# XML Holland Code Challenge

TEI naar EPUB conversie met XProc

Geert Josten, Dayon BV

### Inhoudsopgave

[Inhoudsopgave 1](#_Toc306131681)

[Woord vooraf 1](#_Toc306131682)

[Verantwoording 1](#_Toc306131683)

[Installatie en gebruik 2](#_Toc306131684)

[Ontwerp 4](#_Toc306131685)

[Eindresultaat 7](#_Toc306131686)

### Woord vooraf

Het TEI Milioenenstudien voorbeeldbestand is tamelijk plat van structuur. Een recht toe recht aan vertaling naar bijvoorbeeld XHTML om in EPUB op te nemen kost weinig moeite. Dat leek me geen winnende oplossing, hoe elegant opgezet dan ook. Ik heb mezelf de lat dus hoger gelegd.

De oplossing die ik voor ogen genomen heb is een die qua architectuur in staat is (potentieel dus) meerdere invoerformaten te verwerken, zelfs in gemengde vorm. Met het single-source publishing idee in mijn achterhoofd wilde ik ook graag de mogelijkheid inbouwen om later naast EPUB andere uitvoerformaten te kunnen ondersteunen, bijvoorbeeld PDF. Ook wilde ik de mogelijkheid hebben om zo veel mogelijk voorzieningen van EPUB te benutten, waaronder allerlei navigatiefunctionaliteit. Last but not least wilde ik enige vrijheid bieden in de look en feel van de uitvoer.

Om dit te realiseren vertaal ik de input eerst naar een ‘generiek’ formaat. Ik spreek daarbij over het EBook-formaat, wat overigens een aan mijn eigen fantasie ontsproten formaat is en gemakshalve gebaseerd op XHTML. Alle verrijkingsslagen, zoals invoegen van een inhoudsopgave, index, lijst van figuren en dergelijke, worden op dit generieke formaat losgelaten. Pas daarna worden bewerkingen gedaan, in een aparte stap, ten behoeve van het gekozen uitvoer formaat. Door middel van een soort stijlboek kan de uitvoer daarbij een geheel eigen look en feel gegeven worden.

### Verantwoording

De conversie wordt volledig aangestuurd vanuit XProc. Daarbij wordt veel gebruik gemaakt van de daarin ondersteunde standaarden zoals XSLT, XQuery en XInclude. Ik gebruik echter ook XML Calabash specifieke uitbreidingen op die taal, die door veelvuldig gebruik door de gemeenschap ontstaan zijn. Dit leek me echter wenselijker dan bijvoorbeeld het maken van de EPUB-zip met een losstaande stap, buiten XProc om, te realiseren.

Uiteindelijk heb ik slechts enkele regels Java-code hoeven schrijven om één kleine extra uitbreiding op XML Calabash te schrijven. Deze uitbreiding wordt voor exact één doel gebruikt: het uitlezen van de JAVA\_HOME (of eigenlijk java.home) instelling uit de Java Run-time omgeving. Dit is gedaan om een aanroep van epubcheck op een wat elegantere manier in te kunnen bouwen in het proces. Het vormt standaard onderdeel van deze conversie.

Het tussenformaat is, zoals gezegd, gebaseerd op XHTML. Ik had wat anders kunnen kiezen, een bestaande standaard zoals DITA, of zelfs TEI, maar ik wilde een tussenformaat dat dichter bij het eindresultaat lag. XHTML voldoet voor mijn applicatie, maar zou later, als daar reden toe is, zonder veel problemen vervangen kunnen worden door wat anders.

Het stijlboek is enigszins rudimentair van opzet. Ik had voor volledige abstractie kunnen kiezen, maar dat vond ik het doel voor deze code challenge voorbij schieten. Ik vond vooral dat je wat keuze moet kunnen hebben en de look en feel moet kunnen bijstellen zonder diep in de applicatiecode te hoeven duiken. Daarvoor leek me de gekozen oplossing afdoende. Ook hier geldt dat het vrij eenvoudig verbeterd en uitgebreid kan worden als er specifieke behoeftes zijn.

Helaas had ik geen echte e-reader bij de hand. De epubs zijn, na validatie met epubcheck, uitsluitend getest met Adobe Digital Editions. Deze is in bepaalde opzichten tamelijk beperkt qua mogelijkheden, vooral qua opmaak. Ik denk en hoop dus dat echte e-readers het doorgaans even goed of zelfs beter doen.

### Installatie en gebruik

De applicatie is aangeleverd in de vorm van een zip. Naast deze zip is alleen nog een Java Run-time installatie nodig. XML Calabash, epubcheck, de applicatie zelf en alle overige benodigdheden zitten allemaal in de zip.

De zip bevat de volgende zaken:

* In/
  + Millioenenstudien voorbeeld met bijbehorend voorbeeld configuratie
* Lib/
  + XML Calabash met afhankelijkheden (Saxon en Apache commons)
  + Epubcheck-1.2 met afhankelijkheden (Saxon en Jing)
  + Eigen uitbreiding op XML Calabash met config
* Log/
  + Lege map ten behoeve van proces en debug logging
* Out/
  + Lege map ten behoeve van eindresultaat
* Src/
  + Broncode omvattende de XML Calabash eigen uitbreidingen en de applicatiecode
* Tmp/
  + Lege map ten behoeve van tijdelijke resultaten
* Style/
  + Bevat het ‘stijlboek’. Styles.xml bevat generieke stijlinformatie, een default opmaak, en nog twee extra opmaakkeuzes: dbnl (ontleend aan de website van dbnl) en medieval (een wat speelsere stijl, om mogelijkheden te tonen)
* Dtd/
  + Bevat de DTD catalog met instellingen voor TEI dtd en ook de DTD en bijbehorende modules zelf.
* Logging.properties
  + Configuratie van standaard logging van XML Calabash.
* CatalogManager.properties
  + Configuratie van DTD catalog resolver.
* Run.bat
  + Aansturing van de applicatie, starten zonder parameters voor toelichting.

De applicatie wordt helaas alleen geleverd met aansturing voor de command-box van Windows (zie verderop voor de volledige Java-aanroep). Het is voornamelijk bedoeld om de aanroep wat te vereenvoudigen. Om het voorbeeld van de code challenge te verwerken dient de volgende aanroep gedaan te worden:

run.bat in\mult001mill01\_01.xml

De toevoeging “verbose=false” zorgt er voor dat minder procesinfo op het scherm getoond wordt. Wordt “debug=true” toegevoegd dan wordt in de log map diverse tussenresultaten als XML weggeschreven en wordt juist extra debug-info op het scherm getoond.

Op zelfde manier kunnen nog een boel andere zaken aangestuurd worden, zoals:

* output-style

Voor keuze van één van de drie beschikbare uitvoerstijlen: #default, dbnl of medieval

* styles-file

Relatief pad vanaf de root van de ebook converter naar een alternatief voor styles/styles.xml

* template-file

Relatief pad vanaf de root van de ebook converter naar een alternatief voor styles/default/input-template-nl.xml, bijv. styles/default/input-template-en.xml of styles/bare-template.xml

* toc-max-level

Tot hoe diep secties in Inhoudsopgave meegenomen moeten worden. Default: 1

* toc-page-break

Na hoeveel Inhoudsopgave items een page-break geforceerd moet worden. Default: 20

* index-page-break

Na hoeveel Index items een page-break geforceerd moet worden. Default: 15

* notes-page-break

Na hoeveel Voetnoten op de Notes pagina een page-break geforceerd moet worden. Default: 5

* loi-page-break

Na hoeveel Illustraties op de List of Illustrations pagina een page-break geforceerd moet worden. Default: 20

* lot-page-break

Na hoeveel Tabellen op de List of Tables pagina een page-break geforceerd moet worden. Default: 20

Voor degenen die run.bat niet kunnen gebruiken: de achterliggende Java-aanroep luidt als volgt. NB: dit hoort op één regel, maar is omwille van leesbaarheid over meerdere regels weergegeven:

java

-cp .;lib\calabash-0.9.35.jar;lib\grtjn-xproc-utils.jar

-Dcom.xmlcalabash.phonehome=false

-Djava.util.logging.config.file=logging.properties

com.xmlcalabash.drivers.Main

-D

-E org.apache.xml.resolver.tools.CatalogResolver

-U org.apache.xml.resolver.tools.CatalogResolver

-c lib/grtjn-xproc-utils.xml

-i source=**in\mult001mill01\_01.xml**

src\nl\grtjn\xproc\ebook\main.xpl

**verbose=true**

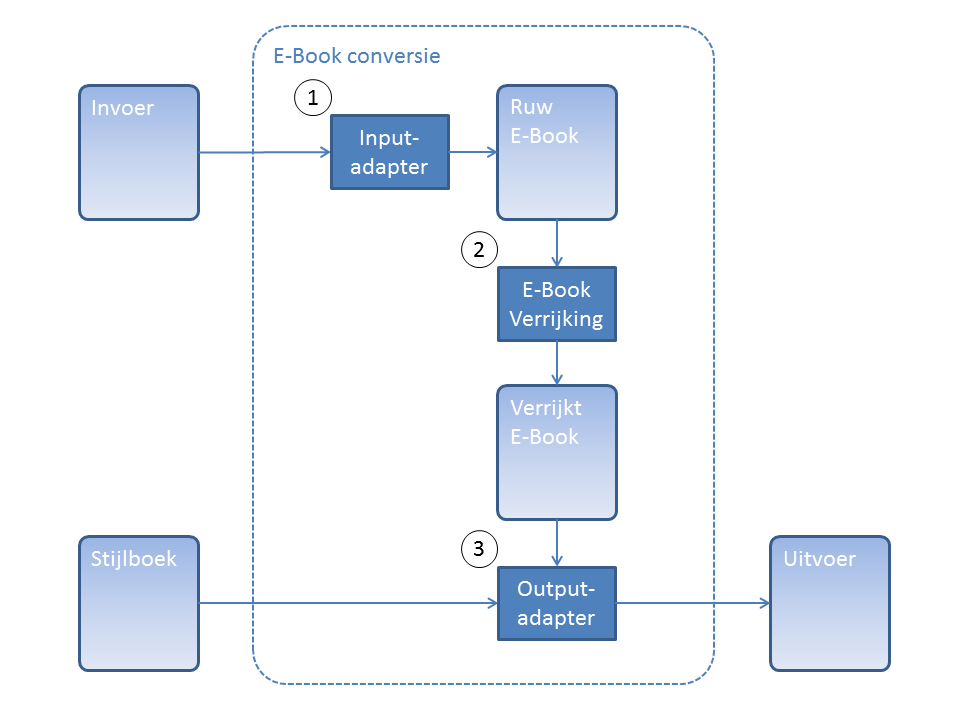
Extra opties zoals verbose=true moeten achteraan toegevoegd worden.

NB: Voor gebruik van achter een proxy en meer details rond de aanroep en het gebruik: zie INSTALL.txt.

### Ontwerp

De applicatie is opgezet als een 3-traps raket:

1. Input-adapter: verwerken van losse input tot gebundeld tussenformaat
2. Verrijking: gebundeld tussenformaat verrijken/bewerken
3. Output-adapter: tussenformaat vertalen naar gewenst uitvoerformaat

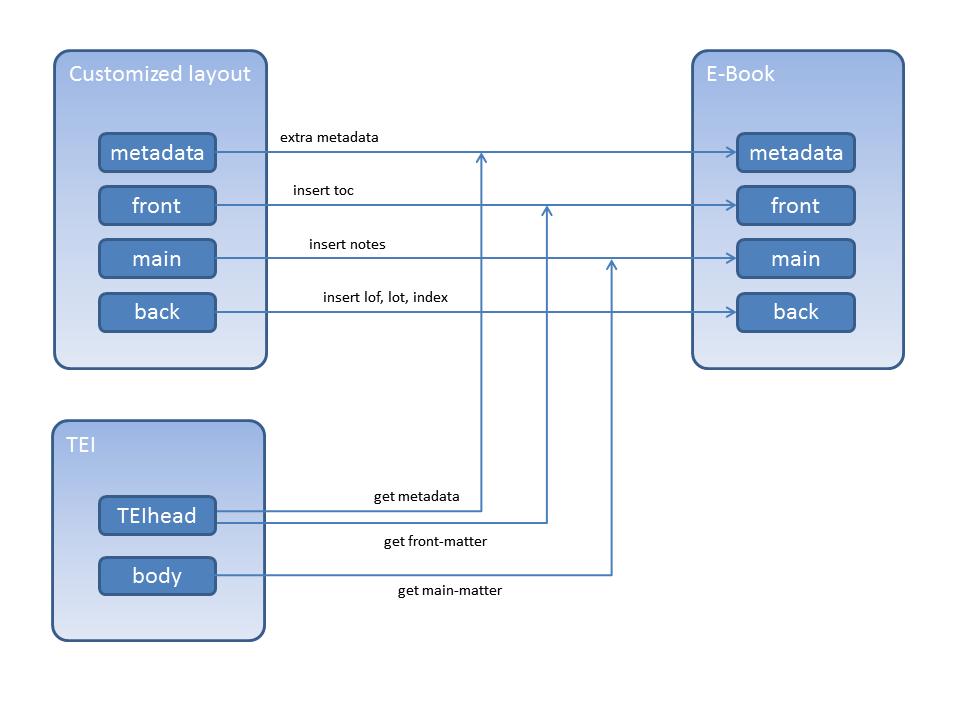


Input-adapter:

Het voorbeeld bestaat uit slechts één bestand. Het is echter denkbaar dat je soms meerdere onderdelen zou willen bundelen tot één uitvoerbestand. Om nauwkeurig aan te kunnen geven wat de gewenste samenstelling is, wordt de mogelijkheid geboden om een configuratiebestand als invoer op te geven. Hierin kan nauwkeurig opgegeven worden in welke volgorde welke onderdelen ingevoegd moeten worden en welk bestandsformaat ze elk hebben. Momenteel ondersteunde bestandsformaten zijn o.a.:

* TEI
* (X)HTML

Binnen invoerbestanden wordt onderscheid gemaakt tussen metadata, voorwerk, hoofdwerk en nawerk. In het volgende schema is een en anders schematisch weergegeven:



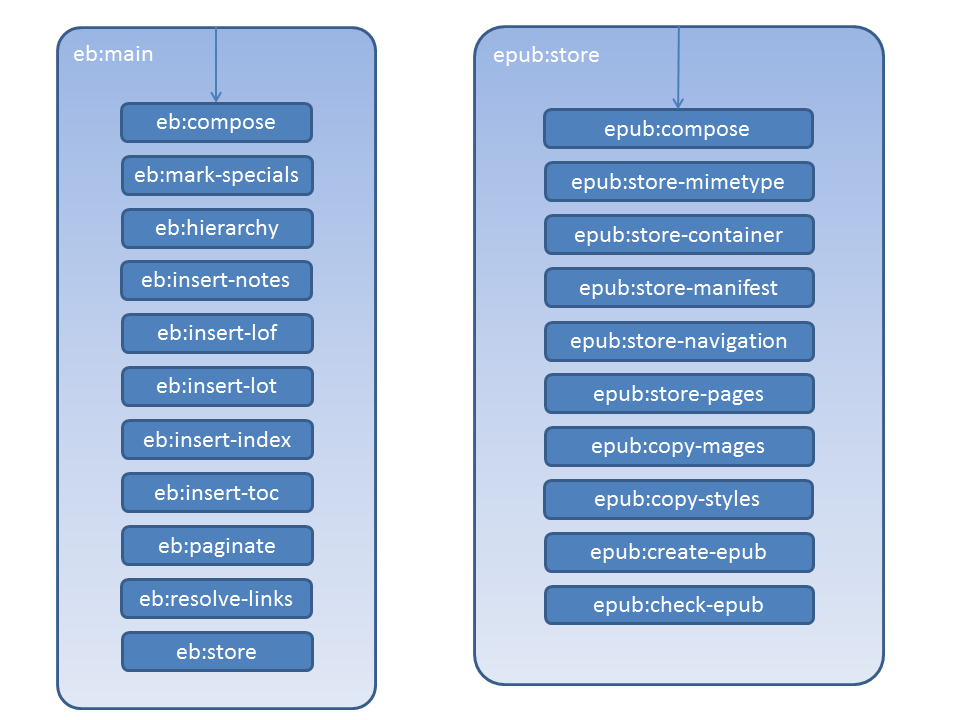
Verrijking:

Het is niet ongebruikelijk om bijvoorbeeld een automatische inhoudsopgave of register toe te voegen. Dit wordt gerealiseerd in de verrijkingsstap. In de configuratie die als invoer meegegeven kan worden, kan tevens aangegeven worden waar en welk soort verrijking ingevoegd dient te worden. Ondersteunde verrijkingen omvatten o.a.:

* Inhoudsopgave
* Register
* Lijst van figuren
* Lijst van tabellen
* Voetnotenpagina (ipv losse voetnoten onderaan elke pagina)

Deze gegenereerde pagina’s gaan gepaard met hyperlinks van elk item naar de juiste positie binnen een pagina. Voor voetnoten geldt zelfs dat deze twee kanten op gaan.

Om deze verrijking te kunnen realiseren worden in de beginfase alle interessante elementen uit de invoer voorzien van markeringen. Ook wordt de dieptestructuur in een document op basis van koppen afgeleid. Aan het einde wordt geavanceerde paginering toegepast op basis van gegenereerde en uit input overgenomen page-breaks. Het hoofdproces doorloopt in hoofdlijn de volgende stappen:



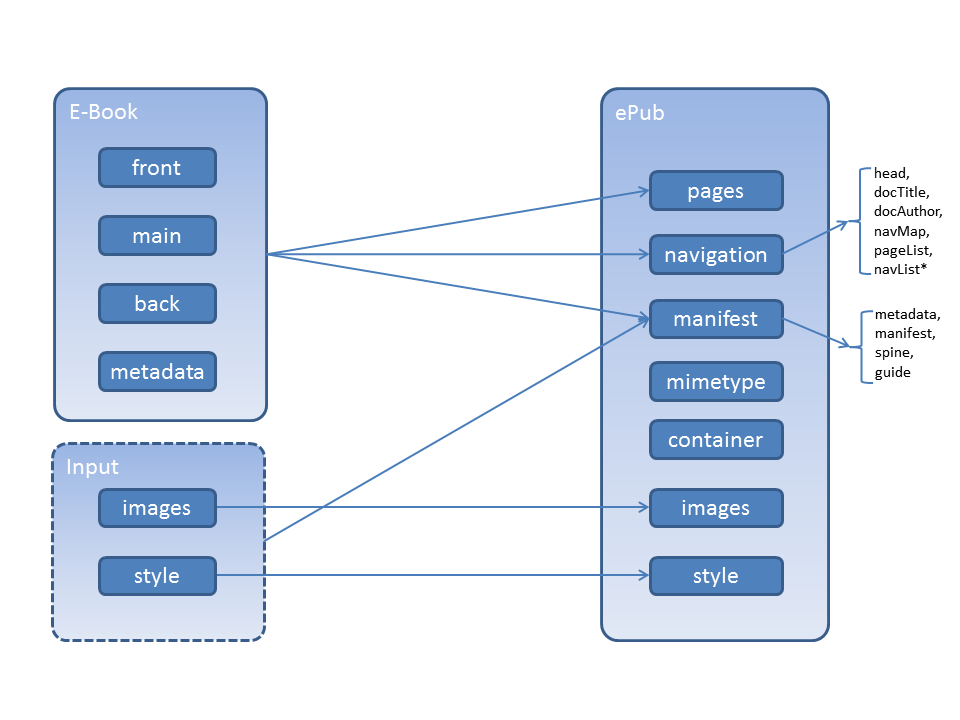
Output-adapter:

Het is erg gebruikelijk om vanuit één bronformaat meerdere publicaties te genereren. In plaats van slechts één uitvoerformaat (EPUB) te ondersteunen, wordt de mogelijkheid geboden, dit (later) uit te breiden met andere formaten (zoals bijvoorbeeld PDF). Tevens kan met behulp van een zogeheten stijlboek een bepaald uiterlijk meegegeven worden aan het eindresultaat. Het stijlboek ondersteund de volgende stijlen:

* Default (een simpele, zakelijke stijl)
* Dbnl (ontleend aan de opmaak van de website van DBNL)
* Medieval (een wat speelsere stijl, die wat meer mogelijkheden laat zien)

Voor het construeren van een epub moet het nodige verzameld en gemetadateerd worden. Naast een manifest waarin metadata en de complete inhoud gespecificeerd moet worden, wordt ook een navigation bestand gevraagd waarin naast inhoudsopgave nog allerlei andere overzichten gevraagd worden. Uiteraard moet de inhoud zelf, bestaande uit de feitelijke pagina’s, afbeeldingen en alle stijlbestanden toegevoegd worden.

Schematisch kan het min of meer als volgt weergegeven worden:



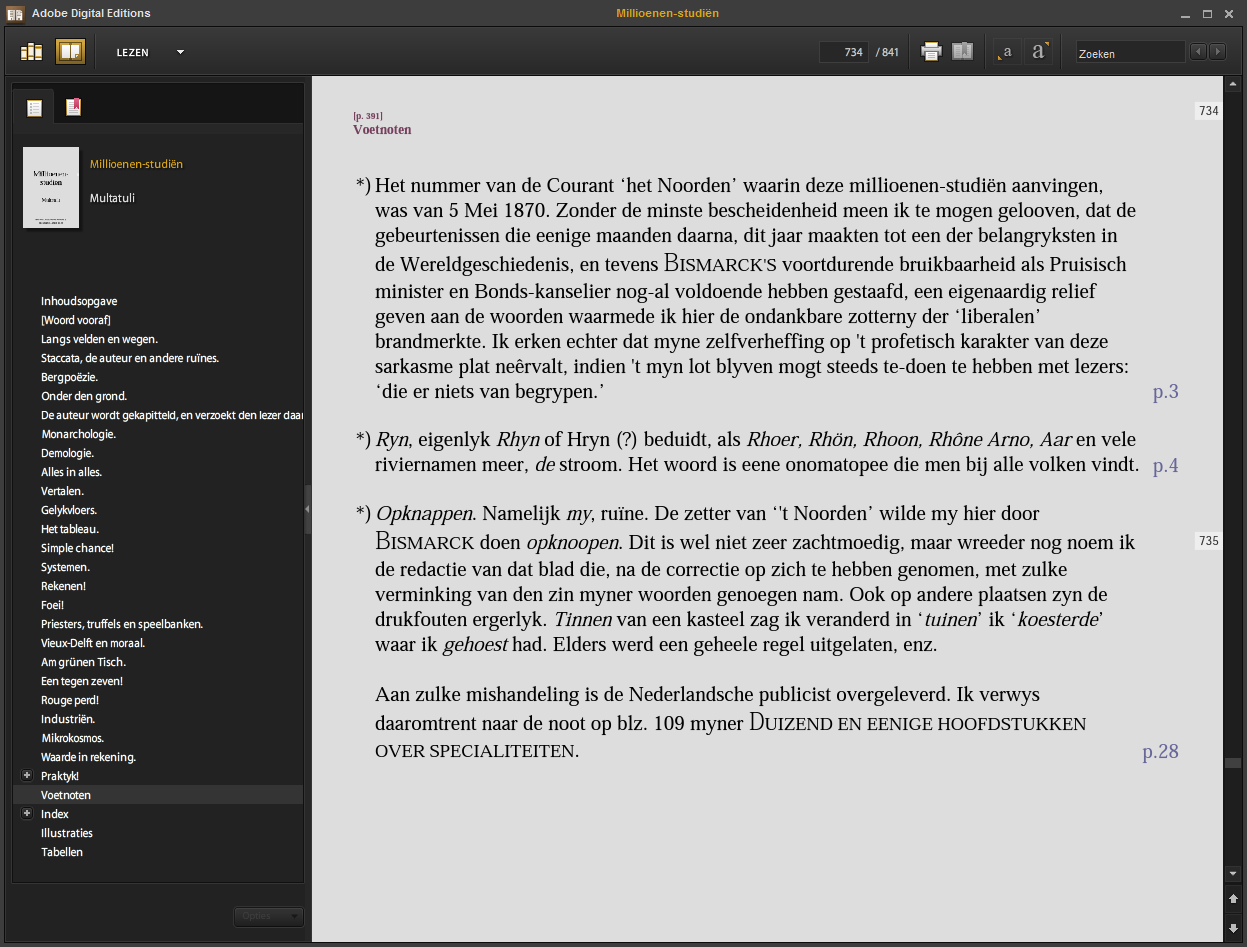
### Eindresultaat

Het ging natuurlijk uiteindelijk om de epub. Hieronder een screenshot van elk van de beschikbare stijlen, genomen van verschillende pagina’s om alvast een impressie van het geheel te geven.

Medieval stijl:



DBNL stijl:



Default stijl:

