

Practicum 2

MPEG-21 Digital Item Declaration Language & W3C Webservices

1. Inleiding

MPEG-21 is een initiatief genomen door het MPEG-consortium (Moving Picture Experts Group) met als doel een raamwerk te definiëren waar bestaande en toekomstige multimedietoepassingen kunnen worden ingepast [1]. Het aantal multimediabestanden en het aantal manieren waarop deze bestanden gecodeerd zijn, is gedurende de laatste jaren sterk toegenomen. Daarenboven is het, dankzij het Internet, erg eenvoudig geworden om deze bestanden, ongeacht hun geografische locatie, te raadplegen. Verder is ook de verzameling van netwerken en terminals, gebruikt om de multimediabestanden te transporteren en af te leveren, sterk uitgebreid gedurende de laatste jaren. Er bestond echter nog geen oplossing die al deze verschillende technieken, groepen en gemeenschappen in staat stelt om hun codeerschema's, hun metadata (extra informatie over de eigenlijke data), hun modellen en procedures voor het verwerken van multimedidata, op een transparante en efficiënte manier uit te wisselen. Het is aan deze problemen dat het MPEG-comité poogt te verhelpen door middel van de MPEG-21-standaard.

De visie die aan de basis van de MPEG-21-standaard ligt, luidt als volgt: "MPEG-21 definieert een multimediaroomwerk dat het transparante gebruik van multimedidata over verschillende netwerken en terminals heen mogelijk maakt en bespoedigt". De MPEG-21-standaard probeert dan ook de volledige verwerkingsketen, startende bij de creatie van de multimedidata, over de productie en de aflevering ervan tot aan de consumptie door de eindgebruiker te modelleren. Het is duidelijk dat een dergelijk raamwerk, wil het enige kans op slagen hebben, de interoperabiliteit tussen verschillende systemen moet vereenvoudigen en, waar mogelijk, de transacties tussen deze verschillende systemen moet automatiseren. Dit dient onder andere mogelijk gemaakt te worden voor de aflevering van multimedidata, voor de beveiliging van de multimedidata en voor de beschrijving van de intellectuele eigendomsrechten daarmee verbonden.

In dit practicum zal ISO/IEC 21000-2 [2] wat meer in detail bestudeerd worden. ISO/IEC 21000-2, beter gekend als Digital Item Declaration, bevat een onderdeel waarin de Digital Item Declaration Language (DIDL) gedefinieerd wordt. Deze taal laat toe om op een interoperabele wijze de relatie uit te drukken tussen de verschillende elementen van een digitaal item, de eenheid van werken in het MPEG-21-raamwerk. In het kader van dit practicum zullen enkele bouwstenen uit DIDL aan bod komen en zal er eveneens onderzocht worden hoe een digitaal item opgesteld kan worden conform de MPEG-21-standaard. Verder zal er gebruikgemaakt worden van een webservice om een generieke Uniform Resource Identifier (URI) [3] om te zetten naar een meer concrete Uniform Resource Locator (URL) [4].

2. Digital Items

Zoals in de vorige paragraaf beschreven, probeert de MPEG-21-standaard een raamwerk aan te bieden dat het gebruik en de integratie van multimedidata bevordert. Om dit te kunnen realiseren moet een definitie ingevoerd worden die toepasbaar is op alle mogelijke multimedidata en hun onderlinge relaties. Binnen MPEG-21 wordt dit concept een *digitaal item* (Eng. Digital Item) genoemd. Een digitaal item wordt gedefinieerd als een gestructureerd, digitaal object met een standaard representatie, identificatie en

bijbehorende metadata. Een digitaal item wordt aanzien als de eenheid van transport en distributie binnen het MPEG-21-raamwerk. Verder is er geen enkele technische betekenis gekoppeld aan dit begrip. Een concreet voorbeeld van een digitaal item is een muziekalbum.

Het is duidelijk dat er heel veel verschillende types inhoud van multimediadata bestaan en op zijn minst evenveel manieren om de context waarbinnen die multimediadata kunnen gebruikt worden te beschrijven. De definitie van een digitaal item, alsook de definitie van een aantal aanverwante concepten, worden in detail besproken in het tweede deel van de MPEG-21-standaard. We vermelden tenslotte nog dat een *digitaal-itemdeclaratie* (DID) een document is dat de opmaak, structuur en organisatie van een digitaal item definieert.

3. Digital Item Declaration Language

Het tweede deel van MPEG-21 werd bedacht met de naam MPEG-21 Digital Item Declaration (ISO/IEC 21000-2) en zal in dit practicum in wat meer detail bestudeerd worden. Het doel van het tweede deel van de MPEG-21-standaard is het aanbieden van een model waarbinnen digitale items op een zo generiek mogelijke manier kunnen beschreven worden. Zoals eerder vermeld, stelt een digitaal item in deze context “een werk” voor, waarop bewerkingen kunnen worden uitgevoerd (consumeren, beschrijven, uitwisselen, enz.). In de plaats van een taal te definiëren, biedt dit model een verzameling termen en concepten aan die kunnen gebruikt worden bij het beschrijven van en het werken met digitale items. Het Digital Item Declaration-gedeelte van de standaard wordt opgedeeld in drie normatieve delen.

- Model: Het Digital Item Declaration-deel van de standaard beschrijft een verzameling van termen en concepten die samen een model vormen voor het definiëren van digitale items. Binnen dit model is een digitaal item een abstracte voorstelling van een werk.
- Voorstelling: Een normatieve beschrijving van de syntax en de semantiek voor elk van de elementen uit het Digital Item Declaration-gedeelte, zoals deze worden genoteerd in XML.
- Schema: Het normatieve XML-schema voor de volledige Digital Item Declaration-grammatica.

De laatste twee onderdelen definiëren in essentie de Digital Item Declaration Language. Het is duidelijk dat dit een taal is die aan de auteur van een digitaal item de mogelijkheid geeft de relatie tussen de verschillende elementen van een digitaal item te beschrijven met behulp van XML. In de volgende paragrafen wordt er een overzicht gegeven van een aantal DIDL-bouwstenen die in dit practicum zullen gebruikt worden. De volledige specificatie van de DIDL bevat naast deze elementen ook nog een verzameling van andere elementen die toelaten om digitale items te configureren en te annoteren. Een bespreking van de syntax en semantiek van deze elementen valt echter buiten het bestek van dit practicum. De geïnteresseerde lezer wordt hiervoor dan ook verwezen naar de DIDL-specificatie.

3.1. Resource

Een *Resource*-element laat toe om multimediale data op te nemen in een digitaal-itemdeclaratie. Meestal wordt dit gedaan aan de hand van een referentie. Multimediale data kunnen, in de context van *Resources*, heel ruim geïnterpreteerd worden. Het betreft hier zowel audio, video, tekst als stilstaande afbeeldingen.

Hieronder staat ter illustratie een XML-representatie van een `Resource`-element. Een dergelijk element bevat twee attributen, het `ref`-attribuut en het `mimeType`-attribuut. Het `ref`-attribuut bevat een URI die verwijst naar een locatie waar de eigenlijke data van de `Resource` terug te vinden is. Het `mimeType`-attribuut bevat het MIME-type [5] van de `Resource`. Dit MIME-type kan bijvoorbeeld door een MPEG-21-applicatie gebruikt worden om de juiste mediaspeler op te starten wanneer men de `Resource` wenst af te spelen. Zo kan een MPEG-21-applicatie er voor kiezen om een audiospeler op te starten indien de `Resource` geluid bevat. Indien de `Resource` een afbeelding bevat, kan er bijvoorbeeld een beeldbewerkingsprogramma opgestart worden.

```
<Resource mimeType="image/jpg" ref="http://server/leukefoto.jpg"/>
```

3.2. Descriptor/Statement

De combinatie van een `Descriptor`-element met een `Statement`-element laat toe om op een eenvoudige wijze een beschrijving toe te voegen aan andere elementen uit de DIDL. Een dergelijke beschrijving kan bijvoorbeeld technische informatie bevatten, zoals de bitsnelheid van een gecodeerde videostroom. Anderzijds kan een `Descriptor/Statement` ook beschrijvende informatie bevatten over de eigenlijke inhoud van het digitaal item. Hieronder staat een voorbeeld van beschrijvende informatie die aan een muziekalbum zou kunnen toegevoegd worden. Merk op dat beschrijvende informatie niet enkel beperkt is tot tekstuele data.

```
<Descriptor>
  <Statement mimeType="text/plain">De tijdloze 100</Statement>
</Descriptor>
```

3.3. Component

Een `Component`-element groepeert een verzameling van `Resource`- en `Descriptor`-elementen tot één logisch geheel. De `Descriptor`-elementen die in een `Component` opgenomen worden, dienen vooral om technische informatie te geven over de inhoud (bitsnelheid, resolutie, enz.) en dienen niet voor beschrijvende informatie. Meerdere `Resource`-elementen in een `Component` opnemen laat bijvoorbeeld toe om verschillende versies van een bepaalde `Resource` op te nemen in één `Component`. Op die manier kan er verwezen worden naar verschillende servers zodat een slimme MPEG-21-applicatie over de mogelijkheid beschikt om er die server uit te kiezen die het dichtst bij de terminal van de eindgebruiker gelegen is, of waarvoor het minst betaald moet worden.

```
<Component>
  <Resource mimeType="audio/mp3"
    ref="http://snelle.dure.server/music.mp3"/>
  <Resource mimeType="audio/mp3"
    ref="http://trage.goedkope.server/music.mp3"/>
</Component>
```

3.4. Item

Een `Item`-element groepeert `Descriptor`-elementen met `Component`-elementen. Een `Item` is een fundamenteel onderdeel van een MPEG-21 digitaal item. Een `Item` zonder sub-Items is een ondeelbaar geheel dat enerzijds is opgebouwd uit `Descriptor`-elementen die een beschrijving bevatten van het `Item`, en dat anderzijds is opgebouwd uit `Component`-elementen die de multimediale data van het betreffende `Item` bevatten. Een `Item` dat sub-Items bevat is opgebouwd uit verschillende onderdelen die elk op zich weer sub-Items kunnen bevatten. Zo is het bijvoorbeeld mogelijk om een muziekalbum uit te drukken als een `Item`. Dit `Item` kan dan een `Descriptor` bevatten die het muziekalbum beschrijft. Verder bevat dit `Item` bijvoorbeeld ook nog een aantal sub-Items die op hun beurt telkens de verschillende muzieksporen van het muziekalbum bevatten. Elk van die sub-Items kunnen dan een `Descriptor` bevatten met de beschrijving van het muziekspoor en een `Component` met de eigenlijke geluidsdata die in het muziekspoor opgenomen zijn. Hieronder wordt een mogelijk skelet voorgesteld van een digitaal item dat een muziekalbum beschrijft.

```
<Item>
  <Descriptor>
    <!-- ... -->
  </Descriptor>
  <Item>
    <Descriptor>
      <!-- ... -->
    </Descriptor>
    <Component>
      <!-- ... -->
    </Component>
  </Item>
  <Item>
    <!-- ... -->
  </Item>
</Item>
```

3.5. DIDL

Omdat elk XML-document slechts één wortelelement kan hebben, is er ook in de DIDL een wortelelement aanwezig. Het wortelelement van de DIDL dient steeds in de naamruimte `urn:mpeg:mpeg21:2002:02-DIDL-NS` te staan.

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS">
  <Item>
    <!-- ... -->
  </Item>
</DIDL>
```

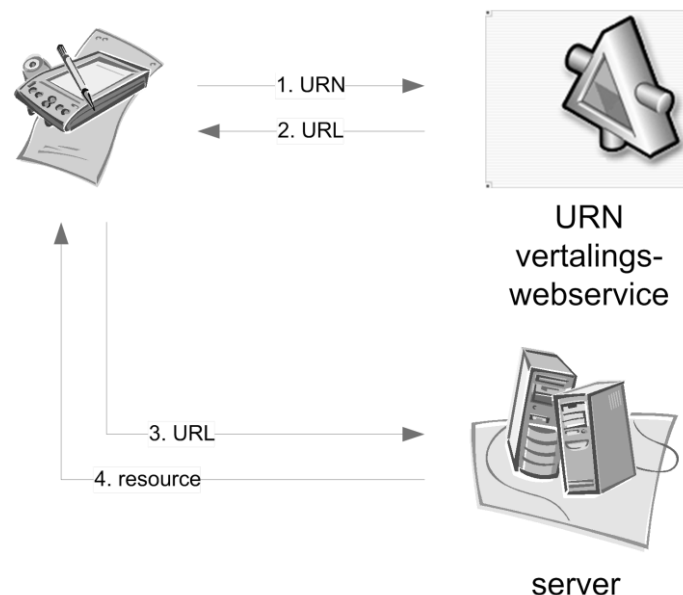
4. Uniform Resource Identifiers

In dit practicum wordt er gebruikgemaakt van Uniform Resource Identifiers om multimediate bronnen op een unieke manier te identificeren. Een URI kan gedefinieerd worden als een compacte representatie die verwijst naar een abstracte of fysieke bron (een elektronisch document, een boek, een persoon, enz.).

Er bestaan verschillende soorten URIs. Het meest gekende type URI is wellicht de Uniform Resource Locator of URL. Een URL is een compacte representatie van een locatie en toegangsmethode tot een bron beschikbaar via het Internet. Het type URI dat in dit practicum aan bod komt is de Uniform Resource Name [6]. Een URN laat toe om een bron te identificeren op een persistente en locatie-agnostische manier. In de praktijk betekent dit dat een instantie (registratie-autoriteit, consortium, enz.) garandeert dat de URN niet zal wijzigen en steeds beschikbaar zal zijn. Een voorbeeld van een URN is het ISBN-nummer van een boek.

Omwille van het feit dat URNs locatie-onafhankelijk zijn, is het nodig om te beschikken over een mechanisme dat toelaat een gegeven URN te vertalen in een locatiespecifieke identificatie. Figuur 1 illustreert een mogelijke vertaling van een URN naar een URL. De volgende 4 stappen worden beschreven:

1. een URN wordt aan een vertalingswebservice gegeven;
2. de vertalingswebservice geeft een URL als resultaat terug;
3. via de URL wordt een server aangesproken;
4. de gewenste data worden teruggegeven door de server.



Figuur 1: Vertaling van een URN naar een URL.

Deze extra indirectie heeft als voordeel dat de werklust over verschillende servers kan gespreid worden (Eng. load balancing). In dit practicum wordt er gebruikgemaakt van een webservice waarbij de werklust tussen verschillende videoaanbieders verdeeld wordt. Daartoe wordt een generieke URN vertaald naar een meer specifieke URL.

5. XML in .NET

Reeds van bij het prille begin hebben de ontwerpers van het .NET-raamwerk rekening gehouden met het gebruik van XML. Daarom is het .NET-raamwerk uitermate geschikt voor de verwerking van XML-documenten. Hieronder volgt een beknopte beschrijving van een aantal klassen die nuttig zijn wanneer XML-documenten met het .NET-raamwerk dienen verwerkt te worden.

Met de klasse `XmlDocument` is het mogelijk om XML-documenten in te lezen, te bevragen, te bewerken en nadien terug op te slaan. Het laden van een XML-bestand

gebeurt met behulp van de `Load()`-methode. Aan deze methode wordt een URL als argument meegegeven.

Om een `XmlDocument` te kunnen bevragen met behulp van `XPath` is het nodig om een `XmlNamespaceManager` aan te maken. Deze `XmlNamespaceManager` is verantwoordelijk voor de binding tussen de prefixen, die gebruikt worden in `XPath`-expressies, en de naamruimten. Dit kan gedaan worden met behulp van de `AddNamespace()`-methode.

Het opvragen van de waarde van een bepaalde knoop kan gerealiseerd worden door de `SelectSingleNode()`-methode op te roepen. Deze methode heeft twee argumenten: het eerste argument is een `XPath`-expressie waarmee een bepaalde knoop geselecteerd wordt. Het tweede argument is een `XmlNamespaceManager`-object dat verantwoordelijk is voor de koppeling tussen de prefixen die gebruikt worden in de `XPath`-expressie en de naamruimten.

Onderaan staat een uitgewerkt voorbeeld dat het gebruik van de hierboven besproken klassen en methoden illustreert.

```
using System.Xml;

// Aanmaken van een XmlDocument.
XmlDocument doc = new XmlDocument();

// Laden van het XML-document.
doc.Load("http://mijnserver/mijnxml.xml");

// Aanmaken van een XmlNamespaceManager.
XmlNamespaceManager nsmgr = new XmlNamespaceManager(doc.NameTable);

// Toevoegen van een naamruimte aan de XmlNamespaceManager.
nsmgr.AddNamespace("mijn", "urn:mijn:favoriete:naamruimte");

// Opzoeken van de waarde van een knoop met behulp van een XPath-expressie.
XmlNode node = doc.SelectSingleNode("/mijn:element", nsmgr);
```

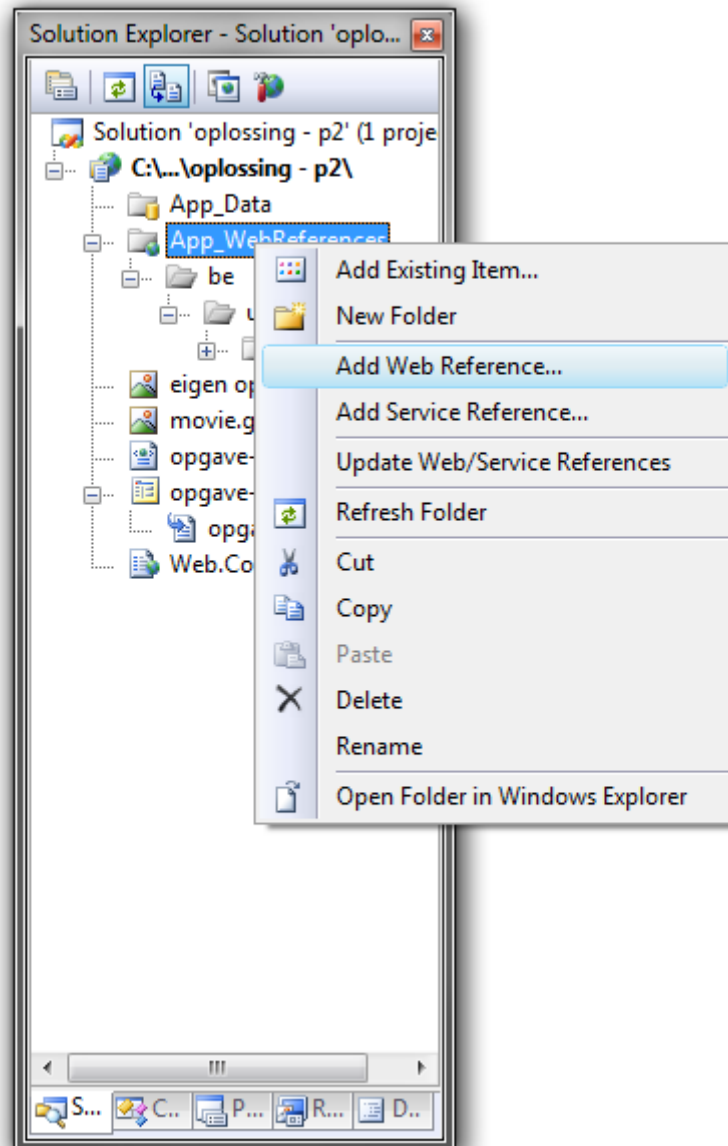
6. Gebruik van webservices in .NET

Een webservice kan zeer eenvoudig gebruikt worden in de .NET-omgeving. Het is namelijk zo dat alle communicatie tussen een webservice en de applicatie die ervan gebruikmaakt, verborgen blijft voor de ontwikkelaar. In de praktijk dient deze enkel rechts te klikken op een projectbestand, om vervolgens te kiezen voor "Add Web Reference". Nadien kan de toegevoegde referentie aangesproken worden als een normale klasse, met andere woorden de toegevoegde referentie kan op dezelfde manier gebruikt worden als een lokale klasse.

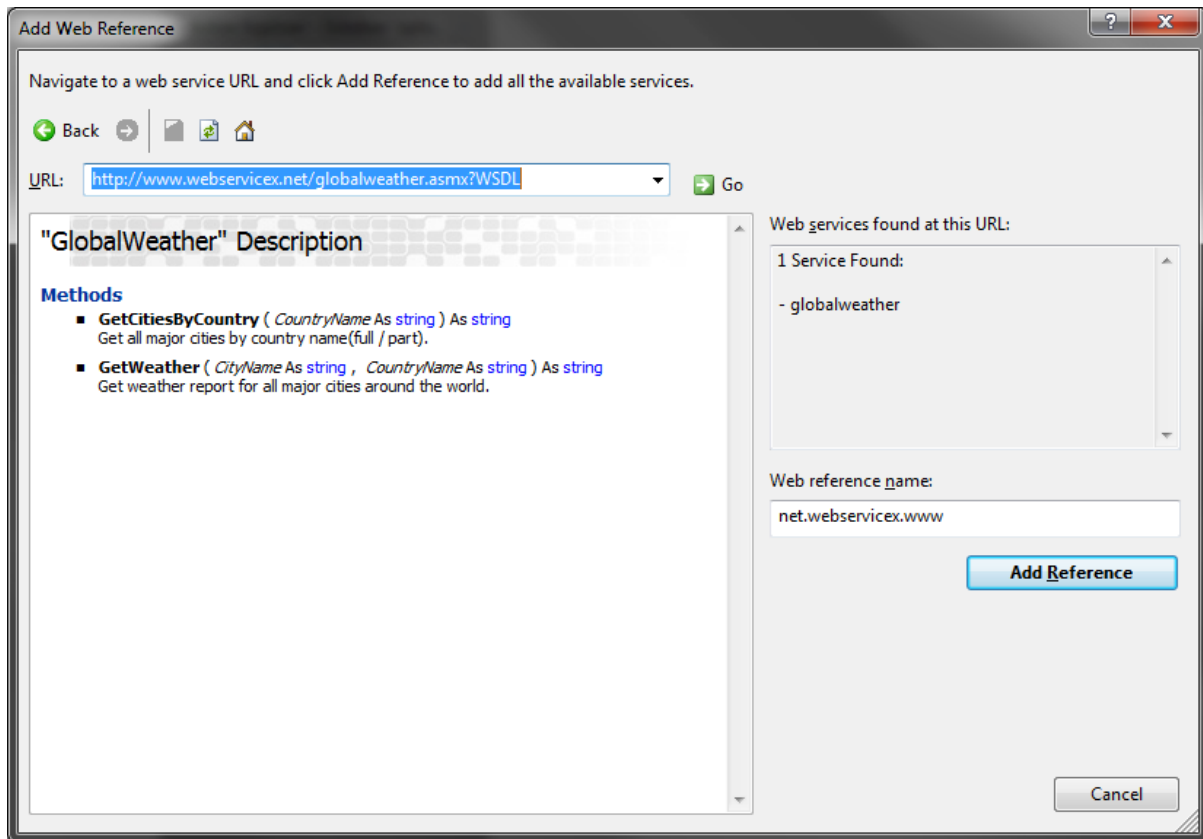
Zo is het mogelijk om, na het toevoegen van een referentie naar de webservice <http://www.webservices.net/globalweather.asmx?WSDL> (zie Figuur 2 en Figuur 3), de code in Listing 1 uit te voeren. De betreffende code zal de lijst van steden voor een land opvragen. Om dit te doen zal automatisch een verbinding gemaakt worden met de webservice en zullen verschillende berichten heen en weer gestuurd worden.

```
net.webservices.www.GlobalWeather gw =  
    new net.webservices.www.GlobalWeather();  
string cities = gw.GetCitiesByCountry("Belgium");
```

Listing 1: Het aanroepen van een webservice.



Figuur 2: Het toevoegen van een Web Reference in Visual Studio (1).



Figuur 3: Het toevoegen van een Web Reference in Visual Studio (2).

7. Deadline

Donderdag 18 april, om 14u00 STIPT

8. Algemene Opmerkingen en Tips

1. Wat dient er ingediend te worden op Minerva in “webservices_groep[x].zip” voor het practicum over MPEG-21 en W3C Webservices?

Eenvoudig gezegd: alles. Dit betekent dat alle bestanden binnen de solution-folder gezippt moeten worden.

2. Bepalen aantal trailers in DID.

In dit practicum is het de bedoeling om het aantal trailers in de DID dynamisch te bepalen in de ASP.NET-webapplicatie. Met andere woorden, het is niet de bedoeling om het aantal aanwezige trailers in de DID hard te coderen.

3. Valideren van XML-documenten.

Voor het valideren van XML-documenten t.o.v. een opgegeven XML Schema, kan het programma XSV gebruikt worden. Dit is een commandolijn-tool die op “Minerva > Documenten > Practica > Practicum_2_Extras” kan gedownload worden; meer informatie is te vinden op [deze pagina](#). Daarnaast zijn er ook online validators beschikbaar op het Internet. Het W3C XML Schema voor MPEG-21 DIDL kan in de zelfde map op Minerva teruggevonden worden.

4. Testen van de webservice

De webserver kan getest worden op:

<http://multimedialab.elis.ugent.be/URIResolver/URIResolver.asmx>.

Op "Minerva > Documenten > Practica > Practicum_2_Extras" kan ook de broncode en de bijbehorende assembly van de webservice zoals deze draait op de server gedownload worden. Deze moet in een application folder uitgevoerd worden. De betreffende broncode kan bijvoorbeeld gebruikt worden om op een eenvoudige manier URNs en URLs te kopiëren naar de DID die het traileralbum beschrijft. De code in het zip-bestand wordt gegeven ter illustratie, en hoeft verder niet gebruikt te worden voor het oplossen van de oefeningen.

9. Referenties

1. B. Rogge. Functionele metadata: een softwareraamwerk voor het opzetten van multimedietoepassingen. PhD Thesis, Ghent University, September 2002.
2. ISO/IEC, "ISO/IEC 21000-2:2003 Information technology -- Multimedia framework (MPEG-21) -- Part 2: Digital Item Declaration," March 2003
3. IETF, "RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax," August 1998
4. IETF, "RFC 1738 Uniform Resource Locators (URL)," December 1994
5. IETF, "RFC 2045 Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," November 1996
6. IETF, "RFC 2141 URN Syntax," May 1997

Opgave 1 - Opstellen van een DID voor een filmcollectie

In het eerste gedeelte van dit practicum is het de bedoeling om een DID te creëren. Deze DID bevat een filmcollectie die bestaat uit de trailers zoals hieronder opgesomd. Voor elk van de trailers zijn volgende gegevens beschikbaar.

1. Black Hawk Down
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/black_hawk_down.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:black_hawk_down
2. Brotherhood of the Wolf
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/brotherhood_of_the_wolf.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:brotherhood_of_the_wolf
3. Charlie Chocolate
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/charlie_chocolate.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:charlie_chocolate
4. Constantine
 - a. Poster:
<http://multimedialab.elis.ugent.be/posters/constantine.jpg>
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:constantine
5. Corpse Bride
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/corpse_bride.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:corpse_bride
6. Day after Tomorrow
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/day_after_tomorrow.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:day_after_tomorrow
7. Finding Neverland
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/finding_neverland.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:finding_neverland
8. House of Flying Daggers
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/flying_daggers.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:flying_daggers
9. Harry Potter and the Prisoner of Azkaban
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/harry_potter_azkaban.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:harry_potter_azkaban
10. Ice Age
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/ice_age.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:ice_age

11. Kingdom of Heaven
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/kingdom_of_heaven.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:kingdom_of_heaven
12. Monsters Inc.
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/monsters_inc.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:monsters_inc
13. Lord of the Rings: the Return of the King
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/return_of_the_king.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:return_of_the_king
14. Sin City
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/sin_city.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:sin_city
15. The Aviator
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/the_aviator.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:the_aviator
16. The Merchant of Venice
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/the_merchant_of_venice.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:the_merchant_of_venice
17. The New World
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/the_new_world.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:the_new_world
18. Underworld
 - a. Poster:
<http://multimedialab.elis.ugent.be/posters/underworld.jpg>
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:underworld
19. Van Helsing
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/van_helsing.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:van_helsing
20. War of the Worlds
 - a. Poster:
http://multimedialab.elis.ugent.be/posters/war_of_the_worlds.jpg
 - b. Trailer:
urn:mpeg:mpeg21:be:ugent:mmlab:trailers:war_of_the_worlds

Verder dient er in de DID ook nog een beschrijving van de ganse filmcollectie opgenomen te worden. In deze beschrijving moet uw groepsnummer opgenomen worden!

Merk op dat de betreffende films werden gedownload vanaf de trailer-website van Apple Computer en naar een AVI-bestand werden geëxporteerd met behulp van QuickTime

Pro. De resulterende AVI-container, met een ongecomprimeerd video- en audiospoor, werd vervolgens ingelezen en verwerkt met de Windows Media Encoder. Een dergelijke manier van transcoderen is doorgaans af te raden wegens de introductie van extra (onnodige) artefacten.

Extra informatie

1. Gebruik het voorbeeld van een digitaal muziekalbum als inspiratie voor het vastleggen van de opbouw van de filmcollectie-DID.
2. De poster is voor elke film als *beschrijvende* informatie te beschouwen.
3. Valideer uw DIDs ten opzichte van het DID-schema. Dit schema kan gevonden worden op "Minerva > Documenten > Practica > Practicum_2_Extras" en online op <http://vstudent.elis.ugent.be/itech/bestanden/21000-02.xsd>.
4. Het MIME-type van de gebruikte afbeeldingen is "image/jpg", terwijl het MIME-type van de gebruikte videosequenties "video/x-ms-wmv" is.
5. Het XML-fragment hieronder kan gebruikt worden als startbasis voor de filmcollectie-DID:

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS
      http://vstudent.elis.ugent.be/itech/bestanden/21000-02.xsd">

  <!-- ... -->

</DIDL>
```

Bestandsnaam: opgave-2-1.xml

(vergeet uiteraard niet het volledige project te uploaden naar Minerva!)

Opgave 2 - Creatie van een ASP.NET DID-applicatie

Maak een dynamische ASP.NET-webpagina die toelaat de bovenstaande DID te verwerken. Deze webpagina bevat de elementen en functionaliteit zoals hieronder beschreven.

Deel 1: Minimale opmaak webpagina.

- Bovenaan de pagina staat de beschrijving van de filmcollectie. Deze beschrijving dient uit de DID gehaald te worden.
- Het totale aantal trailers dat beschreven wordt in de DID dient ergens op de pagina vermeld te worden (dit aantal wordt berekend op basis van de DID).
- Onder de beschrijving kunnen drie verschillende posters van trailers teruggevonden worden. De locatie van deze posters wordt eveneens uit de DID gehaald.
- De selectie van de drie posters gebeurt willekeurig en verandert telkens wanneer de pagina opnieuw ingeladen wordt.

Deel 2: Minimale functionaliteit webpagina.

- Bij het tonen van drie posters, is er een knop aanwezig die toelaat om terug op een willekeurige manier drie posters te tonen.
- Indien op een poster geklikt wordt, wordt een multimediaspeler getoond. Deze mediaspeler zal de trailer van de film afspelen.
- Tot slot staat er nog een knop op het scherm waarmee terug naar het overzicht van de filmcollectie gegaan kan worden.

Extra informatie

1. Gebruik de namespaces `System.Xml` en `System.Xml.XPath`.
2. Maak gebruik van de webservice zoals beschikbaar op de volgende URL om de URN uit de DID om te zetten naar een URL:

<http://multimedialab.elis.ugent.be/URIResolver/URIResolver.aspx>

Deze webservice zal de werklust met betrekking tot het stromen van audio en video verdelen over twee verschillende streaming servers door afwisselend een andere URL terug te geven. Door naar de betreffende URL te surfen krijgt u een overzicht van de webmethoden die door de webservice worden aangeboden, alsook een pagina om de webservice uit te testen.

3. Een Windows Media Player-object kan als volgt opgenomen worden in een ASP.NET-pagina:
 - a. Ga naar het Design-overzicht van de ASP.NET-pagina.
 - b. Bekijk de pagina vervolgens met behulp van het HTML-overzicht.
 - c. Voeg op de plaats waar de mediaspeler moet komen de volgende HTML-code toe: `<DIV id="itechx" runat="server"></DIV>`. Het `x`-symbool dient vervangen te worden door het eigen groepsnummer.
 - d. Ga terug naar het Design-overzicht.
 - e. Om de mediaspeler zichtbaar te maken, kan in de ASP.NET-pagina het volgende codefragment opgenomen worden:

```
itechx.InnerHtml=""+"<OBJECT ID='itechx' width='480' "+
    "height='336' CLASSID='CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6' " +
    "standby='Loading Microsoft Windows Media Player components...' " +
    "TYPE='application/x-oleobject' width='200' height='200' VIEWASTEXT>" +
    "<PARAM NAME='url' VALUE='mms://mijn.filmcollectie.server'>" +
    "<PARAM NAME='AutoStart' VALUE='true'>" +
    "<PARAM NAME='uiMode' VALUE='mini'>" +
"</OBJECT>";
```

4. Meer informatie over de Windows Media Series kan teruggevonden worden op de volgende URL: <http://www.microsoft.com/windows/windowsmedia/default.aspx>. Informatie over het inbedden van een mediaspeler in een webpagina kan op de volgende URL teruggevonden worden: [http://msdn2.microsoft.com/en-us/library/bb262918\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/bb262918(VS.85).aspx).
5. Bovenstaande opgave zou bijvoorbeeld ook kunnen uitgewerkt worden met behulp van MPEG-4, het QuickTime-raamwerk en de Darwin Streaming Server.

Bestandsnaam: opgave-2-2.aspx

(vergeet uiteraard niet het volledige project te uploaden naar Minerva!)