

# Эволюция ассоциаций в XML базах данных

Д.В. Луцив, Мат-Мех факультет СПбГУ

18 октября 2004 г.

1

## Ассоциации в XML-документах

1. Нарушают иерархическую структуру документа
2. Требуют дополнительных механизмов поддержки целостности
3. Наряду с атрибутами и тэгами являются объектами эволюции схемы данных

2

## Реализация

1. Без отражения в схеме документа  
За целостность документа отвечает прикладное ПО.
2. С отражением в схеме документа  
За целостность документа отвечают библиотека работы с XML и прикладное ПО.

3

Наиболее общий способ – установить зависимые друг от друга ограничения на значения реализующих роли ассоциации атрибутов.

Пусть ассоциация направлена от тэга  $t$  к тэгу  $s$ , при этом тэг  $s$  идентифицируется атрибутом  $b$ , а для ссылки используется атрибут  $a$  тэга  $t$ . При этом в схеме документа задаются конструкций `key` и `keyref`.

## Обозначения

$\tau$	граф тэгов в схеме документа
$s, t$	тэги
$a$	атрибут
$m$	тэг или атрибут
$N(t)$	«собственные» атрибуты $t$
$P(t)$	множество непосредственных предков $t$
$PL(t)$	граф предков $t$
$inst(s), inst(a)$	экземпляры элементов схемы в реальных документах
$val(s), val(a)$	значения $inst(s), inst(a)$ соответственно

$$\alpha_x(f, \tau) = \{f(x) | x \in \tau\}$$

5

## Конструкции `key` и `keyref`

*key* определяется в контексте тэга, имеет атрибут *name*.  
Включает тэги (*attribute+*, *selector*).

*keyref* определяется в контексте тэга, имеет атрибуты *name*  
и *refer* (имя *key*). Включает тэги (*attribute+*, *selector*)

*selector* определяет набор тегов, к которым применяется  
ключ (специфично для XML).

6

## Классификация выражений XPath

Общий случай:

$$t \in \tau \rightarrow \{\text{inst}(s \in \tau)\} \cup \{\text{inst}(a) \mid a \in \alpha_z(N(z), \tau)\}$$

«Упрощенные выражения» – без фильтров, терминов, атрибутов, без осей (кроме child::) исчерпывающе описывается уже на уровне схемы (например, `./house|./tower`):

$$\tau \rightarrow 2^\tau$$

7

«Простые выражения» – например, ./house или . – выглядят как

$$\tau \rightarrow \tau$$

Обозначим множество простых выражений  $\tau \rightarrow \tau$  как  $SP(\tau)$

Заметим, что  $SP(t) = \{s | t \in PL(s)\} \cup t$



8

Пусть  $A$  – множество имен атрибутов тэгов  $\tau$ ,  $K$  – множество ключей. Тогда определение простого ключа в схеме – функция следующего вида:

$\text{keydecl} : \tau \times SP \times (A \cup \tau) \rightarrow K$ , причем

$K = \{f_k : \{\text{val}(m)\} \rightarrow \{\text{inst}(s \in \tau)\}\}$ ,  $f_k$  – «ключевая функция»

Заметим, что для  $\text{keydecl}(t, p, m)$  при  $p \in SP$  справедливо:

$$a \in SP(t) \cup \alpha_z(N(z), SP(t)),$$

причем  $m \in SP(t) \Rightarrow P(m) \ni p(t)$ ,  $m \in N(s) \Rightarrow s = p(t)$

Кроме того, в общем случае вместо  $\tau \times SP \times (A \cup \tau)$  следует рассматривать  $\tau \times SP \times 2^{A \cup \tau}$

9

Контекстом ключа назовем множество

$$PL(t) \setminus PL(s)$$

$s$  - вершина контекста

Ключ обеспечивает уникальность тэга в своем контексте

10

Определение *простой* ссылки на *простой* ключ в схеме – функция вида:

$\text{keyrefdecl} : \tau \times SP \times (A \cup \tau) \times K \rightarrow R$ , причем

$R = \{f_r : \{\text{inst}(t)\} \rightarrow \{\text{val}(m)\}\}$ ,  $f_r$  – «функция ссылки»

Заметим, что для  $\text{keyrefdecl}(t, p, m, k)$  при  $p \in SP$  справедливо:

$a \in SP(t) \cup \alpha_z(N(z), SP(t))$ ,

причем  $m \in SP(t) \Rightarrow P(m) \ni p(t)$ ,  $m \in N(s) \Rightarrow s = p(t)$

а так же

$\exists f_k \in K : f_k = \text{keydecl}(t', p', m) \vee t \in SP(t')$

ТО ЕСТЬ КОНТЕКСТ ССЫЛКИ ДОЛЖЕН ВХОДИТЬ В КОНТЕКСТ КЛЮЧА

## Аксиомы целостности ассоциации

1. Целостность простого ключа – наличие единственного элемента в фактическом документе для каждого допустимого аргумента ключевой функции – здесь тавтологична, так как  $f_k$  - функция. На практике обеспечивается валидатором схемы.
2. Аксиома целостности ссылки – запрет «висячих» ссылок. Представлена ниже.

12

## Запрет «висячих» ссылок

$$f_k = \text{keydecl}(t_1, p_1, m_1), f_r = \text{keyrefdecl}(t_2, p_2, m_2, f_k) \\ t_1 \in PL(t_2) \cup \{t_2\}$$

тогда справедливо:

$$\forall e_2 \in \{\text{inst}(p_2(t_2))\} \exists e_1 \in \{\text{inst}(p_1(t_1))\} : (f_k \cdot f_r)(e_2) = e_1$$

То есть,  $f_k \cdot f_r$  корректно определена на всех тэгах вида  $t_2$ .

## Обозначения

- $N_e(t)$  существенные атрибуты  $t$   
 $P_e(t)$  существенные предки  $t$   
 $R_e(t)$  существенные ссылочные функции,  
определенные как  $\text{keyrefdecl}(s, p, *, *)$ ,  $p(s) \in \{t\} \cup PL(t)$   
 $NR_e(t)$  то же, но  $p(s) = t$

14

## Модификации структуры тэгов

1. Добавление атрибута
2. Удаление атрибута
3. Включение тэга
4. Исключение тэга
5. Добавление тэга
6. Удаление тэга
- 1.1. Добавление ключа
- 2.1. Снятие ключа
- 1.2. Добавление ссылки
- 2.2. Снятие ссылки

15

## Замечания

Операции 1-6 так же учитывают наличие ключей и ссылок

Множества, определенные у С. Кукса, сохраняют свои свойства

Здесь мы упрощаем модель до:

$$\text{keydecl} : \tau \times SP \times A \rightarrow K$$

$$\text{keyrefdecl} : \tau \times SP \times A \times K \rightarrow R$$



16

Добавление тэгов, атрибутов и отношений включения

Не требуют дополнительных действий.

17

## Удаление атрибутов

Требует операции снятия ключа для всех ключей на удаляемых атрибутах

## Удаление включения

Если в исключаемом тэге  $t$  есть ключевой атрибут, то вершина контекста ключа движется вверх по  $PL(t)$ , пока в контекст ключа не попадут все определения ссылок на этот ключ

Если в  $t$  есть существенный ссылочный атрибут, то ссылка уничтожается, если в ее контекст не входил корневой тэг.

Если в  $t$  находилась вершина контекста ссылки, то вершина контекста поднимается в направлении контекста ключа.

19

## Удаление тэга

Сводится к удалению атрибутов и включений. Подтэги удаляемого тэга получают «в собственность» существенные ссылки удаляемого.

20

## Добавлени и снятие ключей и ссылок

Добавление ключей и ссылок запрещается, если нарушаются определения ключей или ссылок, или условие отсутствия «висячих» ссылок.

Удаление ключа удаляет висячие ссылки.

При этом пересчитываются  $NR_e$  и  $R_e$ .

21

Дальнейшие исследования

Более тонкое описание контекста

Эволюция ассоциаций с ключевыми и ссылочными тэгами

Использование более полного XPath