



Interneto technologijos

XML Schema
Sudėtingi tipai su
paprastu turiniu

Turinio rūšys, XML Schema tipai

	Žymių/atributų turinio rūšys			
	Mišrus (mixed)	Sudėtingas (complex)	Paprastas (simple)	Tuščias (empty)
Gali turėti vaikinių žymių	Taip	Taip	Ne	Ne
Gali turėti tekstą	Taip	Ne	Taip	Ne

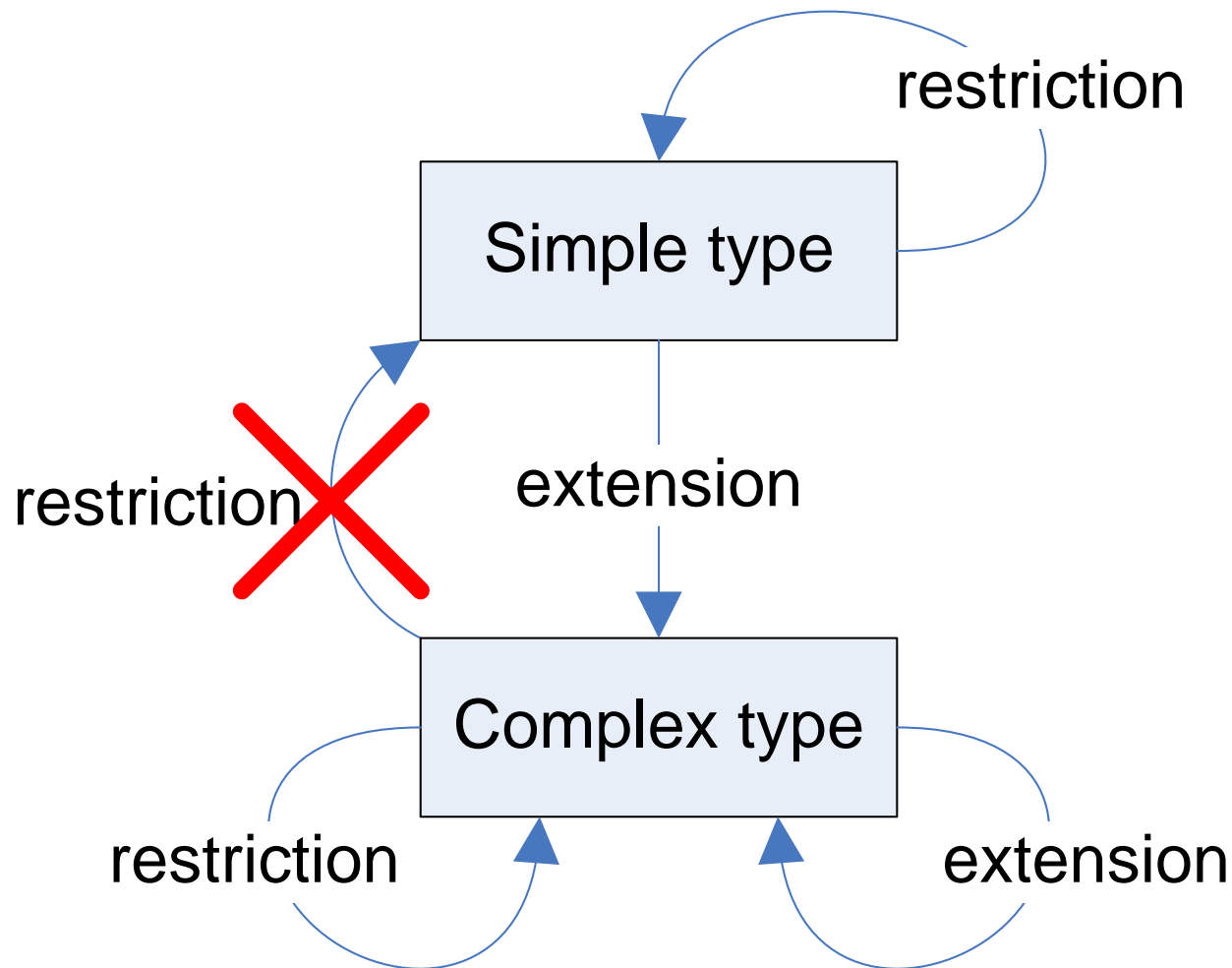
- XML Schema turi dvi *tipų rūšis* :
 - Paprasti tipai (simple types): naudojami paprasto (be atributų) ir tuščio turinio žymių/atributų tipų apibrėžimui
 - Nusako teksto fragmento formatą
 - **Sudėtingi tipai** (complex types): gali būti naudojami visų rūšių turinių apibrėžimui
- Nepainiokite sudėtingo turinio su sudėtingu tipu!



Sudėtingi tipai – įvadas

- Sudėtingo tipo deklaracija susideda iš:
 - atributų deklaracijų (nebūtina)
 - turinio rūšies deklaracijos (būtina)
- **Pastaba:** Paprasti tipai negali apibrėžti atributų, t.y., jei žymei priskirtas paprastas tipas, tai joje negali būti jokių atributų
 - Jei reikia bent vieno atributo, reikia skelbti sudėtingą tipą
- Kiekviena sudėtingo tipo deklaracija yra:
 - arba kito sudėtingo tipo apribojimas
 - arba praplėtimas:
 - paprasto tipo
 - sudėtingo tipo

Sudėtingų tipų su paprastu turiniu išvedimo būdai





Sudėtingo tipo struktūra

```
<complexType
```

```
  name = NCName
```

```
  mixed = boolean : false>
```

```
  Content:
```

```
    simpleContent | complexContent
```

```
</complexType>
```

Sudėtingo tipo turinio rūšies deklaracija

Žymių/atributų turinio rūšys			
Mišrus (mixed)	Sudėtingas (complex)	Paprastas (simple)	Tuščias (empty)
complexContent ir atributas mixed=true	complexContent	simpleContent	complexContent, simpleContent

- Visi šie variantai bus aptariami toliau skaidrėse
 - Paprastas
 - Sudėtingas
 - Mišrus
 - Tuščias

Paprasto turinio struktūra

```
<simpleContent>
  Content: (annotation?, (restriction | extension))
</simpleContent>
<restriction base = QName>
  Content: ( annotation?,
              ( simpleType?,
                ( minExclusive | minInclusive |
                  maxExclusive | maxInclusive |
                  totalDigits | fractionDigits |
                  length | minLength | maxLength |
                  enumeration | whiteSpace | pattern
                ) *
              ) ?,
              (attribute | attributeGroup) *
            )
</restriction>
```

Jei bazinis tipas neturi paprasto tipo (pvz., anyType neturi), tai simpleType privalomas

Bazinis tipas gali būti arba sudėtingas tipas su paprastu turiniu, arba su sudėtingu turiniu ir sąlyga minOccurs=0

Paprasto turinio struktūra (2)

`<extension base = QName>`

Bazinis tipas gali būti paprastas tipas arba sudėtingas tipas su paprastu turiniu

Content:

`(attribute | attributeGroup)*`

`</extension>`

Praplėtimu galima pridėti naujų atributų (ir nieko daugiau)



Praplėtimo pavyzdžiai

- Sukuriame tipą "tekstasSuKalba" su atributu:

```
<xs:complexType name="tekstasSuKalba">  
  <xs:simpleContent>  
    <xs:extension base="xs:string">  
      <xs:attribute name="kalba" type="xs:language"/>  
    </xs:extension>  
  </xs:simpleContent>  
</xs:complexType>
```

- Tipą "tekstasSuKalba" praplečiame dar vienu atributu:

```
<xs:complexType name="tekstasSuKalbaIrData">  
  <xs:simpleContent>  
    <xs:extension base="tekstasSuKalba">  
      <xs:attribute name="data" type="xs:date"/>  
    </xs:extension>  
  </xs:simpleContent>  
</xs:complexType>
```



Apribojimo pavyzdžiai

- Apribokime tipo "tekstasSuKalbaIrData" teksto ilgį iki 255 simbolių:

```
<xs:complexType name="apribotoIlgioTekstas">
  <xs:simpleContent>
    <xs:restriction base="tekstasSuKalbaIrData">
      <xs:maxLength value="255"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```



Apribojimo pavyzdžiai: atributų uždraudimas

- Jei norime uždrausti bazinio tipo atributą, naudojame `use="prohibited"`:

```
<xs:complexType name="PastabaTipas">
  <xs:simpleContent>
    <xs:restriction base="tekstasSuKalbaIrData">
      <xs:attribute name="data" use="prohibited"/>
```



Apribojimo pavyzdžiai: atributo tipo apribojimas

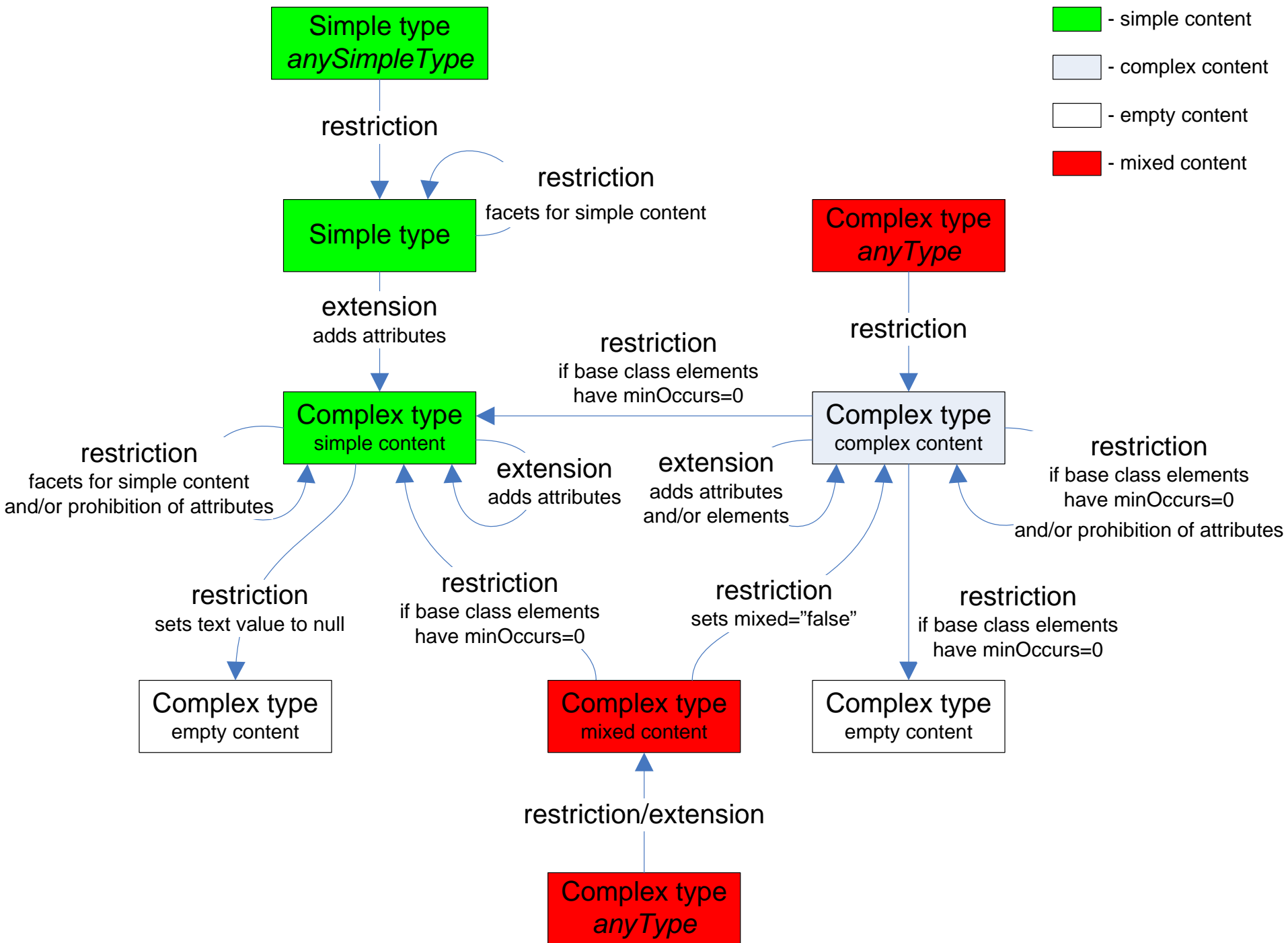
- Tipų apribojimas bus ir tokiu atveju, jei apribosime atributo turinio tipą:

```
<xs:complexType name="NaujasTipas">
  <xs:simpleContent>
    <xs:restriction base="tekstasSuKalba">
      <xs:attribute name="kalba">
        <xs:simpleType>
          <xs:restriction base="xs:language">
            <xs:enumeration value="en"/>
            <xs:enumeration value="lt"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```



Apibendrinimas

- Naujų sudėtingų tipų su paprastu turiniu išvedimas ...
 - *praplėtimu* gali pridėti naujus atributus (žymių/atributų tipų keisti negali)
 - *apribojimu* gali apriboti žymės/atributo tipą. Taip pat gali uždrausti bazinio tipo atributų naudojimą



Apribojimo principas

- **Apribotas tipas negali prieštarauti baziniam tipui**
 - T.y., visi žymių/atributų turiniai, kurie atitinka apribotą tipą, turi atitikti ir bazinį tipą
- Taip apriboti tipo **negalima**:

```
<xs:simpleType name="keistasSkaicius">  
  <xs:restriction base="xs:integer">  
    <xs:pattern value="[a-c]{5}" />  
  </xs:restriction>  
</xs:simpleType>
```



Ne skaičius!!!

Apribojimo principo pavyzdys

```
- <xs:complexType name="tekstasSuAtributais">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="kalba" type="xs:language"/>
      <xs:attribute name="pastaba" type="xs:string"
        use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

- Negalima uždrausti naudoti atributą pastaba (kalba atributą – galima, nes use reikšmė pagal nutylėjimą yra optional):

```
<xs:complexType name="tekstasBeAtributu">
  <xs:simpleContent>
    <xs:restriction base="tekstasSuAtributais">
      <xs:attribute name="kalba" use="prohibited"/>
      <xs:attribute name="pastaba" use="prohibited"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```





Pavyzdys

- Tarkime, turime paprastą tipą "Pavadinimas":

```
<xs:simpleType name="PavadinimoTipas">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="30" />  
  </xs:restriction>  
</xs:simpleType>
```

- Norime jam:

- pridėti atributą (praplėtimas – extension)
 - Rezultate gausime sudėtingą tipą su paprastu turiniu
- nurodyti apribojimą (restriction) – minimalų pavadinimo ilgį padaryti 10



Pavyzdys

- Kadangi reikia ir praplėtimo, ir apribojimo, tai vienu "žingsniu" mes to nepadarysime
 - Pirmu žingsniu reikia sukurti tarpinį tipą, kuris mūsų tipą "Pavadinimas" **praplės** atributu
 - Antru žingsniu šį tarpinį tipą apribosime, nurodydami **ribojimą** to turiniui
- Pastaba: Negalima apribojimo būdu iš paprasto tipo sukurti sudėtingą tipą
 - "mažindami" mažą didesnio gauti negalime 😊



Pavyzdys

■ Tarpinis tipas:

```
<xs:complexType name="PavadinimasSuAtributu">
  <xs:simpleContent>
    <xs:extension base="PavadinimoTipas">
      <xs:attribute name="kalba"
                    type="xs:language" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```



Pavyzdys

■ Galutinis tipas:

```
<xs:complexType name="ApribotasPavSuAtributu">  
  <xs:simpleContent>  
    <xs:restriction base="PavadinimasSuAtributu">  
      <xs:minLength value="10" />  
    </xs:restriction>  
  </xs:simpleContent>  
</xs:complexType>
```