

# XML Schema

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 19 listopada 2005 roku

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

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.

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

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.

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 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 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



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!

## 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!

```
<?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>
</xs:schema>
```

# 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>
</xs:schema>
```

# 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>
</xs:schema>
```

# 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>
</xs:schema>
```



# 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>
</xs:schema>
```

# 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>
</xs:schema>
```

## 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.

```
<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`.

```
<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`”.

```
<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”.

```
<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>
```

```
<!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>
</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>
```

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

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

```
<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"/>
```

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

```
<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>
```

```
<dvd id="76"/>  
  
<xs:element name="dvd">  
  <xs:complexType>  
    <xs:complexContent>  
      <xs:attribute name="id" type="xs:positiveInteger"/>  
  
    </xs:complexContent>  
  </xs:complexType>  
</xs:element>
```

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



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

## 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/>