

# XML Schema - schematy dokumentów

Joanna Jędrzejowicz

Instytut Informatyki

## Jakie są problemy z DTD i ograniczenia DTD

- brak mechanizmu 'typ danych' - tekstowa zawartość elementów nie może być ograniczona,
- brak możliwości określania typów i wartości atrybutów, a także kolejności atrybutów,
- niemożliwa jest zmiana kolejności elementów,
- deklaracja atrybutów i elementów nie mogą zależeć od kontekstu: często potrzebny jest możliwość użycia takiej deklaracji, która będzie zależna od wystąpienia lub braku atrybutu,
- dane znakowe nie mogą być związane z wyrażeniami regularnymi,

### Jakie są problemy z DTD i ograniczenia DTD

- DTD nie udostępnia mechanizmów związanych z modułami, ponownym użyciem typów - trudne jest operowanie dużymi DTD,
- mechanizm ID i IDREF jest zbyt prosty i nie wystarcza w pewnych sytuacjach,
- DTD nie wspiera przestrzeni nazw,
- DTD **nie** jest dokumentem XML

XML Schema umożliwia dokładne opisanie drzewiastej struktury dokumentu XML - typów danych znajdujących się w elementach i atrybutach. Z każdego DTD można automatycznie wygenerować odpowiadający mu schemat, ale nie odwrotnie!

W XML Schema `klientAdres.xsd` elementem głównym jest `schema` z przestrzeni `xsd` identyfikowanej z odpowiednim url (jakim?)

Porównajmy XML Schema `klientAdres.xsd` z dokumentem DTD `klientAdres.dtd` (z poprzedniego wykładu)

- bardziej sprecyzowane typy danych,
- wykorzystanie składni XML,
- możliwość użycia narzędzi XML

Tworząc schemat musimy zadeklarować **wszystkie** elementy i atrybuty, które chcemy wykorzystywać w dokumentach. Deklaracja zawiera nazwę oraz określenie typu danych.

W lokalnych deklaracjach elementów mogą się pojawiać ograniczenia liczebności:

`minOccurs`, `maxOccurs` dla elementów określają ile razy element może wystąpić,

`use` dla atrybutów określa czy atrybut jest wymagany (`required`) czy opcjonalny (`optional`)

por. atrybut `id` w przykładzie **`osoba.xsd`**

## Typy wg zasięgu definicji

- typy nazwane,
- typy anonimowe

## Typy wg zawartości

- typy proste,
- typy złożone o zawartości
  - prostej,
  - elementowej,
  - mieszanej,
  - pustej

## Typy wg pochodzenia

- typy wbudowane,
- typy zdefiniowane w schemacie: rozszerzenia innych typów, ograniczenia innych typów, listy i unie.

Typy **nazwane** są zdefiniowane globalnie i muszą mieć unikatowe nazwy - por. osoby.xsd.

Typ anonimowy nie ma nazwy - jest zdefiniowany wewnątrz deklaracji elementu lub atrybutu i może być użyty tylko raz przez tą deklarację (w przykładzie klientAdres.xsd)



Elementy, którym przypisano typy proste, mają zawartość tekstową, lecz nie mają podelementów ani atrybutów - w przykładzie

osoby.xsd zdefiniowany jest typ prosty `typ_im`

Elementy typu złożonego mogą mieć podelementy i (lub) atrybuty - jakie typy złożone w przykładzie `osoby.xsd`?

Atrybuty są tylko typów prostych.

## Grupy deklaracji elementów

- sequence - składniki grupy muszą wystąpić w takiej kolejności jak w deklaracji,
- choice - musi wystąpić jeden składnik grupy,
- all - składniki grupy muszą wystąpić w dowolnej kolejności

por. przykład [osoby.xsd](#)

Typy wbudowane można podzielić na sześć głównych kategorii:

- typy związane z czasem i datą - duration, dateTime, time, date, gYearMonth, gYear, gDay, gMonth, Day,
- typ logiczny - boolean,
- typy binarne - base64Binary, hexBinary (dane binarne zakodowane w postaci tekstowej),
- typ znakowe - string, normalizedString, token,
- typy liczbowe - float, double, decimal, integer, nonPositiveInteger, negativeInteger, nonnegativeInteger, unsignedLong, UnsignedInt, unsignedShort, unsignedByte, positiveInteger, long, int, short, byte,
- typy związane bezpośrednio z XML - anyURI, QName, NOTATION, language, Name, NCName, ID, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS

# Przykłady

typ	przyklad/komentarz
duration	P1Y2M3DT10H30M12S (1 rok, 2 mies. 3 dni, 10 godz, 30 min, 12 sek)
dateTime	2009-03-11T8:30:00
gYearMonth	2009-03
boolean	0
normalizedString	ciąg znaków bez znaku tabulacji, NL, CR
float	32 bitowa liczba zmiennoprzecinkowa
double	64 bitowa liczba zmiennoprzecinkowa
decimal	liczba stałoprzecinkowa
long	calc. na 8 bajtach
int	calc. na 4 bajtach
short	calc. na 2 bajtach
byte	liczby od -128 do 127
anyURI	dowolny identyfikator URI (także URL)
NCName	lokalna nazwa (bezprefiksu NS i :)

Typy proste (do definiowania elementów i atrybutów) powstają z typów wbudowanych przez

- **restriction** - ograniczenie typu wbudowanego przy pomocy tzw. aspektów (facets),
- **list** - stworzenie listy,
- **union** - suma typów prostych

Przykład **osoby.xsd** zawiera deklaracje typów prostych stworzonych przy pomocy restriction.

## Aspekty

- określenie dokładnej, minimalnej i maksymalnej długości - length, minLength, maxLength (np. kody towarowe),
- ograniczenie zakresu wartości liczbowych - maxInclusive, maxExclusive, minInclusive, minExclusive,
- maksymalna liczba cyfr w liczbie, liczb po przecinku - totalDigits, fractDigits,
- wzorzec - pattern,
- wyliczenie elementów - enumeration,

przykład [osoby.xsd](#)

Listy - w schematach są zbiorami wartości tego samego typu. Ograniczenia:

- można tworzyć tylko listy z typów prostych,
- wszystkie elementy są tego samego typu,
- nie można tworzyć list z list.

(por. [lotek.xsd](#))

Unie służą do łączenia dowolnych typów prostych w jeden typ -  
przykład `stawkiVat.xsd`.