

Technologie van multimediasystemen

Practicum 3: Bespreek Batik

Suzanna Cha Mieke Haesen

27 maart 2003

1 Wat is Batik?

Batik is een combinatie van tools die als invoer beelden in Scalable Vector Graphics formaat gebruiken. Scalable Vector Graphics, afgekort SVG, is een formaat dat twee-dimensionale afbeeldingen in XML beschrijft. Deze afbeeldingen kunnen ook dynamisch en interactief zijn. Op SVG wordt verder ingegaan in volgende sectie. Het belangrijkste doel van Batik is dus een SVG-document als afbeelding weergeven, de afbeelding genereren en manipuleren. Welke tools voorziet Batik nu precies? Hieronder volgt een opsomming.

- **Squiggle SVG Browser:** Dit is de belangrijkste en overkoepelende tool van Batik. Net zoals een Internet browser, zijn hier functies voorzien zoals „Stop processing”, „Reload”, „Forward”, „Back” en „View Source”.

Squiggle dient dus om SVG afbeeldingen weer te geven. Maar de browser voorziet ook extra mogelijkheden. Er is een „Find” functie om tekst te zoeken in de afbeelding. De gebruiker kan inzoomen en uitzoomen én hierbij is het mogelijk een „Thumbnail” weer te geven die bij een ingezoomd beeld de positie van het browserscherm aanduidt ten opzichte van de volledige afbeelding.

Enkele eenvoudige transformaties zoals translatie, rotatie en schaleren zijn in de browser uit te voeren. Deze transformaties worden echter niet opgenomen in het SVG document. Het is wel mogelijk de getransleerde afbeelding te converteren naar een raster formaat. Deze conversie is mogelijk door een andere Batik tool die in de browser is opgenomen, en zodadelijk aan bod komt. De gebruiker kan eveneens

de Document Object Model tree van een afbeelding bekijken.

- **SVG Rasterizer:** Deze tool biedt de mogelijkheid om SVG bestanden te converteren naar raster formaten zoals JPEG, PNG of Tiff. De gebruiker kan zelf ook formaten toevoegen.
- **SVG Font Converter:** Hiermee kan de gebruiker een SVG Font creëren voor een verzameling karakters van een True Type Font bestand. Dit garandeert dat de SVG documenten ongeacht het systeem, overal hetzelfde worden afgebeeld.
- **SVG Pretty Printer:** Zoals de naam reeds doet vermoeden, is de pretty printer een tool die de broncode van een SVG document gestructureerd weergeeft.

Batik is een onderdeel van het Apache XML project. Apache wil met Batik meteen een hele toolkit voor SVG ter beschikking stellen. Niet enkel als losstaande programma's, het is ook mogelijk dat programmeurs de tools kunnen toevoegen in hun Java applicaties en applets. Indien de gebruiker een SVG tag nodig heeft die Batik niet ondersteunt, beschikt men over de mogelijkheid deze tag zelf te definiëren. Er wordt gewerkt aan een Batik browser waarbij het ook mogelijk is om animaties weer te geven, maar deze versie is nog niet beschikbaar.

2 W3C

W3C (World Wide Web Consortium) is een internationaal consortium dat standaard webtechnologieën ontwikkelt en promoot, waaronder de specificaties voor Scalable Vector Graphics (SVG) 1.0. W3C bestaat sinds oktober 1994 en heeft ondertussen al meer dan vijftig technische specificaties voor de Web-infrastructuur ontwikkeld. Er is echter nog veel werk te doen, vooral wanneer computers, telecommunicatie en multimedia technologieën zich gaan verenigen. Hiervoor werkt W3C al aan de basis van de nieuwe Web-generatie. Het doel op lange termijn omvat drie punten :

- **Universal Access :** toegankelijkheid van het Web voor iedereen, door technologieën te promoten die rekening houden met verschillen in cul-

tuur, taal, opvoeding, bekwaamheid, materiële middelen, enz.

- **Semantic Web** : ontwikkelen van een software omgeving die het voor elke gebruiker mogelijk maakt zo optimaal mogelijk gebruik te maken van de beschikbare middelen op het Web.
- **Web of Trust** : de ontwikkeling van het Web leiden met zorgvuldige overwegingen omtrent nieuw uitgevonden legale, commerciële en sociale resultaten, voorgebracht door de technologie.

Hierdoor tracht W3C het Web te laten uitgroeien tot zijn maximale mogelijkheden.

De specificatie van SVG 1.0 beschrijft de syntax ervan en is te vinden op <http://www.w3.org/TR/SVG/>. Er bestaan ook specificaties van andere SVG versies (1.1, Mobile Profiles), maar vermits Batik werkt volgens 1.0 zullen we hier niet verder op ingaan. SVG is, zoals reeds gezegd, een taal om twee-dimensionale afbeeldingen te beschrijven in XML. Er zijn drie soorten grafische objecten; vector graphic shapes, images en text.

3 Code voorbeelden

In deze sectie worden een aantal zelf gemaakte code voorbeelden voorgesteld. Deze voorbeelden zijn niet zomaar mogelijke SVG documenten. Eerst zal er dieper worden ingegaan op code voorbeelden die in Batik wel mogelijk zijn, maar niet in sommige andere SVG browsers. Vervolgens worden code voorbeelden gegeven die juist zijn volgens de W3C specificatie, maar niet in Batik verwerkt kunnen worden.

Voor de vergelijking maken we gebruik van Batik 1.5 beta 1. Deze versie ondersteunt nog geen animaties. Er wordt een nieuwe versie voorzien van Batik, versie 2.0, die wel animaties zal kunnen genereren. Deze versie is echter nog niet beschikbaar.

3.1 Extra mogelijkheden t.o.v. andere SVG browsers

Om een vergelijking te kunnen maken, wilden we aanvankelijk de prestaties van Batik vergelijken met de volgende SVG browsers:

- Adobe SVG viewer 3.0 (<http://www.adobe.com/svg/>)

- Amaya (<http://www.w3.org/Amaya/Amaya.html>)
- CSIRO SVG Toolkit (<http://sis.cmis.csiro.au/svg/>)
- Mozilla SVG project (<http://www.mozilla.org/projects/svg/>)
- X-Smiles 0.71 (<http://www.xsmiles.org>)

Blijkbaar maakt X-Smiles gebruik van de CSIRO SVG Toolkit om SVG documenten weer te geven. En Adobe brengt het er vrij goed vanaf. Toch voor de eenvoudige voorbeelden die hier aan bod komen. Er blijven dus nog drie browsers over voor de vergelijkende studie: Amaya, CSIRO en Mozilla. W3C heeft een „Test Suite” voorzien. Dit is een verzameling van SVG documenten. Aan de hand van deze Test Suite, kan W3C de performantie van de verschillende browsers vergelijken. Deze SVG documenten bevatten echter vrij omslachtige code. En vermits ons gevraagd was zelf code voorbeelden te maken, hebben we ons beperkt tot de basis van SVG. We zullen ons bij de voorbeelden beperken tot de relevante code.

3.1.1 Voorbeeld 1



Figuur 1: Weergave van voorbeeld 1

De code van voorbeeld 1 beschrijft een afbeelding zoals in figuur 1. Dit voorbeeld wordt enkel in Batik juist weergegeven. In CSIRO wordt de achtergrond juist weergegeven, maar zijn de woorden „horizontal” en „vertical” niet van boven naar onder geschreven. Dit betekent dus dat het attribuut „writing-mode” in deze browser niet ondersteund wordt, evenals de attributen „glyph-orientation-vertical” en „glyph-orientation-horizontal” die aangeven of die tekst verticaal of horizontaal moet worden geschreven. De code voor het juist uitschrijven van het woord „vertical” ziet er als volgt uit.

```

<!-- Tekst van boven naar onder: verticaal -->
<g glyph-orientation-vertical="0" writing-mode="tb">
  <g font-family="Helvetica" font-size="30"
    transform="translate(410,30)">
    <text x="0">Vertical</text>
  </g>
</g>

```

De andere browsers kampen met hetzelfde probleem. Bovendien wordt daarin de achtergrond niet getoond. Dit wil dus zeggen dat het element „linearGradient”, dat ervoor zorgt dat de kleuren interpoleren, niet herkend wordt in Amaya en Mozilla. Hoe „linearGradient” gebruikt wordt, volgt in dit code fragment:

```

<g>
  <defs>
    <!-- Kleurenschakering voor de achtergrond -->
    <linearGradient id="linGrad">
      <stop offset="0" stop-color="yellow"/>
      <stop offset="0.5" stop-color="red"/>
      <stop offset="1" stop-color="blue"/>
    </linearGradient>
  </defs>
  <!-- Rechthoek die de achtergrond bepaalt -->
  <g color="inherit" fill="none" stroke="none">
    <rect fill="url(#linGrad)" height="360"
      width="480" x="0" y="0"/>
  </g>
</g>

```

3.1.2 Voorbeeld 2

Een andere extra mogelijkheid van Batik ten opzichte van andere browsers is het gebruik van het „filter”-element. Sommige browsers ondersteunen het niet, anderen slechts gedeeltelijk. Filteren is een bewerking uitvoeren op een bestaande afbeelding. Dit kan een externe afbeelding zijn, maar het kan ook een figuur zijn, of een samenstelling van figuren. Filteren biedt verschillende mogelijkheden. Een veel voorkomende toepassing is het meermaals gebruiken van de afbeelding. Zo kan men bijvoorbeeld een zwart en waziger kopie van een afbeelding maken en deze kopie een beetje verschuiven zodat de oorspronkelijke afbeelding een schaduw heeft gekregen. Dit is een eenvoudig

voorbeeld van wat er mogelijk is met een filter. We passen dit voorbeeld toe op een woordje „Test”. Hieronder volgt een fragment van de code:

```
<defs>
  <filter height="120"
    id="TekstFilter" width="200" x="0" y="0">
    <!-- De schaduw zelf -->
    <feGaussianBlur in="SourceAlpha" result="waziger"
      stdDeviation="4"/>
    <!-- De schaduw verschuiven -->
    <feOffset dx="4" dy="4" in="waziger" result="schaduw"/>
    <!-- Het object waar we schaduw voor maken -->
    <feComposite in="SourceGraphic" in2="SourceAlpha"
      result="tekst"/>
    <!-- Weergeven van de schaduw
      en de tekst in de juiste volgorde -->
    <feMerge>
      <feMergeNode in="schaduw"/>
      <feMergeNode in="tekst"/>
    </feMerge>
  </filter>
</defs>

<!-- De tekst die moet worden weergegeven -->
<text filter="url(#TekstFilter)"
  font-family="Times New Roman" font-size="50"
  font-style="italic" x="50" y="70">Test</text>
```

Als we in Batik de code converteren naar een .png formaat, krijgen we hetzelfde resultaat als in figuur 2.

Aan de hand van dit voorbeeld kan men zien dat zelfs deze eenvoudige filter in de andere browsers niet wordt toegepast. In dit voorbeeld is het het element „Filter” dat niet wordt toegepast door CSIRO, Amaya en Mozilla. De functies die hier het filteren zelf verwezenlijken zijn vervat in volgende elementen: „feGaussianBlur”, „feOffset”, „feComposite” en „feMerge”. Vermits er eindeloos veel toepassingen zijn met filters, beperken we ons hier tot dit eenvoudige voorbeeld. Met behulp van de W3c Test Suite kan men zien dat Batik alle filters ondersteunt, in tegenstelling tot de andere browsers.

Test

Figuur 2: Weergave van voorbeeld 2

3.2 Mogelijkheden volgens W3C, maar niet ondersteund door Batik

Op het gebied van statische objecten voldoet Batik aan de W3C specificaties (getest met behulp van de W3C Test Suite). Maar op het gebied van animaties zijn nog niet alle 'onvolmaaktheden' opgelost. De nieuwe versie (2.0) van Batik die in de maak is zou wel alle vormen van dynamische objecten ondersteunen, maar deze is, zoals reeds gezegd, nog niet beschikbaar. In volgende secties zullen we enkele voorbeelden geven van dynamische objecten die Batik niet op de juiste manier weergeeft. We zullen ook vergelijken met andere SVG browsers (Adobe en CSIRO).

3.2.1 Voorbeeld 3

Voorbeeld 3 is een rechthoek die tevoorschijn komt, weer verdwijnt, en dit (oneindig) blijft herhalen. Geleidelijk via grijs gaat de figuur over naar het eindresultaat, een zwarte rechthoek. Zowel Adobe als CSIRO geven dit voorbeeld perfect weer. Batik daarentegen toont een zwarte rechthoek die niet geanimeerd wordt. De animate-tag wordt dus (nog) niet ondersteund.

Code fragment :

```
<rect width="60" height="40" transform="translate(15,30)">  
  <animate attributeType="CSS" attributeName="opacity"  
    from="0" to="1" begin="0s"  
    dur="5s" repeatCount="indefinite" />  
</rect>
```

3.2.2 Voorbeeld 4

In dit voorbeeld laten we zien dat Batik geen pointer-events ondersteunt. Het voorbeeld bevat een gele rechthoek en een open cirkel, die met groen of rood

gevuld wordt als de cursor zich binnen resp. buiten de rechthoek bevindt. Adobe ondersteunt dit wel, en CSIRO gedeeltelijk. Bekijken we voorbeeld 4 in de CSIRO SVG viewer, dan treedt er een fout op bij het verlaten van de rechthoek; het cirkeltje kleurt enkel rood als de cursor zich onder of rechts van de rechthoek bevindt.

Code fragment :

```
<defs>
  <!-- scripting met gebruik van ecma-script (is overigens
  de default script-taal voor SVG) -->
  <script type="text/ecma-script"><![CDATA[
  function Cursor(evt, idname, color)
  {
    // verwijzen naar object
    var doc = evt.target.ownerDocument;
      //getElementById(idname) : geeft het 1e object
      //terug met hetzelfde ID attribuut als de
      //gespecificeerde waarde (idname='circle').
    var circle = doc.getElementById(idname);
      //setAttribute kan de waarde van een attribuut veranderen,
      //hier krijgt het attribuut 'fill' (van circle) de nieuwe
      //waarde 'color'
    circle.setAttribute('fill', color);
  }
  ]]></script>
</defs>
<g font-family="Helvetica" font-size="12" text-anchor="start">
  <!-- pointer-events -->
  <g transform="translate(30,50)">
    <!--gele rechthoek waarover men met de muis kan
    bewegen om het resultaat te zien-->
    <!-- onmouseout en onmouseover zorgen voor de
    interactie met de cirkel : groen gevuld als de
    cursor over de rechthoek gaat, rood gevuld als de
    cursor zich weer buiten de rechthoek verplaatst-->
    <rect fill="yellow" height="30"
    onmouseout="Cursor(evt, 'circle', 'red')"
    onmouseover="Cursor(evt, 'circle', 'green')"
    stroke="none" width="50"/>
    <text fill="black" stroke="none" x="0" y="45">
      Place your cursor inside the rectangle and the little
```

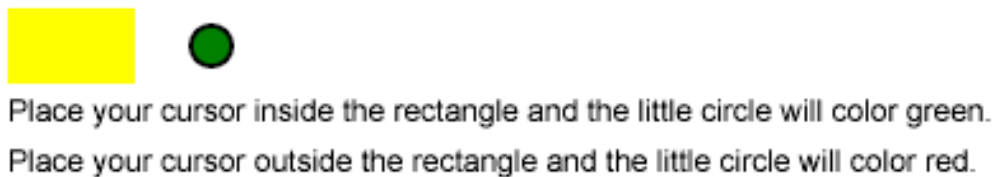


```

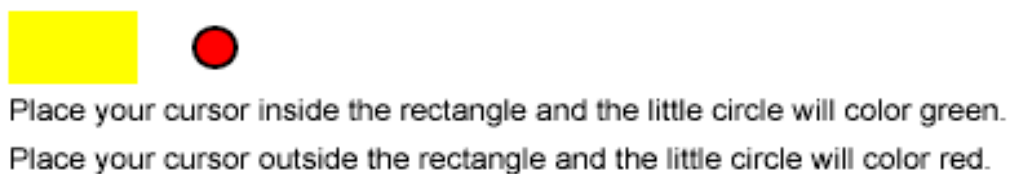
        circle will color green.
    </text>
    <text fill="black" stroke="none" x="0" y="65">
        Place your cursor outside the rectangle and the little
        circle will color red.
    </text>
    <g stroke="black" stroke-width="2">
        <circle cx="80" cy="15" fill="none" id="circle" r="8"/>
    </g>
</g>

```

De hierop volgende Figuren 3 en 4 geven het correcte resultaat weer (door Adobe SVG viewer) van voorbeeld 4 als de cursor zich respectievelijk binnen en buiten de rechthoek bevindt. Batik laat een statisch beeld zien waarbij er niet wordt gereageerd op de plaats van de cursor t.o.v. de rechthoek (zie Figuur 5).



Figuur 3: Weergave van voorbeeld 4 (Adobe)



Figuur 4: Weergave van voorbeeld 4 (Adobe)



Place your cursor inside the rectangle and the little circle will color green.

Place your cursor outside the rectangle and the little circle will color red.

Figuur 5: Weergave van voorbeeld 4 (Batik)

4 Software Architectuur van Batik

Om toepassingen op Batik te kunnen maken, dient men een beetje vertrouwd te zijn met de software architectuur en de source code van Batik. Batik is geïmplementeerd in Java. In feite kan de software architectuur van Batik worden onderverdeeld in drie lagen.

1. Low level modules
2. Kern modules
3. Applicatie modules

Deze niveau's zullen we nu nader bestuderen.

4.1 Low level modules

Op het laagste niveau bevinden zich de Low level modules. Deze modules worden niet rechtstreeks gebruikt door software ontwikkelaars, maar ze vormen een basis voor de volgende laag. Onderdelen die vervat zijn in deze laag zijn de Graphic Vector Toolkit (GVT), de Renderer en de SVG Parser. De GVT is een module die een overzicht van de DOM boom representeert. De Renderer maakt het mogelijk dat de GVT boom wordt weergegeven en deze verzorgt ook andere taken. De SVG Parser tenslotte, bevat „Micro Parsers”. Dat zijn parsers voor complexe SVG attributen.

4.2 Kern modules

De Batik kern modules zijn afhankelijk van de low level modules en ze vormen het middelpunt van de Batik architectuur. Men kan ze beschouwen als de laag die de low level modules linkt met de applicatie modules. De kern modules kan men dus gebruiken om toepassingen met Batik te maken. Deze laag in de architectuur bevat enkele gebruiksklare modules: de SVG Generator, de

SVG DOM, het JSVGCanvas en de Transcoder. Als men in een applicatie een afbeelding wil omzetten naar SVG formaat, kan men de SVG Generator gebruiken. De SVG DOM laat de programmeur toe SVG documenten te manipuleren in een Java programma. Voor het weergeven van een SVG document en om er transformaties op toe te passen, beschikt men over het JSVGCanvas. De transcoder is een module die een document omzet naar een bepaalde uitvoer.

In deze groep van modules zit ook een module die niet rechtstreeks gebruikt kan worden, de Bridge. Deze module is echter niet minder belangrijk. Hierin wordt ervoor gezorgd dat een SVG document wordt geconverteerd naar de interne representatie die Batik gebruikt voor afbeeldingen, meerbepaald de GVT. Voor de toepassingen maakt men gebruik van deze modules. In de volgende sectie wordt hier verder op in gegaan bij het maken van een eigen toepassing.

4.3 Applicatie modules

Zoals reeds eerder vermeld, bevinden zich op deze laag de applicaties. In Batik zelf zijn dit de SVG Browser, de SVG Pretty Printer, de SVG Font Converter en de SVG Rasterizer, die we in sectie 1 reeds hebben besproken. Hoewel deze applicaties zeer nuttig kunnen zijn, gaat het hier niet over de belangrijkste doelstelling van Batik. Deze applicaties zijn enkel voorbeelden van de mogelijkheden met Batik. Dus in feite behoren de toepassingen op Batik ook tot dit niveau. En dus ook onze eigen toepassing zal bij deze laag in de Batik architectuur gerekend mogen worden.

5 Toepassingen geïmplementeerd met Batik

Batik stelt zijn source code ter beschikking voor het ontwikkelen van Java applicaties en applets. En daar wordt gretig gebruik van gemaakt. In deze sectie zullen we wat dieper ingaan op enkele toepassingen geïmplementeerd met Batik. Eerst bekijken we de mogelijkheden en bestaande toepassingen. Vervolgens implementeren we zelf een kleine voorbeeldtoepassing met behulp van Batik.

5.1 Mogelijke en bestaande toepassingen

Vermits Batik verschillende tools aanbiedt, zijn er ook veel mogelijkheden wat toepassingen betreft. De SVG Rasterizer wordt vaak gebruikt om SVG

documenten om te zetten naar bitmaps. Een voorbeeld van zo'n toepassing is Apache Cocoon (<http://xml.apache.org/cocoon/>). Cocoon is een framework dat de gebruiker de mogelijkheid wil bieden XML en XSLT technologieën te gebruiken voor Server applicaties. Een ander voorbeeld is XWeb (<http://xweb.sourceforge.net/>). Deze tool maakt het mogelijk volledige websites te genereren met behulp van XML. Een belangrijk onderdeel hiervan is het weergeven van knoppen of banners waarvan de code in SVG is geschreven.

Batik wordt ook opgenomen in toepassingen voor de weergave van SVG documenten. Bitflash Brilliance (<http://www.bitflash.com/products/brilliance.asp>), een editor voor het ontwikkelen van SVG documenten, doet dit niet enkel met behulp van de code, maar ook met behulp van designer tools. Op die manier is het mogelijk een afbeelding te maken waarbij het SVG document automatisch aangepast wordt.

Een andere interessante toepassing van Batik is JFreeChart (<http://www.object-refinery.com/jfreechart/index.html>). Met dit programma kan men grafieken maken in SVG en andere formaten. Het enige wat men moet doen, is de statistische gegevens invoeren, en het type grafiek kiezen.

Als we nu zelf toepassingen proberen te bedenken, en daarbij teruggaan naar voorbeeld 2, dan is een eenvoudig voorbeeld een programma dat een willekeurige figuur of tekst in SVG als invoer neemt, de schaduw van deze figuur genereert, en met behulp van de SVG rasterizer, een raster formaat genereert. Een ander voorbeeld is een SVG editor die de mogelijkheid biedt om een pretty print uit te voeren op het document. Dit gaan we proberen te implementeren. Voor meer informatie hierover verwijzen we naar de volgende sectie.

Zoals reeds eerder vermeld, is X-smiles een browser die CSIRO gebruikt voor de weergave van SVG documenten. Dat is ook mogelijk met behulp van Batik. Als bijvoorbeeld een internet browser de Batik browser zou gebruiken, kan men SVG documenten in een gewone browser bekijken. Adobe werkt ook op die manier, SVG documenten kunnen met behulp van Adobe in Internet Explorer worden bekeken.

5.2 Een zelf gemaakte mini-toepassing

Zoals reeds eerder vermeld, hebben we getracht een SVG editor te implementeren waarin het mogelijk is een SVG document gestructureerd weer te geven. In de SVGBrowser van Batik is het mogelijk de source code te bekijken. Hier is niet mogelijk de code aan te passen en op te slaan, vandaar dat een SVG editor ons nuttig lijkt.

5.2.1 De code

De editor is ontwikkeld met behulp van Borland JBuilder. In het volgende codefragment wordt de code gegeven voor het uitvoeren van de pretty print. Eerst een kort overzichtje van de imports, en vervolgens de code die wordt uitgevoerd als de gebruiker een pretty print aanvraagt.

De geïmporteerde Batik packages:

```
import org.apache.batik.i18n.LocalizableSupport;
import org.apache.batik.transcoder.Transcoder;
import org.apache.batik.transcoder.TranscoderInput;
import org.apache.batik.transcoder.TranscoderOutput;
import org.apache.batik.transcoder.TranscodingHints;
import org.apache.batik.transcoder.svg2svg.SVGTranscoder;
```

De code om een pretty print uit te voeren:

```
//Pretty print
void jMenuItemPrettyPrint_actionPerformed(ActionEvent e) {
    try{
        //Tekst uit editor halen
        String tekst = jEditorPane.getText();

        TranscoderInput in;
        in = new TranscoderInput(tekst);
        TranscoderOutput out;
        out = new TranscoderOutput(tekst);

        //Pretty print zelf uitvoeren
        transcoder.transcode(in, out);

        //Aangepaste tekst opnieuw in editor weergeven
        jEditorPane.setText(tekst);
    } catch (Exception exc) {
        jEditorPane.setText("Pretty print mislukt");
    }
}
```

De pretty print wordt uitgevoerd door het aanroepen van de functie „transcode(in, out)”. Deze functie is voorzien in de klasse org.apache.batik.transcoder.svg2svg.SVGTranscoder. Hierin wordt de functie prettyPrint() opgeroepen die om de tekst te structureren gebruik maakt

van de klasse `org.apache.batik.transcoder.svg2svg.PrettyPrinter`. Deze klassen bevinden zich op het middelste niveau in de software architectuur, namelijk de kern modules.

5.2.2 Conclusie

Het is zeer handig voor programmeurs om gebruik te kunnen maken van reeds bestaande packages. In ons geval echter, loopt er iets mis bij het uitvoeren van de `transcode`. In de editor hebben we een `String` staan die de SVG code bevat, en het is niet mogelijk een `String` mee te geven aan deze functie. Er wordt een „`InputStream`” verwacht die ofwel een „`Reader`” ofwel een „`Document`” bevat. Wegens tijdsgebrek hebben we dit niet meer kunnen aanpassen.

Er valt ook op te merken dat de `SVGBrowser`-applicatie die Batik voorziet, de `pretty print` eveneens onjuist uitvoert. Hierin wordt dezelfde functie „`transcode()`” opgeroepen en bij de uitvoering ervan gebeurt er niets. De `pretty print` applicatie van Batik (`SVGpp`) gebruikt ook „`transcode()`”, en deze werkt wel juist.

6 Meer info

Op volgende url zijn zowel de voorbeelden uit sectie 3 als de toepassing op Batik uit sectie 5.2, de SVG editor, te vinden.

<http://meeksieke.tripod.com/Practicum3/index.htm>

Deze laatste kan gedownload worden als `JBuilder project(.jpx)` in een zip file. De Batik packages zijn ook aan dit project toegevoegd. Er is tevens de mogelijkheid de klassen in `.java` formaat apart te downloaden, net als de nodige Batik packages.

Ook dit verslag en de presentatie zijn er te raadplegen.

De voorbeelden en de java klassen dienen met de juiste extensie te worden opgeslagen (`.svg` en `.java`). De SVG bestanden kunnen dan bekeken worden met behulp van een SVG browser of geopend worden in kladblok. De java klassen kunnen geopend worden met bv. Borland JBuilder.

Opmerking hierbij is dat om de een of andere reden de browser bepaalde tags ongewild verandert (bv. kleine letters worden hoofdletters) waardoor er fouten kunnen optreden in de SVG documenten. Indien dit het geval is, kan er naar ons gemaild worden voor de juiste versies.

7 Referenties

Volgende bronnen hebben we gebruikt voor onderzoek.

- Een introductie tot SVG:
<http://alpha.luc.ac.be/Didactiek/WNI/IK3/TMS/xmlcursus/index.html>
- De website van Batik:
<http://xml.apache.org/batik/>
- W3C, hier vindt men de volledige SVG specificatie evenals besprekingen van SVG viewers:
<http://www.w3.org/Graphics/SVG/Overview.htm8>
- Andere SVG browsers, gebruikt voor een vergelijkende studie:
 - Adobe SVG viewer 3.0:
<http://www.adobe.com/svg/>
 - Amaya:
<http://www.w3.org/Amaya/Amaya.html>
 - CSIRO SVG Toolkit:
<http://sis.cmis.csiro.au/svg/>
 - Mozilla SVG project:
<http://www.mozilla.org/projects/svg/>
 - X-Smiles 0.71:
<http://www.xsmiles.org>
- Toepassingen die gebruik maken van Batik:
 - Apache Cocoon: <http://xml.apache.org/cocoon/>
 - XWeb: <http://xweb.sourceforge.net/>
 - Bitflash Brilliance:
<http://www.bitflash.com/products/brilliance.asp>
 - JFreeChart:
<http://www.object-refinery.com/jfreechart/index.html>
- Java website als hulpmiddel bij implementatie van de Batik toepassing:
<http://java.sun.com/>