены на типах *тип-уровня*.

Были определены два иерархических графа: *тип-уровня*, один для типов узлов, другой для типов связей. Расширим определение графа типа *GT* так, чтобы он стал направленным графом *тип-уровня.* При этом иерархия типа узла, так же как и иерархия типа связи, будет частичным графом *GT*. Типы узлов и типы связей являются узлами *GT*. Каждое структурное ограничение, в свою очередь, также представлено как узел *GT*. Тип узла или тип связи (узлы *GT*), которые имеют структурные ограничения, связаны с узлом, представляющим ограничение. Таким образом, расширенное определение *GT* предпо-

лагает, что типами узлов этого графа являются типы узлов, типы связей и ограничения.

Трёхуровневая метамодель. Последний описанный шаг приводит к необходимости ввода трёхуровневой метамодели.

Определён граф типов в терминах типов узлов и типов связей. Определены типы узлов и типы связей.

При рассмотренном двухуровневом подходе были введены типы (типы узлов, типы связей), определенные на *тип-уровне*.

Теперь мы говорим, что типы узлов и типы связей есть типы узлов графа типов (*GT*) *тип-уровня*. Следовательно, тип узла и тип связи должны быть *частным случаем* объекта более высокого уровня. Решением проблемы является ввод третьего уровня (рисунок 3), который назван *мета-тип-уровень*. Этот уровень был введён для того, чтобы типы узлов и типы связей как типы узлов *тип-уровня* являлись *частными случаями метатипов*. *Метатипы* будут определяться на *метатип-уровне*.

В более общей формулировке: типы элементов *пример-уровня* являются *частными случаями* типов узлов *тип-уровня*, которые в свою очередь являются *частными случаями* типов узлов *метатип-уровня*.

Формально это можно представить следующим образом:

TG = (*TN*, *TL*, *dom*(*TG*)) – типовой граф (*пример-уровень*),

GT = (*GTN*, *GTL*, *dom*(*GT*)) – граф типа (*тип-уровень*),

MT = (*MNT*, *MLT*) – метатип (*метатип-уровень*),

где для *пример-уровня*

$\forall\ tn \in TN : type(tn) \in GTN, \quad \forall\ tl \in TL : type(tl) \in GTN,$

для *тип-уровня*

$\forall\ gtn \in GTN : type(gtn) \in MNT, \quad \forall\ gtl \in GTL : type(gtl) \in MLT.$

Иерархии типов узлов и связей являются частичными графами *GT* и отображаются на *GT* так, что типы узлов, так же как и типы связей, являются узлами *GT*, а отношения тип узла – тип связи являются связями *GT*.

Множество типов узлов *MNT* на *метатип-уровне* как элементы *GTN* определяются как

$gtn = \langle nt, mt_{node} \rangle,\ nt \in (mt_{node}) \wedge mt_{node} \in MNT.$

Множество типов связей *MLT*, определённых на *мета-тип-уровне* через элементы *GTL,* определяются как:

$gtl = \langle gtn_i, gtn_j, gtls, mt_{link} \rangle,\ gtn_i, gtn_j \in GTN \wedge \wedge mt_{link} \in MLT,$

при этом в *gtls* возможна пустая последовательность промежуточных связей (элементов *GTL*).

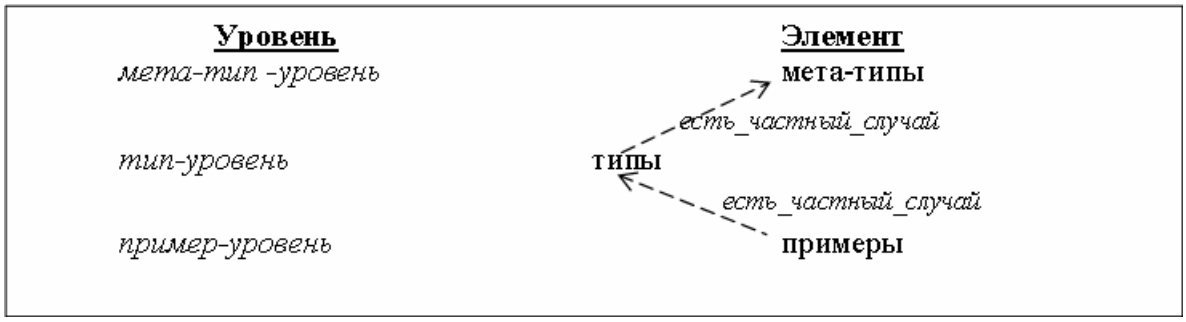

Рисунок 3

Заключение. В работе решена задача разработки теоретических основ для моделирования XML структур с хорошо определённым способом отображения на них семантики предметной области. Разработана трёхуровневая метамодель, включающая формальный метод отображения домена предметной области в терминах понятий и знания правил предметной области, которая позволяет осуществить последовательное разделение содержания вместе с определением типа структурных ограничений.

Библиографический список

1. *J.L. Schnase, J.J. Leggett, D.L. Hicks, R.L. Szabo,*"Semantic data modeling of hypermedia associations", ACM Trans. Inf. Syst. 11, 1, Jan. 1993, p. 27–50.

2. *J. Peckham, F. Mariansky,* "Semantic data models", ACMComput. Surv. 20,3, 1988, p. 153–189.

3. *H.A. Schmidt, J.R. Swenson,* "On the semantics of the relational data models", Proceedings of the SIGMOD San Jose, Calif., 1975.

4. *Abrial, Klimbie, Koffemen*, "Data Semantics In Database Management", Eds. North-Holland, Amsterdam, 1974, p. 1–59.

5. *M. Gogolla, U. Hohenstein,* "Towards a semantic view of an extended entity-relationship model", ACM Trans. Database Syst. 16, 3, 1991, p. 369–416.

6. W3C XLink, XML Linking Language Proposed Recommendation, http://www.w3c.org/XML/linking, Dec. 2000.

7. *W. Wang, R. Rada,* "Structured Hypertext with DomainSemantics"; ACM TIS, 16,4, 1998, pp. 372–412.

8. *J. Conklin,* "Hypertext: An introduction and survey", Computer 20, 9, 1987, p.17–41.