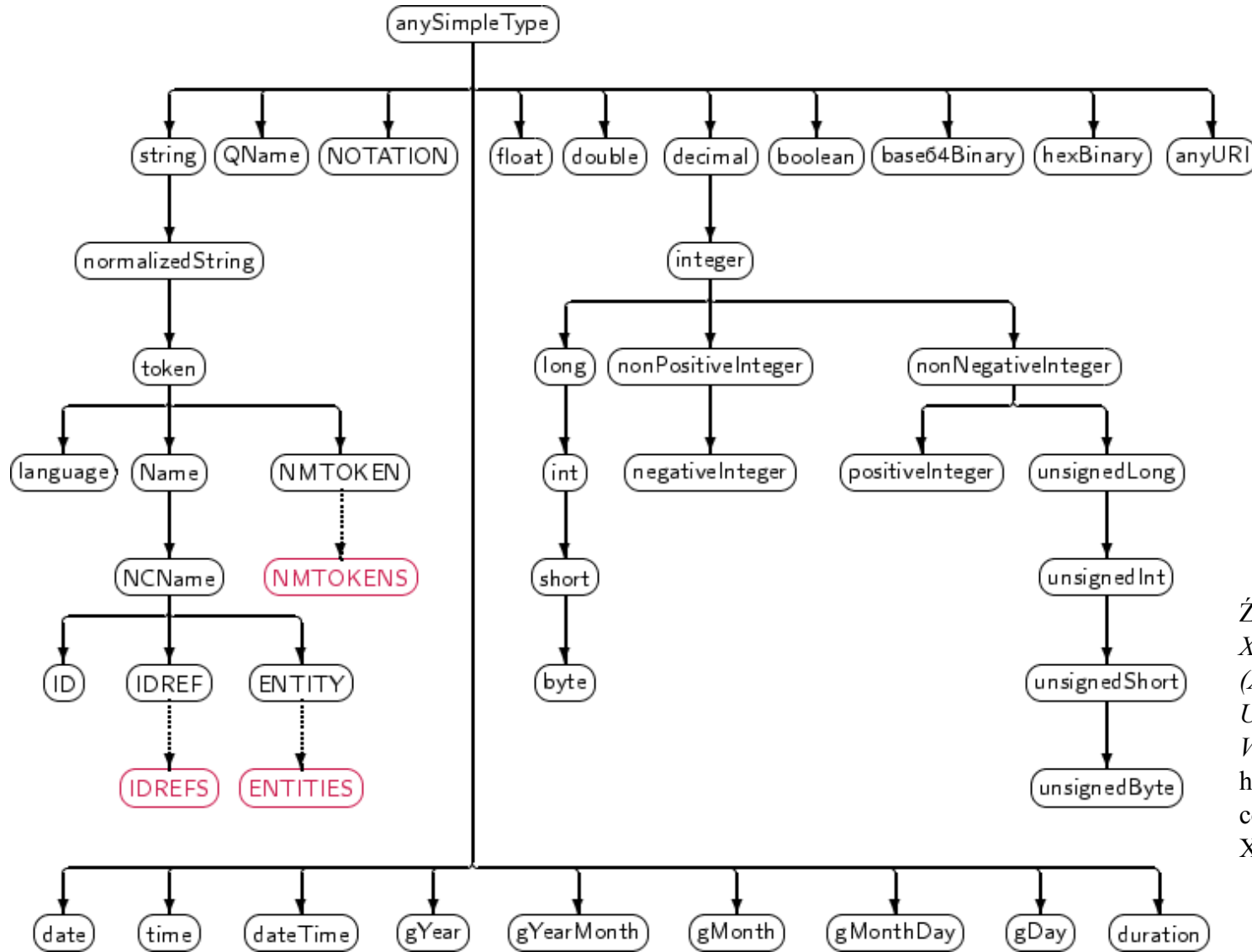


Definiowanie typów dokumentów

Część 3. XML Schema

Wbudowane typy proste



Źródło:
*XML Enhancements to Java™
(XJ).
User Manual for Release
Version 1.0*
<http://www.research.ibm.com/xj/doc/XJmanual/XJmanual.html>

Wbudowane typy proste (wybrane)

- `string` – napis
- `normalizedString` – napis, w którym każdy biały znak jest podczas przetwarzania zastępowany przez spację
- `token` – napis, w którym każdy ciąg białych znaków jest podczas przetwarzania zastępowany przez jedną spację, zaś białe znaki na początku i końcu są usuwane
- `QName` – nazwa kwalifikowana
- `NCName` – nazwa bez dwukropka
- `base64Binary` – dane binarne zapisane w kodowaniu Base64
- `hexBinary` – dane binarne zapisane szesnastkowo

Wbudowane typy proste (wybrane)

<i>Typ</i>	<i>Poprawne wartości</i>	
decimal	12.3	+000012.300
float, double	+24.3e-3	12
	NaN	-INF
QName	os:osoba	osoba
date	1968-04-02	1968-04-02-05:00
	1968-04-02Z	-0045-02-02
time	13:20:00.887	13:30:00-05:00
dateTime	1968-04-02T13:20:00.887	
gYearMonth	1968-04	
gMonthDay	--04-02	
duration	P2Y6M5DT12H35M30S	

Wyprowadzanie typów

- Wyprowadzanie typów prostych:
 - ograniczanie,
 - tworzenie list,
 - tworzenie unii.
- Wyprowadzanie typów złożonych:
 - ograniczanie,
 - rozszerzanie:
 - typów prostych,
 - typów złożonych.

Ograniczanie typów prostych

- Aspekty (ang. *facets*):
 - `minExclusive`, `minInclusive`,
 - `maxExclusive`, `maxInclusive`,
 - `length`, `minLength`, `maxLength`,
 - `totalDigits`, `fractionDigits`,
 - `enumeration`,
 - `pattern`,
 - `whiteSpace`.
- Tylko `pattern` i `enumeration` można użyć wielokrotnie w jednej definicji typu.

Ograniczanie typów prostych – przykłady

```
<xsd:simpleType name="NumerLottoTyp">  
  <xsd:restriction base="xsd:integer">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="49"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="DokumentTyp">  
  <xsd:restriction base="xsd:token">  
    <xsd:enumeration value="dowód osobisty"/>  
    <xsd:enumeration value="paszport"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="NIPTyp">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="\d{3}-\d{3}-\d{2}-\d{2}"/>  
    <xsd:pattern value="\d{3}-\d{2}-\d{2}-\d{3}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Ograniczanie typów prostych

- Przestrzeń wartości typu wyprowadzonego musi być podzbiorem przestrzeni wartości typu bazowego.
- Typy wbudowane mają określone niektóre aspekty, np.: typ `byte`:
 - `minInclusive`: -128,
 - `maxInclusive`: 127.

- Przykład – niepoprawne wyprowadzenie:

```
<xsd:simpleType name="ExtendedByte">  
  <xsd:restriction base="xsd:byte">  
    <xsd:minInclusive value="-256"/>  
    <xsd:maxInclusive value="255"/>  
  </xsd:restriction>  
</xsd:simpleType>
```


Aspekt whiteSpace

- `preserve` – wszystkie białe znaki są pozostawiane bez zmian
- `replace` – każdy biały znak jest podczas przetwarzania zastępowany przez spację
- `collapse` - każdy biały znak jest podczas przetwarzania zastępowany przez spację, a następnie każdy ciąg spacji jest zastępowany przez jedną spację, zaś spacje na początku i na końcu są usuwane

Tworzenie list

```
<xsd:simpleType name="NumerLottoTyp">  
  <xsd:restriction base="xsd:integer">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="49"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="NumeryLottoTyp">  
  <xsd:list itemType="NumerLottoTyp"/>  
</xsd:simpleType>
```

```
<xsd:simpleType name="KuponLottoTyp">  
  <xsd:restriction base="NumeryLottoTyp">  
    <xsd:length value="6"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Tworzenie list – inaczej

```
<xsd:simpleType name="KuponLottoTyp">
  <xsd:restriction>
    <xsd:simpleType>
      <xsd:list>
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="49"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:list>
    </xsd:simpleType>
    <xsd:length value="6"/>
  </xsd:restriction>
</xsd:simpleType>
```

Tworzenie unii

```
<xsd:simpleType name="RozmiarLiczbowyTyp">  
  <xsd:restriction base="xsd:integer">  
    <xsd:minInclusive value="2"/>  
    <xsd:maxInclusive value="18"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="RozmiarSMLTyp">  
  <xsd:restriction base="xsd:token">  
    <xsd:enumeration value="S"/>  
    <xsd:enumeration value="M"/>  
    <xsd:enumeration value="L"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="RozmiarTyp">  
  <xsd:union memberTypes="RozmiarLiczbowyTyp  
                          RozmiarSMLTyp"/>  
</xsd:simpleType>
```

Tworzenie unii – inaczej

```
<xsd:simpleType name="RozmiarTyp">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="2"/>
        <xsd:maxInclusive value="18"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="S"/>
        <xsd:enumeration value="M"/>
        <xsd:enumeration value="L"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

Rozszerzanie typów

- Wartości typu bazowego nie muszą być poprawnymi wartościami typu wyprowadzonego.
- Rozszerzanie zawartości prostej – dodawanie atrybutów do:
 - typu prostego,
 - typu złożonego o zawartości prostej.
- Rozszerzanie zawartości złożonej:
 - dodawanie do typu bazowego dodatkowych elementów i/lub atrybutów,
 - elementy dodawane w rozszerzeniu występują zawsze na końcu, po elementach zadeklarowanych w typie bazowym.

Rozszerzanie zawartości prostej

```
<xsd:complexType name="MasaTyp">  
  <xsd:simpleContent>  
    <xsd:extension base="xsd:decimal">  
      <xsd:attribute name="jm" type="xsd:string"/>  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

```
<xsd:complexType name="MasaNiedokładnaTyp">  
  <xsd:simpleContent>  
    <xsd:extension base="MasaTyp">  
      <xsd:attribute name="dokładność"  
        type="xsd:decimal"/>  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```

Rozszerzanie zawartości złożonej

```
<xsd:complexType name="OsobaTyp">  
  <xsd:sequence>  
    <xsd:element name="imie" type="xsd:string"/>  
    <xsd:element name="nazwisko" type="xsd:string"/>  
  </xsd:sequence>  
</xsd:complexType>
```

```
<xsd:complexType name="OsobaZDokumentemTyp">  
  <xsd:complexContent>  
    <xsd:extension base="OsobaTyp">  
      <xsd:choice>  
        <xsd:element name="dowód-os"  
          type="DowódTyp"/>  
        <xsd:element name="paszport"  
          type="PaszportTyp"/>  
      </xsd:choice>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```


Ograniczanie typów złożonych

- Przestrzeń wartości typu wyprowadzonego musi być podzbiorem przestrzeni wartości typu bazowego.
- Dozwolone operacje:
 - ograniczenie zawartości prostej (tak jak typu prostego),
 - ograniczenie atrybutu:
 - ograniczenie typu atrybutu,
 - zmiana atrybutu opcjonalnego na wymagany (`required`) lub zabroniony (`prohibited`),
 - dodanie, zmiana lub usunięcie wartości domyślnej,
 - dodanie wartości ustalonej, jeśli jej nie było.
 - ograniczenie modelu zawartości, np.:
 - ściślejsze ograniczenia liczebności (`minOccurs`, `maxOccurs`),
 - usunięcie elementów opcjonalnych w grupach `sequence` i `all`,
 - wybranie podzbioru elementów w grupie `choice`,
 - ograniczenie typu poszczególnych podelementów.

Ograniczanie zawartości prostej i atrybutów

```
<xsd:complexType name="MasaNiedokladnaTyp">
  <xsd:simpleContent>
    <xsd:extension base="xsd:decimal">
      <xsd:attribute name="jm" type="xsd:string"/>
      <xsd:attribute name="dokładność"
        type="xsd:decimal"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

```
<xsd:complexType name="MasaDoUniesieniaTyp">
  <xsd:simpleContent>
    <xsd:restriction base="MasaNiedokladnaTyp">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="50"/>
      <xsd:attribute name="dokładność" use="prohibited"/>
      <xsd:attribute name="jm" fixed="kg"/>
    </xsd:restriction>
  </xsd:simpleContent>
</xsd:complexType>
```

Ograniczanie zawartości złożonej

```
<xsd:complexType name="BazowyTyp">
  <xsd:sequence>
    <xsd:element name="a" type="xsd:string"
      minOccurs="3" maxOccurs="7"/>
    <xsd:element name="b" type="xsd:string"
      minOccurs="0"/>
    <xsd:element name="c" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OgraniczonyTyp">
  <xsd:complexContent>
    <xsd:restriction base="BazowyTyp">
      <xsd:sequence>
        <xsd:element name="a" type="xsd:string"
          minOccurs="4" maxOccurs="6"/>
        <xsd:element name="c" type="xsd:string"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

Typy

- 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.

Inne metody modularyzacji schematów

- Grupy do wielokrotnego wykorzystania (*reusable groups*):
 - nazwane grupy modeli,
 - grupy atrybutów.
- Grupy zamienne (*substitution groups*).

Grupy do wielokrotnego wykorzystania

```
<xsd:group name="ImięNazwisko">
  <xsd:sequence>
    <xsd:element name="imie" type="xsd:string"/>
    <xsd:element name="nazwisko" type="xsd:string"/>
  </xsd:sequence>
</xsd:group>
```

```
<xsd:attributeGroup name="Biurokracja">
  <xsd:attribute name="NIP" type="NIPTyp"/>
  <xsd:attribute name="PESEL" type="PESELTyp"/> </
xsd:attributeGroup>
```

```
<xsd:complexType name="OsobaTyp">
  <xsd:sequence>
    <xsd:group ref="ImięNazwisko"/>
    <xsd:element name="obywatelstwo"
      type="xsd:token"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="Biurokracja"/> </
xsd:complexType>
```

Grupy zamienne

```
<xsd:element name="katalog">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="produkt"
        maxOccurs="unbounded" />
    <xsd:sequence>
      <xsd:complexType>
    </xsd:element>
  <xsd:element name="produkt" type="ProduktTyp" />
  <xsd:element name="komputer" type="KomputerTyp"
    substitutionGroup="produkt" />
  <xsd:element name="ubranie" type="UbranieTyp"
    substitutionGroup="produkt" />
```

Grupy zamienne – egzemplarz

```
<katalog>  
  <produkt>...</produkt>  
  <komputer>...</komputer>  
  <ubranie>...</ubranie>  
  <produkt>...</produkt>  
</katalog>
```


Więzy integralności

- Więzy integralności (*identity constraints*):
 - `unique` – wartości w ramach określonego zakresu muszą być unikatowe,
 - `key` - wartości w ramach określonego zakresu muszą być unikatowe i nie mogą być puste,
 - `keyref` – odwołanie do wartości wskazanego klucza.
- Zalety w porównaniu z atrybutami typu `ID` – `IDREF`:
 - można tworzyć wiele niezależnych od siebie ograniczeń,
 - odwołanie wskazuje konkretny klucz,
 - więzy integralności mogą być nałożone na kombinację wartości,
 - więzy integralności mogą być nałożone na zawartości elementów.

Więzy integralności

```
<xsd:element name="zamówienia-i-faktury"
              type="ZamówieniaFakturyTyp">
  <xsd:unique name="fakturaId">
    <xsd:selector xpath="faktury/faktura"/>
    <xsd:field xpath="@nr"/>
    <xsd:field xpath="rok"/>
  </xsd:unique>
  <xsd:keyref name="zamówienieRef"
              refer="zamówienieId">
    <xsd:selector
      xpath="faktury/faktura/do-zamówienia"/>
    <xsd:field xpath="@nr"/>
  </xsd:keyref>
  <xsd:key name="zamówienieId">
    <xsd:selector xpath="zamówienia/zamówienie"/>
    <xsd:field xpath="numer"/>
  </xsd:key>
</xsd:element>
```