

Programowanie komunikacji człowiek-komputer

dr inż. Joanna Ochelska-Mierzejewska

XLink ang. XML Linking Language

pol. język odnośników XML

Odsyłacze hipertekstowe

- Odsyłacze HTML-owe
 - Łączą dwa dokumenty; źródło i cel linku,
 - Źródłem linku jest zawsze element opisujący link (<A>,)
- Wady odsyłaczy HTML-owych
 - Nie potrafią odzwierciedlać różnych złożonych relacji występujących pomiędzy dokumentami, które łączą
 - Łączą tylko dwa dokumenty
 - Są wbudowane w dokument, przez co nie mogą być przechowywane w osobnych, zewnętrznych plikach, bazach
 - Wymagają znacznego nakładu pracy związanego z ciągłą i kłopotliwą aktualizacją
 - Mogą wskazywać na całe dokumenty lub tylko na konkretne punkty dokumentów

XLink

- XLink — rozszerzona koncepcja powiązań
 - Informacja o linkach zapisana w dowolnym elemencie
 - Niestatna nazwa elementu
 - Istotne atrybuty z przestrzenią nazw XLink
 - Więcej niż dwa końce linku (hiperlink → relacja)
 - Możliwość reprezentacji linku poza wiązanymi zasobami
- Status
 - korzenie historyczne: HyTime
 - XLink 1.0 — rekomendacja W3C, czerwiec 2001
 - XLink 1.1 — wersja robocza (aktualizacja w 2008)

Terminologia

- **Zasób** (ang. resource) — dowolna adresowalna jednostka informacji lub usługa (plik, program, wynik kwereń)
- **Dowiązanie** (ang. link) — wyrażona jawnie, przy pomocy **elementu wiążącego** (ang. linking element), relacja pomiędzy zasobami **uczestniczącymi** (ang. participate) w dowiązaniu
- **Przejście** (ang. traversal) — użycie pary zasobów (źródłowego — starting i docelowego — ending) połączonej dowiązaniem
- **Łuk** (ang. arc) — informacja o przejściu między zasobami oznaczonymi dwoma etykietami (w określonym kierunku)
 - **Łuk wychodzący** (ang. outbound) — z zasobu lokalnego do zewnętrznego
 - **Łuk wchodzący** (ang. inbound) — z zewnętrznego do lokalnego
 - **Łuk niezależny** (ang. third party) — między zewnętrznymi
- UWAGA: zasób jest traktowany jako **zewnętrzny** (ang. remote) zawsze, gdy jest adresowany za pomocą URI (nawet, gdy znajduje się w tym samym dokumencie czy elemencie linkującym, co posługujący się nim link)

Dowiązania XLink

- Link prosty (ang. simple link)
 - To link wychodzący
 - Wiąże dokładnie dwa zasoby: lokalny z zewnętrznym
 - Zawiera dokładnie jeden łuk pomiędzy zasobami
- Link rozszerzony (ang. extended link)
 - Wiąże dowolną liczbę zasobów lokalnych i zewnętrznych
 - Łuki opisują sposoby przechodzenia pomiędzy zasobami
 - Określa rolę zasobów uczestniczących w linku
 - Określa rolę łuków

7

Link prosty – przykład

```

<artykuł xmlns:xlink="http://www.w3.org/1999/xlink">
  <autor>Diego A. Golombeck</autor>
  <tytuł>Pozytywny wpływ <sponsor xlink:type="simple"
    xlink:href="http://www.example.com/
    shop.xml#wisagras">winygry</sponsor>
    na neutralizowanie zespołu naglej zmiany
    strefy czasowej u chomików
  </tytuł>
</artykuł>

<linkProsty xlink:href="http://edu.ics.p.lodz.pl"
  xlink:type="simple">
  Strona Instytutu Informatyki
</linkProsty>

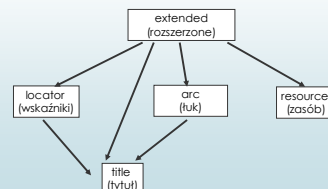
<linkProsty xlink:href="#ksiazki" xlink:type="simple">
  Strona Instytutu Informatyki
</linkProsty>

```

8

Link rozszerzony

- Odśylacz rozszerzony może składać się z następujących podelementów, występujących w dowolnej kolejności



9

Link rozszerzony – przykład

```

<wydział xlink:type="locator" xlink:label="ftima"
  xlink:href="http://www.ics.p.lodz.pl/ftima.xml" />

<wydział xlink:type="locator" xlink:label="ftima"
  xlink:href="ftima.xml" />

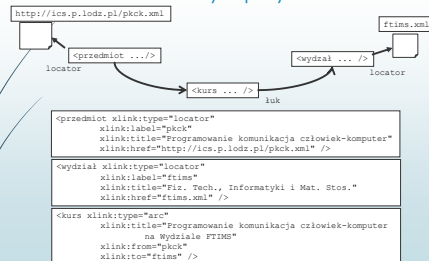
<przedmiot xlink:type="resource" xlink:label="PRCK">
  Ciepła Osoba </info>
  <nazwisko> Ochłaska-Mierzejewska </nazwisko>
</przedmiot>

<rodzina xlink:type="extended">
  <mls:rola xlink="http://www.w3.org/1999/xlink">
    <osoba xlink:type="locator" xlink:href="httpio.xml"
      xlink:label="rodzina" xlink:title="Rodzina"/>
    <osoba xlink:type="locator" xlink:href="wiesia.xml"
      xlink:label="rodzina" xlink:title="Rodzina"/>
    <osoba xlink:type="locator" xlink:href="rony.xml"
      xlink:label="dziecko" xlink:title="Rodzina"/>
    <osoba xlink:type="locator" xlink:href="nikana.xml"
      xlink:label="dziecko" xlink:title="Rodzina"/>
    <osoba xlink:type="locator" xlink:href="oliwia.xml"
      xlink:label="dziecko" xlink:title="Rodzina"/>
    <link xlink:type="arc" xlink:from="rodzina"
      xlink:to="dziecko"/>
  </mls:rola>
</rodzina>

```

10

Link rozszerzony – przykład



11

Atrybuty w linkach rozszerzonych

- type – rola elementu w linku (simple, extended, locator, arc, resource, title, none)
- href – URI zasobu zewnętrznego
- role – abstrakcyjny identyfikator roli zasobu w powiązaniu (URI)
- arcrole – j.w. ale dla pojedynczego łuku między zasobami
- title – etykieta tekstowa zasobu, łuku (opcjonalna)
- show – jak prezentować (new, replace, embed, other, none)
- actuate – kiedy aktywować (onLoad, onRequest, other, none)
- label – etykieta zasobu używana jako identyfikator (do użycia we from i to, nie muszą być unikalne)
- from, to – wskazanie (w łuku) na etykieta zasobu

12

Atrybuty w linkach rozszerzonych

Atrybut	Typ elementu					
type	simple	extended	locator	arc	resource	title
href	+	+	+	+	+	+
role	o	+				
arcrole	o	o	o		o	
title	o			o		
show	o	o	o	o	o	
actuate	o			o		
label	o			o		
from				o	o	
to					o	

+- musi
o - może

SVG

ang. Scalable Vector Graphics

13

Grafika wektorowa

14

- Składa się z prostych linii i krzywych zdefiniowanych przez obiekty matematyczne – wektory
- Wektory opisują grafikę w kategoriach geometrycznych
- Nie zależy od rozdzielczości
- Nadaje się do tworzenia grafiki tekstowej, wizerunków, emblematów, znaków firmowych, gdzie linie muszą być ostre i wyraźne

SVG

15

- SCALABLE - dający się modyfikować przestrzennie (dwuwymiarowo)
- VECTOR - czyli opisywany za pomocą definiowania linii, krzywych i punktów
- GRAPHICS - składnia SVG dysponuje bogatym wachlarzem określeń opisujących wektorową lub mieszaną (rastrowo/wektorową) grafikę
 - „wypełnia przestrzeń, która powstała wskutek ograniczonej możliwości integrowania grafik zarówno przez HTML, jak i XML”
- Powstał w 1999r. jako część składni języka XML
- Grafika, która powstaje w wyniku interpretacji kodu jest w pełni skalowalna, a plik z kodem jest mały
- Dodatkowym atutem jest doskonała jakość wydruku strony zawierającej elementy SVG

Struktura dokumentu

16

Witaj świecie

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

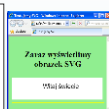
<svg width="350px" height="80px" xmlns="http://www.w3.org/2000/svg">
  <desc> Opis co dokument zawiera </desc>
  <title> Pierwszy dokument SVG </title>
  <!-- Komentarze tak jak w HTML -->
  <g id="tekst1">
    <text x="120" y="40" font-face="Verdana" font-size="24">
      Witaj świecie</text>
    </g>
  </svg>
```

SVG a (X)HTML

17

```
<html>
  <head>
    <title> Testujemy SVG </title>
    </head>
    <body bgcolor="lightgreen">
      <div color="yellow">
        <div align="center"> Zaczynam wyświetlać obrazek SVG </div>
        <div color="yellow">
          <EMBED src="zys1.svg" width="350" height="80">
            type="image/svg+xml"
            pluginspage="http://www.adobe.com/svg/viewer/install/"
          </div>
        </div>
      </body>
    </html>

<html xmlns:svg="http://www.w3.org/2000/svg">
  <object id="AdobeSVG">
    <CLASSID "clsid:7815a80-c61-4bbf-866a-3d390a4b4e2">
  </object>
  <head>
    <title> SVG Example </title>
    </head>
  <body>
    <?import namespace="svg" implementation="#AdobeSVG"?>
    <svg:svg width="350" height="80" xmlns="http://www.w3.org/2000/svg">
      <svg:svg id="tekst1">
        <svg:text x="120" y="40" font-face="Verdana" font-size="24">Witaj świecie</svg:text>
      </svg:svg>
    </svg:svg>
  </body>
</html>
```



Podstawowe kształty – linie

18

- Narysowanie linii polega na połączeniu dwóch punktów – w tym celu stosujemy element <line>
- Atrybuty
 - x1 y1 - położenie punktu, z którego zaczynamy rysować
 - x2 y2 - współrzędne punktu, w którym kończymy rysowanie
 - stroke - określamy kolor linii
 - stroke-width - określamy grubość linii

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
  "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="800" height="800" xmlns="http://www.w3.org/2000/svg">

  <g stroke="red">
    <line x1="100" y1="300" x2="300" y2="100" stroke-width="5" />
    <line x1="300" y1="300" x2="300" y2="100" stroke="yellow" stroke-width="10" />
    <line x1="500" y1="300" x2="300" y2="100" stroke-width="15" />
    <line x1="700" y1="300" x2="300" y2="100" stroke="blue" stroke-width="20" />
  </g>
</svg>
```




19


Podstawowe kształty - prostokąt

- Używamy elementu `<rect>`
- Atrybuty
 - `x1 y1` - położenie lewego górnego rogu prostokąta
 - `height i width` - wielkości obiektu; wysokości i szerokości
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie
 - `rx i ry` - zaokrąglenie rogów kwadratu w stopniach
 - Gdy zastosujemy tylko parametry `rx`, wtedy `ry` automatycznie przyjmie wartość identyczną dla `rx`

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <rect x="200" y="100" width="100" height="200" fill="blue" stroke="red" stroke-width="10" />
</svg>
```



```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <rect x="40" y="40" width="20" height="20" rx="10" ry="10" fill="blue" stroke="black" stroke-width="5" />
</svg>
```



20

Podstawowe kształty – wielokąt

- Służy element `<polygon>`
- Atrybuty
 - `points` - określa poszczególne punkty rogów figury
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="120" height="100" viewBox="0 0 120 100" xmlns="http://www.w3.org/2000/svg">
  <rect x="1" y="1" width="118" height="98" fill="none" stroke="blue" stroke-width="2" />
  <polygon fill="red" stroke="blue" stroke-width="10"
    points="350,75 375,161 469,161 397,215 423,301 350,250 277,301 303,215
    231,161 321,161" />
  <polygon fill="lime" stroke="blue" stroke-width="10"
    points="830,75 850,137.5 858,262.5 850,325 742,242.6 742,137.5" />
</svg>
```



21

Podstawowe kształty – krzywa


- Polega na podawaniu kolejnych współrzędnych punktów, które mają zostać połączone – element `<polyline>`
- Atrybuty
 - `points` - określa poszczególne punkty rogów figury
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie

22

Podstawowe kształty – krzywa

- Polega na podawaniu kolejnych współrzędnych punktów, które mają zostać połączone – element `<polyline>`
- Atrybuty
 - `points` - określa poszczególne punkty rogów figury
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="120" height="100" viewBox="0 0 120 100" xmlns="http://www.w3.org/2000/svg">
  <desc>Złoty/deco</desc>
  <polyline fill="none" stroke="blue" stroke-width="10"
    points="50,375 150,375 150,325 250,325 250,375 350,375 350,250 450,250 450,375 550,375
    550,175 650,175 650,375 750,375 750,100 850,100 850,375 950,375 950,25 1050,25
    1050,375 1150,375" />
</svg>
```




23

Podstawowe kształty – koło

- Element `<circle>`
- Atrybuty
 - `cx cy` - współrzędne środka
 - `r` - promień
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="120" height="100" viewBox="0 0 120 100" xmlns="http://www.w3.org/2000/svg">
  <circle cx="600" cy="200" r="100" fill="red" stroke="blue" stroke-width="10" />
</svg>
```

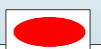


24

Podstawowe kształty – elipsa

- Element `<ellipse>`
- Atrybuty
 - `cx cy` - współrzędne środka
 - `rx ry` - promienie
 - `stroke` - określamy kolor linii
 - `stroke-width` - określający grubość linii
 - `fill` - wypełnienie

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="120" height="100" viewBox="0 0 120 100" xmlns="http://www.w3.org/2000/svg">
  <ellipse cx="900" cy="200" rx="200" ry="100" fill="red" />
</svg>
```



25

Teksty

- Element `<text>` `</text>`
- Atrybuty
 - `x` i `y` – położenie
 - `fill` – kolor tekstu lub poprzez CSS
 - rozmiar, rodzaj czcionki – poprzez parametry podobne do CSS lub CSS

Prosty tekst w SVG

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <text x="55" y="55" font-family="Verdana" font-size="55" fill="blue">
    Prosty tekst w SVG
  </text>
  <rect x="1" y="1" width="700" height="100" fill="none" stroke="blue" stroke-width="2" />
</svg>
```

26

Teksty

- Element `<text>` `</text>`
- Atrybuty
 - `x` i `y` – położenie
 - `fill` – kolor tekstu lub poprzez CSS
 - rozmiar, rodzaj czcionki – poprzez parametry podobne do CSS lub CSS

Inne formatowanie tekstu

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400" height="100" xmlns="http://www.w3.org/2000/svg">
  <text x="55" y="55" style="fill:rgb(70,0,200); font-family:Arial CE; font-size:24">
    Inne formatowanie tekstu
  </text>
</svg>
```

27

Teksty

- Element `<tspan>` `</tspan>` pozwala na dowolnie formatowanie fragmentów tekstu
- Rozsuwanie liter, zmianę położenie wyrazów, pogrubienie/pochylenie czcionki

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg">
  <g font-family="Verdana" font-size="45">
    <text x="200" y="150" fill="blue">
      Prosty
      <tspan dx="2em" dy="-50" font-weight="bold" fill="red">
        tekst
      </tspan>
      <tspan dy="100">
        w SVG.
      </tspan>
    </text>
  </g>
</svg>
```

Prosty tekst w SVG.

28

Teksty

- Element `<tspan>` pozwala na dowolnie formatowanie fragmentów tekstu
- Rozsuwanie liter, zmianę położenie wyrazów, pogrubienie/pochylenie czcionki

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg">
  <g font-family="Verdana" font-size="45">
    <text fill="rgb(0,125,255)" font-size="45">
      <tspan x="300 350 400 450 500 550 600 650 700" y="100">
        Rozsuwanie
        literki
      </tspan>
    </text>
  </g>
  <rect x="1" y="1" width="998" height="298" fill="none" stroke="blue" stroke-width="2" />
</svg>
```

Rozsuwanie literki

29

Teksty

- Atrybut `text-decoration` dodaje podkreślenie, przekreślenie lub „nadkreślenie” tekstu

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg">
  <g font-family="Verdana" font-size="45">
    <text x="100" y="150" font-size="45" text-decoration="underline">
      Tekst podkreślony
    </text>
    <text x="100" y="230" font-size="45" text-decoration="overline">
      Tekst z linią nadk.
    </text>
  </g>
  <rect x="1" y="1" width="998" height="298" fill="none" stroke="blue" stroke-width="2" />
</svg>
```

Tekst podkreślony
Tekst z linią nadk.

30

Teksty

- Element `<path>` umożliwia stworzenie ścieżki (na początku dokumentu), po której deklarujemy
- Odwolanie do ścieżki w elemencie `<text>`

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <path id="MyPath">
      d="M 100 200
        C 200 150 300 0 400 100
        S 500 200 600 300 700 200
        C 800 100 900 150 900 100" />
    </defs>
    <use xlink:href="#MyPath" fill="none" stroke="red" />
    <text font-family="Verdana" font-size="45" fill="blue">
      Jest to tekst pisany po stworzonej ścieżce
    </text>
  </svg>
```

Jest to tekst pisany po stworzonej ścieżce

31

Wypełnienia

- Wypełnianie obszaru jednolitym kolorem poprzez atrybut `fill`
 - Wartość tego atrybutu: nazwa np. blue lub w RGB rgb(0, 0, 255) lub #0000FF
- Wypełnienie obszaru przy pomocy gradientu liniowego (ang. *linear gradient*) lub radialnego (ang. *radial gradient*)
 - Gradient definiujemy w elemencie `<defs>`

32

Gradient liniowy

- Definicja w elemencie `<linearGradient>`
- Atrybuty
 - `id` – definicja nazwy gradientu
 - `offset` – atrybut określający granicę koloru, czyli miejsce jego wystąpienia (określamy w procentach)
 - `stop-color` – nazwa koloru w tym miejscu

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400px" height="200px" viewBox="0 0 400 200" preserveAspectRatio="xMinYMin meet">
  <title>three-stop gradient</title>
  <defs>
    <linearGradient id="three_stops">
      <stop offset="0%" style="stop-color: #ffcc00;"/>
      <stop offset="33.33%" style="stop-color: #006699;"/>
      <stop offset="100%" style="stop-color: #66cc99;"/>
    </linearGradient>
  </defs>
  <rect x="25" y="25" width="200" height="100"
        style="fill:url(#three_stops); stroke: black;"/>
</svg>
```

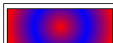


33

Gradient radialny

- Definicja w elemencie `<radialGradient>`
- Atrybuty
 - `cx` i `cy` – współrzędne środka zewnętrznego koła gradientu
 - `fx` i `fy` – współrzędne środka gradientu (gdy brak jest parametrów przyjmowane są wartości równe `cx` i `cy`)
 - `r` – promień
 - `id` – definicja nazwy gradientu
 - `offset` – atrybut określający granicę koloru, czyli miejsce jego wystąpienia (określamy w procentach)
 - `stop-color` – nazwa koloru w tym miejscu

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="800px" height="400px" viewBox="0 0 800 400" xmins="http://www.w3.org/2000/svg">
  <g>
    <defs>
      <radialGradient id="MyGradient" gradientUnits="userSpaceOnUse">
        <stop offset="0%" style="stop-color: red;"/>
        <stop offset="50%" style="stop-color: blue;"/>
        <stop offset="100%" style="stop-color: red;"/>
      </radialGradient>
    </defs>
    <rect fill="url(#MyGradient)" stroke="black" stroke-width="5"
          x="100" y="100" width="600" height="200"/>
  </g>
</svg>
```

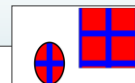


34

Wypełnienie własnym wzorem

- Element `<pattern>`

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="800px" height="400px" xmins="http://www.w3.org/2000/svg">
  <defs>
    <pattern id="Pattern" patternUnits="userSpaceOnUse">
      <path d="M 0 0 L 5 0 L 5 5 L 0 5 L 0 0" fill="red" stroke="blue" />
    </pattern>
  </defs>
  <rect fill="url(#Pattern)" stroke="black" x="500" y="15" width="200" height="200" />
  <ellipse fill="url(#Pattern)" stroke="black" stroke-width="5"
           cx="400" cy="200" rx="50" ry="70" />
</svg>
```



35

Transformacje

- Obiekty możemy transformować przez zmianę ich położenia, obroty, pochYLENIA
- Realizowane przez atrybut `<transform>`
- Wartości atrybutu
 - `translate(tx [tx])` – przesunięcie obiektu o `tx` – jednostek względem osi x i o `ty` – jednostek względem osi y
 - `scale(sx [sy])` – zmiana rozmiaru obiektu, jeśli `sy` nie jest podany, to przypuszczamy, że jest taki sam jak `sx` (np. wartość `<sx>` wynosi 0.5, to obiekt zostaje pomniejszony o 50% oryginalnej szerokości)
 - `rotate(rotate-angle)` – obrót obiektu o zadany kąt dookoła punktu centralnego (0,0) (np. jeżeli wartość `rotate-angle` wynosi 45, to obiekt obróci się o 45 stopni według wskazówek zegara)
 - `skewX(angle)` – pochyla element względem osi x o kąt `angle`
 - `skewY(angle)` – pochyla element względem osi y o kąt `angle`

36

Transformacje

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="120px" height="40px" viewBox="0 0 1200 400" xmins="http://www.w3.org/2000/svg">
  <rect x="1" y="1" width="1198" height="398" fill="none" stroke="blue" stroke-width="2" />
  <g transform="translate(300 200)">
    <ellipse rx="250" ry="100" fill="red" />
  </g>
  <ellipse transform="translate(900 200) rotate(-30)" rx="250" ry="100" fill="none"
           stroke="blue" stroke-width="20" />
</svg>
```



37

Filtry

- Tworzymy za pomocą elementu `<filter>` z parametrem `id` w elemencie `<defs>`
- Tworząc własne filtry używamy następujących elementów:
 - `<feBlend>` - służy do mieszania dwóch obiektów. Zestawia się z parametrami:
 - `in` - służy do przetwarzania, rozpraszania kolorów, luminancji
 - `in2` - służy do ewaluacji (koloru, luminancji, prędkości obiektu)
 - `mode` - określa rodzaj połączenia
 - `<feColorMatrix>` - służy do ramowania obiektów
 - `<feGaussianBlur>` - służy do rozmywania obiektów
 - `stdDeviation` - odległość między punktami
 - `preserveAspectRatio` - błądź obiektu punktowo
 - `result` - błądź obiektu punktowo
 - `<feOffset>` - służy do przesunięcia obiektu
 - `<feImage>` - służy do wstawiania obrazu
 - `<feMerge>` - służy do łączenia obiektów
 - `<feMorphology>` - służy do tworzenia kształtów
 - `<fePaint` - służy do tworzenia kolorów
 - `<feRadialGradient>` - służy do tworzenia gradientów
 - `<feRect` - służy do tworzenia prostokątów
 - `<feText` - służy do tworzenia tekstu
 - `<feTurbulence>` - służy do tworzenia falowania

38

Animacje

- Element `<animate>` zmienia atrybuty elementu, który chcemy zanimować
- Atrybuty (wspólne dla wszystkich elementów `<animate>`)
 - `attributeName` - nazwa atrybutu
 - `attributeType` - typ atrybutu (CSS, XML lub auto)
 - `begin` - czas rozpoczęcia zmiany
 - `end` - koniec zmiany atrybutu
 - `dur` - czas trwania zmiany
 - `fill` - parametr stanu końcowego (freeze, remove)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <rect x="10" y="10" width="80" height="80" fill="rgb(255,255,0)" />
  <animate attributeName="width" attributeType="XML" begin="0s" dur="9s" fill="freeze"
    from="0" to="600" />
  <animate attributeName="height" attributeType="XML" begin="0s" dur="9s" fill="freeze"
    from="0" to="220" />
</rect>
</svg>
```

39

Animacje

- Element `<set>` pozwala zmieniać parametry stylu CSS z `visibility: hidden` na `visibility: visible`
- Atrybut `to` określa nową wartość zmienianego atrybutu

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <text style="font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12pt;
    font-style: normal;" x="40" y="100">
    Tekst w kolorze czerwonym
  </text>
  <set attributeName="fill" attributeType="CSS" to="#ff0000" />
</svg>
```

40

Animacje – ruch po torze

- Element `<animateMotion>` pozwala na przesunięcie początku obiektu po zadanej ścieżce
- Atrybuty
 - `path` - określa ścieżkę, po której odbywa się ruch
 - `M` - określa pierwszy punkt ścieżki
 - `L` - określa następne punkty ścieżki

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <text x="0" y="0" font-family="Verdana" font-size="45">
    Ruchomy tekst
  </text>
  <animateMotion path="M 40 L 40 100 L 40 10 40 0" dur="5s" repeatCount="indefinite"/>
</svg>
```

41

Animacja – transformacje

- Element `<animateTransform>` służy do nadawania obiektom takich transformacji jak `translate`, `scale`, `rotate`, `skewX`, `skewY`
- Atrybuty
 - `type` - określa rodzaj przekształcenia
 - `from` - określa początkową wartość parametru transformacji
 - `to` - określa końcową wartość parametru transformacji

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="500" height="500" viewBox="250 250 500 500">
  <text style="font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12pt;
    font-style: normal;" x="0" y="0">
    Tekst obracający się...
  </text>
  <animateTransform attributeName="transform" attributeType="XML" dur="5s"
    repeatCount="indefinite" type="rotate" from="0" to="360" fill="freeze" />
</svg>
```

42

Linki

- Element `<a>`
- Atrybuty
 - `href` - podobne do tych z HTML

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400" height="200" xmlns:xlink="http://www.w3.org/1999/xlink">
  <a xlink:href="http://www.w3.org/2000/svg">
    <ellipse cx="80" cy="80" rx="80" ry="80" fill="#3a678c" />
  </a>
  <a xlink:href="#cat.svg">
    <text x="200" y="30" style="font-size: 12pt;"> KOT </text>
  </a>
</svg>
```

43

Grafika rastrowa SVG

- Element `<image>` wstawia grafikę rastrową

Atrybuty

- `id` – określa identyfikator
- `xlink:href` – określa lokalizację i nazwę pliku graficznego
- `x, y` – określają współrzędne stawienia lewego górnego rogu grafiki rastrowej w dokumencie SVG
- `width, height` – określa szerokość i wysokość wstawianego obrazu rastrowego

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400" height="200" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.w3.org/2000/svg">
  <image id="plik" xlink:href="dinozaur.gif" x="10" y="10" width="100" height="100" />
</svg>
```

44

Interakcja z użytkownikiem

Nazwa zdarzenia	Opis	Nazwa atrybutu	Nazwa zdarzenia	Opis	Nazwa atrybutu
activate	Zdarzenie aktywowane w momencie nacisnięcia klawisza	onactivate	click	Zdarzenie aktywowane w momencie kliknięcia myszy w element	onclick
mouseover	Zdarzenie aktywowane w momencie, gdy kursor przesuwa się nad obiektem	onmouseover	mouseout	Zdarzenie aktywowane w momencie, gdy kursor przesuwa się z obiektu	onmouseout
repeat	Zdarzenie aktywowane w momencie, gdy zakończy się animacja obiektu	onend	repeat	Zdarzenie aktywowane w momencie, gdy powtórzy się animacja obiektu	onrepeat
scroll	Zdarzenie aktywowane w momencie, gdy nastąpi zmiana pozycji dokumentu	onscroll	scroll	Zdarzenie aktywowane w momencie, gdy zmieni się widoczność dokumentu	onscroll
scroll	Zdarzenie aktywowane w momencie, gdy nastąpi zmiana pozycji dokumentu	onscroll	scroll	Zdarzenie aktywowane w momencie, gdy zmieni się widoczność dokumentu	onscroll

45

Interakcja z użytkownikiem – przykład

SVG\zdarzenia\zdarzenieOnClick.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="600" height="500" xmlns="http://www.w3.org/2000/svg">
  <script type="text/ecmascript">
    <![CDATA[
      function circle_click(evt) {
        var circle = evt.target;
        var currentRadius = circle.getAttribute("r");
        if (currentRadius == 100)
          circle.setAttribute("r", currentRadius + 2);
        else
          circle.setAttribute("r", currentRadius * 0.5);
      }
    ]]>
  </script>
  <rect x="150" y="150" width="50" height="50" fill="none" stroke="blue" />
  <circle onclick="circle_click(evt)" cx="300" cy="225" r="100" fill="red" />
  <text x="300" y="480" font-family="Verdana" font-size="35" text-anchor="middle">
    Click on circle to change its size
  </text>
</svg>
```

46

Interakcja z użytkownikiem – przykład

SVG\zdarzenia\zmianaAtrybutow.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="600" height="500" xmlns="http://www.w3.org/2000/svg">
  <script type="text/ecmascript">
    <![CDATA[
      function oj(evt) {
        var s1 = svgDocument.getElementById("prost");
        s1.setAttribute("style", "stroke:rgb(255,0,0); stroke-width:7px;");
        s1.setAttribute("fill", "yellow");
      }
    ]]>
  </script>
  <rect onclick="oj()" id="prost" x="102" y="33" width="156" height="77" fill="rgb(0,0,255)"
    style="stroke:rgb(0,0,0); stroke-width:1" />
</svg>
```

47

Interakcja z użytkownikiem – przykład

SVG\zdarzenia\zdarzenieAnimacja.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="500" height="500" xmlns="http://www.w3.org/2000/svg">
  <rect id="start" x="150" y="180" width="180" height="40"
    style="fill:rgb(0,0,255);stroke:rgb(0,0,0); stroke-width:1" cursor="pointer"/>
  <rect id="stop" x="225" y="180" width="180" height="40"
    style="fill:rgb(255,0,0);stroke:rgb(0,0,0); stroke-width:1" cursor="pointer"/>
  <rect x="23" y="101" width="157" height="29" style="fill:rgb(0,255,0);stroke:rgb(0,0,0);
    stroke-width:1" />
  <animateMotion begin="start_click" end="stop_click" dur="4s" repeatCount="indefinite"
    path="M 40 10 L 300 350 L 40 350 L 40 10" />
</rect>
</svg>
```

48

Przykłady

SVG\przyklady\balwan.svg	SVG\przyklady\balwan_interakcja.svg	SVG\przyklady\choinka skrypt.svg
SVG\przyklady\choinka.svg	SVG\przyklady\choinka_tlf.svg	