



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Rozproszone technologie WebServices

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie
pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie



Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl

Wprowadzenie do technologii XML

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 11 kwietnia 2010

Program wykładu

Celem wykładu jest przedstawienie technologii XML i pokrewnych do niej technologii:

- XML: DTD, XML Schema, XSLT.
- Analiza leksykalna dokumentów XML.
- Protokoły XML-RPC i SOAP.
- Usługi internetowe budowane w oparciu o WebServices.
- Usługi WebServices na systemach mobilnych

Ćwiczenia będą realizowane przy użyciu różnych technologii.

Laboratorium

Zaliczenie laboratorium będzie wystawiane na podstawie zaliczenia wszystkich ćwiczeń cząstkowych.

Czym jest XML?

Najmodniejszym tematem w branży informatycznej. W każdej gazecie informatycznej znajdziemy wzmiankę o XML. Google na zapytanie o XML zwraca 343.000.000 dokumentów, Amazon oferuje 32.878 książek o tej tematyce.

Czym jest XML?

Najmodniejszym tematem w branży informatycznej. W każdej gazecie informatycznej znajdziemy wzmiankę o XML. Google na zapytanie o XML zwraca **343.000.000** dokumentów, Amazon oferuje 32.878 książek o tej tematyce.

Czym jest XML?

Najmodniejszym tematem w branży informatycznej. W każdej gazecie informatycznej znajdziemy wzmiankę o XML. Google na zapytanie o XML zwraca 343.000.000 dokumentów, Amazon oferuje **32.878** książek o tej tematyce.

Czym jest XML?

XML (Extensible Markup Language) - wywodzi się od języka SGML i jest językiem znaczników służącym do **opisu danych**. Dane przechowywane są w postaci **tekstowej** w dokumencie o ściśle określonej strukturze. XML możemy uznać za:

- technologie opisu i przechowywania danych,
- rodzinę technologii do prezentacji i przetwarzania danych,
- podstawowy składnik nowoczesnych technologii rozproszonych.
- XML jest stosowany również do opisu swoich własnych rozszerzeń.

Czym jest XML?

XML (Extensible Markup Language) - wywodzi się od języka SGML i jest językiem znaczników służącym do **opisu danych**. Dane przechowywane są w postaci **tekstowej** w dokumencie o ściśle określonej strukturze. XML możemy uznać za:

- technologie opisu i przechowywania danych,
- rodzinę technologii do prezentacji i przetwarzania danych,
- podstawowy składnik nowoczesnych technologii rozproszonych.
- XML jest stosowany również do opisu swoich własnych rozszerzeń.

Czym jest XML?

XML (Extensible Markup Language) - wywodzi się od języka SGML i jest językiem znaczników służącym do **opisu danych**. Dane przechowywane są w postaci **tekstowej** w dokumencie o ściśle określonej strukturze. XML możemy uznać za:

- technologie opisu i przechowywania danych,
- rodzinę technologii do prezentacji i przetwarzania danych,
- podstawowy składnik nowoczesnych technologii rozproszonych.
- XML jest stosowany również do opisu swoich własnych rozszerzeń.

Czym jest XML?

XML (Extensible Markup Language) - wywodzi się od języka SGML i jest językiem znaczników służącym do **opisu danych**. Dane przechowywane są w postaci **tekstowej** w dokumencie o ściśle określonej strukturze. XML możemy uznać za:

- technologie opisu i przechowywania danych,
- rodzinę technologii do prezentacji i przetwarzania danych,
- podstawowy składnik nowoczesnych technologii rozproszonych.
- XML jest stosowany również do opisu swoich własnych rozszerzeń.

Czym jest XML?

XML (Extensible Markup Language) - wywodzi się od języka SGML i jest językiem znaczników służącym do **opisu danych**. Dane przechowywane są w postaci **tekstowej** w dokumencie o ściśle określonej strukturze. XML możemy uznać za:

- technologie opisu i przechowywania danych,
- rodzinę technologii do prezentacji i przetwarzania danych,
- podstawowy składnik nowoczesnych technologii rozproszonych.
- XML jest stosowany również do opisu swoich własnych rozszerzeń.

Gdzie stosujemy XML?

XML jest standardem przemysłowym i stosowany jest we wszystkich dziedzinach informatyki:

- prawo - LegalXML, matematyka - MatchML, chemia - ChemXL.
- dokumenty i grafika: SVG, OpenOffice.
- komunikacja: XML-RPC, SOAP.
- prezentacja, transformacja, walidacja: XSL-FO, XSLT, XML-Schema.

Gdzie stosujemy XML?

XML jest standardem przemysłowym i stosowany jest we wszystkich dziedzinach informatyki:

- prawo - LegalXML, matematyka - MatchML, chemia - ChemXL.
- dokumenty i grafika: SVG, OpenOffice.
- komunikacja: XML-RPC, SOAP.
- prezentacja, transformacja, walidacja: XSL-FO, XSLT, XML-Schema.

Gdzie stosujemy XML?

XML jest standardem przemysłowym i stosowany jest we wszystkich dziedzinach informatyki:

- prawo - LegalXML, matematyka - MatchML, chemia - ChemXL.
- dokumenty i grafika: SVG, OpenOffice.
- komunikacja: XML-RPC, SOAP.
- prezentacja, transformacja, walidacja: XSL-FO, XSLT, XML-Schema.

Gdzie stosujemy XML?

XML jest standardem przemysłowym i stosowany jest we wszystkich dziedzinach informatyki:

- prawo - LegalXML, matematyka - MatchML, chemia - ChemXL.
- dokumenty i grafika: SVG, OpenOffice.
- komunikacja: XML-RPC, SOAP.
- prezentacja, transformacja, walidacja: XSL-FO, XSLT, XML-Schema.

Gdzie stosujemy XML?

XML jest standardem przemysłowym i stosowany jest we wszystkich dziedzinach informatyki:

- prawo - LegalXML, matematyka - MatchML, chemia - ChemXL.
- dokumenty i grafika: SVG, OpenOffice.
- komunikacja: XML-RPC, SOAP.
- prezentacja, transformacja, walidacja: XSL-FO, XSLT, XML-Schema.

XML - samo opisujące się dane

- Znacznik opisuje znaczenie i funkcję elementu.
- Programista/aplikacja może przetwarzać dokument mając na uwadze typ elementu a nie sposób jego prezentacji.
- XML umożliwia bezpieczną wymianę danych pomiędzy różnymi systemami i prezentację wyników na różnych platformach: komputer osobisty, PDA, kasa fiskalna itd.

XML - samo opisujące się dane

- Znacznik opisuje znaczenie i funkcję elementu.
- Programista/aplikacja może przetwarzać dokument mając na uwadze typ elementu a nie sposób jego prezentacji.
- XML umożliwia bezpieczną wymianę danych pomiędzy różnymi systemami i prezentację wyników na różnych platformach: komputer osobisty, PDA, kasa fiskalna itd.

XML - samo opisujące się dane

- Znacznik opisuje znaczenie i funkcję elementu.
- Programista/aplikacja może przetwarzać dokument mając na uwadze typ elementu a nie sposób jego prezentacji.
- XML umożliwia bezpieczną wymianę danych pomiędzy różnymi systemami i prezentację wyników na różnych platformach: komputer osobisty, PDA, kasa fiskalna itd.

Przykład

Opis płyt dvd znajdujących się w wypożyczalni

```
<?xml version='1.0' encoding='iso-8859-2'?>
<katalog>
  <dvd id='1'>
    <tytuł>Rambo I</tytuł>
    <cena>5</cena>
    <sztuk>10</sztuk>
  </dvd>
  <dvd id='2'>
    <tytuł>Rambo II</tytuł>
    <cena>7</cena>
    <sztuk>12</sztuk>
  </dvd>
</katalog>
```

Czym nie jest XML?

- XML nie jest językiem programowania. Nie można przeprowadzić w nim obliczeń.
- Nie opisuje w jaki sposób dane mają być prezentowane. Jednak technologie rozwinięte wokół XML pozwalają na transformację i prezentację danych.

Czym nie jest XML?

- XML nie jest językiem programowania. Nie można przeprowadzić w nim obliczeń.
- Nie opisuje w jaki sposób dane mają być prezentowane. Jednak technologie rozwinięte wokół XML pozwalają na transformację i prezentację danych.

HTML ...

- Podobnie jak XML, HTML jest językiem znaczników.
- W odróżnieniu od XML, HTML opisuje sposób prezentacji, a nie dane.
- Przeglądarki HTML starają się czytać i interpretować nieprawidłowe dokumenty HTML (brak kończących znaczników itd.).
- HTML ma tylko jedno medium prezentacji. Jest nim standardowy ekran komputera.

HTML ...

- Podobnie jak XML, HTML jest językiem znaczników.
- W odróżnieniu od XML, HTML opisuje sposób prezentacji, a nie dane.
- Przeglądarki HTML starają się czytać i interpretować nieprawidłowe dokumenty HTML (brak kończących znaczników itd.).
- HTML ma tylko jedno medium prezentacji. Jest nim standardowy ekran komputera.

HTML ...

- Podobnie jak XML, HTML jest językiem znaczników.
- W odróżnieniu od XML, HTML opisuje sposób prezentacji, a nie dane.
- Przeglądarki HTML starają się czytać i interpretować nieprawidłowe dokumenty HTML (brak kończących znaczników itd.).
- HTML ma tylko jedno medium prezentacji. Jest nim standardowy ekran komputera.

HTML ...

- Podobnie jak XML, HTML jest językiem znaczników.
- W odróżnieniu od XML, HTML opisuje sposób prezentacji, a nie dane.
- Przeglądarki HTML starają się czytać i interpretować nieprawidłowe dokumenty HTML (brak kończących znaczników itd.).
- HTML ma tylko jedno medium prezentacji. Jest nim standardowy ekran komputera.

Problemy wynikające z obecnego stanu technologii HTML

- Jedna strona HTML jest wyświetlana w różny sposób przez różne przeglądarki.
- Ze względu na rozszerzenia implementowane przez firmy niezależnie od komisji standaryzacyjnej, konieczne jest przygotowanie wielu wersji tej samej strony dla najpopularniejszych przeglądarek.

Rozwiązanie ...

W celu uniezależnienia dokumentów HTML od konkretnego medium prezentacji powstała nowa wersja HTML, która jest podzbiorem języka XML. Technologia ta nazywa się XHTML.

Problemy wynikające z obecnego stanu technologii HTML

- Jedna strona HTML jest wyświetlana w różny sposób przez różne przeglądarki.
- Ze względu na rozszerzenia implementowane przez firmy niezależnie od komisji standaryzacyjnej, konieczne jest przygotowanie wielu wersji tej samej strony dla najpopularniejszych przeglądarek.

Rozwiązanie ...

W celu uniezależnienia dokumentów HTML od konkretnego medium prezentacji powstała nowa wersja HTML, która jest podzbiorem języka XML. Technologia ta nazywa się XHTML.

Problemy wynikające z obecnego stanu technologii HTML

- Jedna strona HTML jest wyświetlana w różny sposób przez różne przeglądarki.
- Ze względu na rozszerzenia implementowane przez firmy niezależnie od komisji standaryzacyjnej, konieczne jest przygotowanie wielu wersji tej samej strony dla najpopularniejszych przeglądarek.

Rozwiązanie ...

W celu uniezależnienia dokumentów HTML od konkretnego medium prezentacji powstała nowa wersja HTML, która jest podzbiorem języka XML. Technologia ta nazywa się XHTML.

Struktura dokumentu XML

Rekomendacja W3C znajduje się na stronie:

`http://www.w3.org/TR/REC-xml/`

Dokument XML składa się z trzech sekcji:

- Prolog.
- Treść.
- Epilog.

Jedynie sekcja treść jest obowiązkowa. Pozostałe dwie nie muszą występować. Dodatkowym elementem jest komentarz:

```
<!-- mój komentarz -->
```

Prolog

Prolog stanowi pierwszą sekcję dokumentu i powinien rozpoczynać się od deklaracji XML. Deklaracja XML wygląda następująco:

```
<?xml version='1.0'?'>
```

Deklaracja ta zawiera informacje o typie dokumentu (XML) i jego wersji (1.0). Dodatkowo można podać informację o kodowaniu i zaznaczyć czy dokument wymaga zewnętrznych dokumentów:

```
<?xml version='1.0' encoding='utf-8'
      standalone='yes'?'>
```

Treść

W treści dokumentu znajdują się właściwe dane. Treść dokumentu XML zawiera tylko jeden element objęty parą znaczników:

```
<dane> ...</dane>
```

W początkowej parze znaczników może być zagnieżdżona dowolna ilość elementów (znaczników) XML.

Epilog

Epilog jest najczęściej pomijany. Może zawierać instrukcje dotyczące przetwarzania dokumentu.

Element

Element jest podstawowym pojemnikiem do przechowywania danych. Definiuje typ i przeznaczenie przechowywanych danych.

```
<imię> Bartłomiej </imię>
```

Element może nie posiadać danych. Dozwolony jest brak zawartości pomiędzy znacznikami:

```
<imię> </imię>
```

Można użyć skróconego zapisu:

```
<imię/>
```

Element

Element jest podstawowym pojemnikiem do przechowywania danych. Definiuje typ i przeznaczenie przechowywanych danych.

```
<imię> Bartłomiej </imię>
```

Element może nie posiadać danych. Dozwolony jest brak zawartości pomiędzy znacznikami:

```
<imię> </imię>
```

Można użyć skróconego zapisu:

```
<imię/>
```

Element

Element jest podstawowym pojemnikiem do przechowywania danych. Definiuje typ i przeznaczenie przechowywanych danych.

```
<imię> Bartłomiej </imię>
```

Element może nie posiadać danych. Dozwolony jest brak zawartości pomiędzy znacznikami:

```
<imię> </imię>
```

Można użyć skróconego zapisu:

```
<imię/>
```

Element i atrybuty

Każdy element może posiadać dodatkowe atrybuty:

```
<dvd id='19922' typ='komedia'>Shrek</dvd>
```

Użycie atrybutu jest ściśle określone:

- Wszystkie wartości atrybutów muszą być ujęte w cudzysłowy lub apostrofy.
- Nie można użyć kilkakrotnie tej samej nazwy wewnątrz jednego atrybutu.
- W wartościach atrybutów nie mogą występować znaki < i &. Zamiast nich stosujemy < i &.

Element i atrybuty

Każdy element może posiadać dodatkowe atrybuty:

```
<dvd id='19922' typ='komedia'>Shrek</dvd>
```

Użycie atrybutu jest ściśle określone:

- Wszystkie wartości atrybutów muszą być ujęte w cudzysłowy lub apostrofy.
- Nie można użyć kilkakrotnie tej samej nazwy wewnątrz jednego atrybutu.
- W wartościach atrybutów nie mogą występować znaki < i &. Zamiast nich stosujemy < i &.

Element i atrybuty

Predefiniowane znaki w dokumentach XML:

Sekwencja:	Znak:
<	<
>	>
&	&
"	"
'	'

Dodatkowe informacje o składni XML

- XML jest wrażliwy na wielkość liter. Znacznik `<from>` i `<From>` to dwa różne znaczniki.
- XML uwzględnia białe znaki pomiędzy znacznikami.
- Jedyne poprawne zakończenie linii dokumentu XML to znak LF (`'\n'`).

Dodatkowe informacje o składni XML

- XML jest wrażliwy na wielkość liter. Znacznik `<from>` i `<From>` to dwa różne znaczniki.
- XML uwzględnia białe znaki pomiędzy znacznikami.
- Jedyne poprawne zakończenie linii dokumentu XML to znak LF (`'\n'`).

Dodatkowe informacje o składni XML

- XML jest wrażliwy na wielkość liter. Znacznik `<from>` i `<From>` to dwa różne znaczniki.
- XML uwzględnia białe znaki pomiędzy znacznikami.
- Jedyne poprawne zakończenie linii dokumentu XML to znak LF (`'\n'`).

Znaczniki

W XML znaczniki muszą być nazywane zgodnie z zasadami:

- Nazwa może zawierać litery, cyfry i inne znaki drukowalne.
- Nazwa musi zaczynać się od litery.
- Nazwa nie może zaczynać się od ciągu 'xml', 'XML', 'Xml'.
- Nazwa nie może zawierać znaków białych.

Definicja typu dokumentu

Aby dokument XML był użyteczny konieczne jest oprócz zdefiniowania składni zdefiniowanie znaczników i określenie sekwencji w jakich mogą występować.

Opis logiczny dokumentu realizuje się między innymi za pomocą definicji typu dokumentu **DTD** (Document Type Definition).

Definicja DTD

DTD jest dokładną specyfikacją tego, co może pojawić się w dokumencie XML i ściśle określa strukturę drzewa dokumentu (poziomy zagnieżdżeń). Dokument XML, który nie zawiera opisu DTD określa się mianem **niepoprawnego**.

DTD jest szkieletem dokumentu XML.

Podstawą DTD jest element

Podstawę DTD stanowi deklaracja **ELEMENT**, która ma postać:

```
<!ELEMENT nazwa_elementu ...>
```

Za nazwą znacznika można wymienić zawarte w nim elementy podrzędne.

Elementy podrzędne

Zasady dodawania elementów podrzędnych:

- , - Przecinek używany jest do tworzenia listy elementów podrzędnych, które muszą wystąpić w odpowiedniej kolejności.
- | - Znakiem 'pipe' rozdziela się elementy opcjonalne.
- ? - Pytajnik oznacza opcjonalny element podrzędny.
- * - Dowolna liczba wystąpień elementu podrzędnego.
- + - Co najmniej jedno wystąpienie elementu podrzędnego.
- (...) - Grupowanie elementów podrzędnych.

Przykład

```
<!ELEMENT kanapka (chleb, (dżem | nutella), chleb) >
```

```
<!ELEMENT kanapka (chleb, (dżem | nutella)?, chleb) >
```

```
<!ELEMENT napój (herbata, (kakao | sok)?) >
```

```
<!ELEMENT posiłek (kanapka+, napój) >
```

Przykład

```
<!ELEMENT kanapka (chleb, (dżem | nutella), chleb) >
```

```
<!ELEMENT kanapka (chleb, (dżem | nutella)?, chleb) >
```

```
<!ELEMENT napój (herbata, (kakao | sok)?) >
```

```
<!ELEMENT posiłek (kanapka+, napój) >
```

Przykład

```
<!ELEMENT kanapka (chleb, (dżem | nutella), chleb) >
```

```
<!ELEMENT kanapka (chleb, (dżem | nutella)?, chleb) >
```

```
<!ELEMENT napój (herbata, (kakao | sok)?) >
```

```
<!ELEMENT posiłek (kanapka+, napój) >
```

Przykład - wypożyczalnia DVD

Opis struktury drzewiastej bez uwzględnienia typów

```
<!ELEMENT katalog (dvd+) >  
<!ELEMENT dvd (tytuł, cena, sztuk) >
```

Typy elementów

Typy danych

- CDATA - napis.
- ID - nazwa unikatowa w dokumencie XML.
- IDREF - odwołanie do innego elementu po ID.
- ENTITY - nazwa zewnętrznej jednostki.
- NOTATION - zdefiniowana na zewnątrz notacja.

Kwantyfikatory typów danych

- #REQUIRED - atrybut wymagany.
- #IMPLIED - atrybut opcjonalny.
- #FIXED <wartość> - atrybut musi mieć podaną wartość.

Typy elementów

Typy danych

- CDATA - napis.
- ID - nazwa unikatowa w dokumencie XML.
- IDREF - odwołanie do innego elementu po ID.
- ENTITY - nazwa zewnętrznej jednostki.
- NOTATION - zdefiniowana na zewnątrz notacja.

Kwantyfikatory typów danych

- #REQUIRED - atrybut wymagany.
- #IMPLIED - atrybut opcjonalny.
- #FIXED <wartość> - atrybut musi mieć podaną wartość.

Przykład - wypożyczalnia DVD

Opis struktury drzewiastej z uwzględnieniem typów

```
<!ELEMENT katalog (dvd+) >  
<!ELEMENT dvd (tytuł, cena, sztuk) >  
<!ATTLIST dvd id ID #REQUIRED >  
<!ELEMENT tytuł (#PCDATA) >  
<!ELEMENT cena (#PCDATA) >  
<!ELEMENT sztuk (#PCDATA) >
```

Powiązanie DTD z dokumentem XML

Pełna definicja dokumentu XML z wypożyczalni DVD:

```
<!DOCTYPE katalog [  
  <!ELEMENT katalog (dvd+) >  
  <!ELEMENT dvd (tytuł, cena, sztuk) >  
  <!ATTLIST dvd id ID #REQUIRED >  
  <!ELEMENT tytuł (#PCDATA) >  
  <!ELEMENT cena (#PCDATA) >  
  <!ELEMENT sztuk (#PCDATA) >  
>
```

Dołączanie DTD do dokumentu XML

Opis struktury dokumentu XML w postaci definicji DTD można dołączyć do dokumentu na dwa sposoby:

- Poprzez umieszczenie go w prologu dokumentu:

```
<?xml version="1.0"?>  
<!DOCTYPE znacznik-root [  
...  

```

- Dołączenie zewnętrznego pliku z opisem DTD do dokumentu XML:

```
<?xml version="1.0"?>  
<!DOCTYPE znacznik-root SYSTEM "nazwa_pliku">
```

Dołączanie DTD do dokumentu XML

Opis struktury dokumentu XML w postaci definicji DTD można dołączyć do dokumentu na dwa sposoby:

- Poprzez umieszczenie go w prologu dokumentu:

```
<?xml version="1.0"?>  
<!DOCTYPE znacznik-root [  
...  

```

- Dołączenie zewnętrznego pliku z opisem DTD do dokumentu XML:

```
<?xml version="1.0"?>  
<!DOCTYPE znacznik-root SYSTEM "nazwa_pliku">
```

Wstęp

XML Schema służy do definiowania struktury dokumentu XML (pełni podobną funkcję jak DTD). XML Schema pozwala na:

- zdefiniowanie elementów dokumentu
- zdefiniowanie atrybutów poszczególnych elementów
- określenie które elementy są zagnieżdżone (hierarchia elementów)
- określenie kolejności elementów
- zdefiniowanie wartości elementów (pusty czy zawiera tekst?)
- określenie typu wartości elementów i atrybutów
- zdefiniowanie domyślnych i stałych wartości elementów i atrybutów

XML Schema jako następca DTD

Z powodów takich jak:

- XML Schema jest technologią, którą można rozszerzać w razie potrzeby
- XML Schema jest znacznie rozbudowana w stosunku do DTD
- dokumenty XML Schema są zapisane jako XML
- XML Schema wspiera typy danych i umożliwia definiowanie złożonych typów danych
- XML Schema pozwala na definiowanie przestrzeni nazw

spodziewa się, że XML Schema będzie używana w większości aplikacji i wyprze DTD. XML Schema jest oficjalną rekomendacją W3C.

Definiowanie typów danych

XML Schema umożliwia definiowanie typów danych co jest największą zaletą w stosunku do DTD.

- łatwiej jest zdefiniować poprawną zawartość dokumentu
- możliwa jest walidacja i sprawdzenie poprawności danych
- łatwiejsza jest współpraca z bazami danych
- prościej jest konwersja pomiędzy typami danych

XML Schema a dokumenty XML

Ciekawą własnością XML Schema jest to, że jest opisany za pomocą XML. Ponieważ XML Schema został napisany w XML to:

- nie trzeba uczyć się nowego języka (opisu)
- możliwa jest edycja XML Schema przy pomocy standardowych edytorów XML
- można użyć parserów XML do parsowania XML Schema
- można użyć technologii DOM do tworzenia i manipulowania XML Schema
- można użyć arkuszy XSLT do transformacji XML Schema

XML Schema a bezpieczeństwo wymiany danych

XML jest najczęściej wykorzystywany w celu wymiany danych pomiędzy dwoma systemami. Niezwykle ważne jest aby nadawca i odbiorca dokumentu mieli takie same wymagania odnośnie jego struktury i zawartości.

XML Schema opisuje typy danych w taki sposób, że możliwa jest jednorodna interpretacja zawartości treści przez nadawcę i odbiorcę.

XML Schema a bezpieczeństwo wymiany danych

XML jest najczęściej wykorzystywany w celu wymiany danych pomiędzy dwoma systemami. Niezwykle ważne jest aby nadawca i odbiorca dokumentu mieli takie same wymagania odnośnie jego struktury i zawartości.

XML Schema opisuje typy danych w taki sposób, że możliwa jest jednorodna interpretacja zawartości treści przez nadawcę i odbiorcę.

Przykład:

```
<date type="date">1980-03-24</date>
```

Data zostanie poprawnie zinterpretowana przez wszystkie systemy, gdyż XML Schema definiuje datę w postaci RRRR-MM-DD.

XML Schema jest rozszerzalny

XML Schema podobnie jak sam XML jest rozszerzalny (ponieważ jest napisany w XML 😊) Poprzez rozszerzanie rozumie się:

- użycie jednej definicji XML Schema wewnątrz innej definicji
- definiowanie własnych typów złożonych w oparciu o typy standardowe (wbudowane)
- odwoływanie się do wielu schematów XML Schema z jednego dokumentu XML

Dlaczego XML Schema

Poprawnie sformatowany dokument to dokument XML zgodny z zasadami:

- musi rozpoczynać się od deklaracji XML
- musi posiadać unikalny element główny
- wszystkie elementy (tagi) muszą być zakończone
- elementy XML (tagi) są wrażliwe na wilkość liter
- elementy muszą być prawidłowo zagnieżdżone
- wszystkie wartości atrybutów muszą być ujęte w cudzysłowy
- znaki specjalne muszą być zastąpione określonymi jednostkami XML

Dlaczego XML Schema

Poprawnie sformatowany dokument to za mało!

Poprawnie sformatowany dokument to dokument XML zgodny z zasadami:

- musi rozpoczynać się od deklaracji XML
- musi posiadać unikalny element główny
- wszystkie elementy (tagi) muszą być zakończone
- elementy XML (tagi) są wrażliwe na wilkość liter
- elementy muszą być prawidłowo zagnieżdżone
- wszystkie wartości atrybutów muszą być ujęte w cudzysłowy
- znaki specjalne muszą być zastąpione określonymi jednostkami XML

Nawet poprawnie sformatowany dokument może zawierać błędy!

Prosty dokument XML

```
<?xml version='1.0' encoding='iso-8859-2'?>
<dvd>
  <tytuł>Rambo I</tytuł>
  <cena>5</cena>
  <sztuk>10</sztuk>
</dvd>
```

Prosta definicja DTD

```
<!ELEMENT dvd (tytuł, cena, sztuk) >  
<!ELEMENT tytuł (#PCDATA) >  
<!ELEMENT cena (#PCDATA) >  
<!ELEMENT sztuk (#PCDATA) >
```

Definicja XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
  <xs:element name="dvd">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="tytuł" type="xs:string"/>
        <xs:element name="cena" type="xs:integer"/>
        <xs:element name="sztuk" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://swierczu.com"  
xmlns="http://swierczu.com"  
elementFormDefault="qualified">  
  <xs:element name="dvd">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="tytuł" type="xs:string"/>  
        <xs:element name="cena" type="xs:integer"/>  
        <xs:element name="sztuk" type="xs:integer"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://swierczu.com"  
xmlns="http://swierczu.com"  
elementFormDefault="qualified">  
  <xs:element name="dvd">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="tytuł" type="xs:string"/>  
        <xs:element name="cena" type="xs:integer"/>  
        <xs:element name="sztuk" type="xs:integer"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://swierczu.com"  
xmlns="http://swierczu.com"  
elementFormDefault="qualified">  
  <xs:element name="dvd">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="tytuł" type="xs:string"/>  
        <xs:element name="cena" type="xs:integer"/>  
        <xs:element name="sztuk" type="xs:integer"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://swierczu.com"  
xmlns="http://swierczu.com"  
elementFormDefault="qualified">  
  <xs:element name="dvd">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="tytuł" type="xs:string"/>  
        <xs:element name="cena" type="xs:integer"/>  
        <xs:element name="sztuk" type="xs:integer"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://swierczu.com"  
xmlns="http://swierczu.com"  
elementFormDefault="qualified">  
  <xs:element name="dvd">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="tytuł" type="xs:string"/>  
        <xs:element name="cena" type="xs:integer"/>  
        <xs:element name="sztuk" type="xs:integer"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>
```

Element <schema>

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
...
</xs:schema>
```

Element <schema>

Element <schema> jest elementem głównym dokumentu XML Schema. Element ten może zawierać wiele atrybutów.

Element <schema>

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
...
</xs:schema>
```

xmlns:xs ...

Atrybut xmlns:xs oznacza, że elementy i typy danych użyte w definicji XML Schema pochodzą z przestrzeni nazw „http://www.w3.org/2001/XMLSchema”. Wszystkie elementy pochodzące z tej przestrzeni nazw powinny mieć prefix xs.

Element <schema>

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
...
</xs:schema>
```

targetNamespace ...

Atrybut targetNamespace oznacza, że elementy zdefiniowane w XML Schema (dvd, tytuł, cena, sztuk) pochodzą z przestrzeni nazw „http://swierczu.com”.

Element <schema>

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
...
</xs:schema>
```

xmlns ...

Domyślną przestrzenią nazw jest przestrzeń „http://swierczu.com”.

Element `<schema>`

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://swierczu.com"
xmlns="http://swierczu.com"
elementFormDefault="qualified">
...
</xs:schema>
```

`elementFormDefault="qualified"`

Użycie tego atrybutu oznacza, że wszystkie elementy zdefiniowane w tym schemacie XML Schema i użyte w innym dokumencie XML muszą być poprzedzone nazwą przestrzeni nazw.

Dołączenie XML Schema do dokumentu

```
<?xml version="1.0" encoding="iso-8859-2"?>
<dvd xmlns="http://swierczu.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://swierczu.com note.xsd">
  <tytuł>Rambo I</tytuł>
  <cena>5</cena>
  <sztuk>10</sztuk>
</dvd>
```

Dołączenie XML Schema do dokumentu

```
<?xml version='1.0' encoding='iso-8859-2'?>
<dvd xmlns="http://swierczu.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://swierczu.com note.xsd">
    <tytuł>Rambo I</tytuł>
    <cena>5</cena>
    <sztuk>10</sztuk>
</dvd>
```

`xmlns="http://swierczu.com"`

Deklaracja domyślnej przestrzeni nazw. Wszystkie elementy użyte w dokumencie powinny być zadeklarowane w przestrzeni nazw „http://swierczu.com”.

Dołączenie XML Schema do dokumentu

```
<?xml version='1.0' encoding='iso-8859-2'?>
<dvd xmlns="http://swierczu.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://swierczu.com note.xsd">
  <tytuł>Rambo I</tytuł>
  <cena>5</cena>
  <sztuk>10</sztuk>
</dvd>
```

`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`

Deklaruje przestrzeń nazw xsi jako

„`http://www.w3.org/2001/XMLSchema-instance`”.

Dołączenie XML Schema do dokumentu

```
<?xml version='1.0' encoding='iso-8859-2'?>
<dvd xmlns="http://swierczu.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://swierczu.com note.xsd">
  <tytuł>Rambo I</tytuł>
  <cena>5</cena>
  <sztuk>10</sztuk>
</dvd>
```

`xsi:schemaLocation="http://swierczu.com note.xsd"`

Lokalizacja dokumentu XML Schema dla przestrzeni nazw xsi.

Bardziej złożony dokument XML

```
<?xml version='1.0' encoding='iso-8859-2'?>
<katalog>
  <dvd>
    <tytuł>Rambo I</tytuł>
    <cena>5</cena>
    <sztuk>10</sztuk>
  </dvd>
  <dvd>
    <tytuł>Rambo II</tytuł>
    <cena>7</cena>
    <sztuk>11</sztuk>
  </dvd>
</katalog>
```

Definicja DTD

```
<!ELEMENT katalog (dvd)+ >  
<!ELEMENT dvd (tytuł, cena, sztuk) >  
<!ELEMENT tytuł (#PCDATA) >  
<!ELEMENT cena (#PCDATA) >  
<!ELEMENT sztuk (#PCDATA) >
```

Definicja XML Schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="katalog">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="dvd" minOccurs="1" maxOccurs="unbounded">  
          <xs:complexType>  
            <xs:sequence>  
              <xs:element name="tytuł" type="xs:string"/>  
              <xs:element name="cena" type="xs:integer"/>  
              <xs:element name="sztuk" type="xs:integer"/>  
            </xs:sequence>  
          </xs:complexType>  
        </xs:element>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
</xs:schema>
```

Definicja XML Schema - referencje

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="tytuł" type="xs:string"/>
  <xs:element name="cena" type="xs:integer"/>
  <xs:element name="sztuk" type="xs:integer"/>
  <xs:element name="dvd">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="tytuł">
        <xs:element ref="cena">
        <xs:element ref="sztuk">
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="katalog">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="dvd" minOccurs="1" maxOccurs="unbounded">
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Definicja XML Schema - referencje

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="tytuł" type="xs:string"/>
<xs:element name="cena" type="xs:integer"/>
<xs:element name="sztuk" type="xs:integer"/>
<xs:element name="dvd">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tytuł">
      <xs:element ref="cena">
      <xs:element ref="sztuk">
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="katalog">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="dvd" minOccurs="1" maxOccurs="unbounded">
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Definicja XML Schema - referencje

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="tytuł" type="xs:string"/>
<xs:element name="cena" type="xs:integer"/>
<xs:element name="sztuk" type="xs:integer"/>
<xs:element name="dvd">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tytuł">
      <xs:element ref="cena">
      <xs:element ref="sztuk">
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="katalog">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="dvd" minOccurs="1" maxOccurs="unbounded">
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Typy danych

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Prosty element

```
<xs:element name="xxx" type="yyy"/>
```

```
<xs:element name="color" type="xs:string" default="red"/>
```

```
<xs:element name="color" type="xs:string" fixed="red"/>
```

Atrybuty

```
<xs:attribute name="xxx" type="yyy"/>
```

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
```

```
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
```

```
<xs:attribute name="lang" type="xs:string" use="optional"/>
```

```
<xs:attribute name="lang" type="xs:string" use="required"/>
```

Typy złożone

```
<dvd id="76"/>
```

Typy złożone

```
<dvd id="76"/>

<xs:element name="dvd">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:integer">
        <xs:attribute name="id" type="xs:positiveInteger"/>

      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Typy złożone

```
<dvd id="76"/>

<xs:element name="dvd">
  <xs:complexType>
    <xs:complexContent>
      <xs:attribute name="id" type="xs:positiveInteger"/>

      </xs:complexContent>
    </xs:complexType>
  </xs:element>
```

Definiowanie typów

```
<xs:simpleType name="nazwiskoTyp">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="32"/>  
  </xs:restriction>  
</xs:simpleType>
```

Definiowanie typów

```
<xs:simpleType name="peselTyp">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9]{10}"/>  
  </xs:restriction>  
</xs:simpleType>
```

Odnosiniki:

Specyfikacje XML Schema W3C:

- Primer: <http://www.w3.org/TR/xmlschema-0/>
- Structures: <http://www.w3.org/TR/xmlschema-1/>
- Datatypes: <http://www.w3.org/TR/xmlschema-2/>



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Rozproszone technologie WebServices

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie
pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie



Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl