

# Homer MultiText: het editieproces

## I. WAT ZIJN WE EIGENLIJK AAN HET DOEN?

Het editen van een VenetusA-folio verloopt in een aantal stappen:

### HOOFDTEKST:

- **Paleografisch sample** maken van het eerste vers op de pagina
  - Je werkt via de browser, in: <https://github.com/hmteeditors/leiden> in de directory "collections"
  - Je maakt een zgn. csv-bestand (comma separated values), d.w.z. een tabel
  - *Waarom?* M.b.v. deze samples wordt een database aangelegd waarin toekomstige gebruikers alle verschijningsvormen van bepaalde glyphen kunnen oproepen en vergelijken
  - De tabel bestaat uit 4 kolommen:
    - (1) het ID van je observatie, geformuleerd, als urn: urn:cite:op:leiden1.1 (=observatie nr. 1 van team 1 uit Leiden)
    - (2) de locatie van je observatie in het corpus van de Griekse literatuur, als urn: urn:cts:greekLit:tlg0012.tlg001.msA:13.51@ε (d.w.z. in auteur 0012 (=Homerus), in tekst 001 (=Ilias), in tekstvariant msA (=VenetusA), in boek 13, vers 51 lezen wij ε)
    - (3) een citaat uit de afbeelding van de folio (je "knipt" met de "Cite & Quote Image-tool"), als urn: urn:cite:hmt:vaimg.VA165RN-0336@0.2,0.1983,0.026,0.0248
    - (4) opmerkingen: ligaturen meld je hier; *omschrijf* ze ook (dus niet alleen "ligature", maar "ετ ligature")
  - **vuistregel:** selecteer zoveel mogelijk de "blote" letter, d.w.z. zonder diakritische tekens; als dat niet lukt (en je bv. niet om het accent heen kan) geef dan (nog steeds) weer wat je ziet (dus de letter met accent).
  - **Commit** (in de github, dus gewoon klikken op "Commit Changes")
- **Versregels indexeren**
  - Je werkt via de browser, in: <https://github.com/hmteeditors/leiden> in de directory "indices"
  - Je maakt een csv-tabel: je werkt in het csv-bestand dat venA-textToImage-Iliad.csv heet
  - *Waarom?* M.b.v. deze tabel wordt een vers uit de VenetusA gekoppeld aan een vers van de *Ilias*, zodat je kunt zeggen: "dit stukje folio is vers 51 uit boek 13 van de *Ilias*."
  - De tabel bestaat uit 2 kolommen
    - (1) de locatie van het vers in het corpus van de Griekse literatuur, als urn: urn:cts:greekLit:tlg0012.tlg001.msA:13.51
    - (2) een citaat uit de afbeelding van de folio: urn:cite:hmt:vaimg.VA165RN-0336@0.197,0.1976,0.378,0.0338
  - **Commit** (in de github, dus gewoon klikken op "Commit Changes")
- **Hardcopy-editen**
  - Je werkt op een papieren uitdraai van de Perseus-tekst
  - Je vergelijkt deze uitdraai met de tekst op de foto van je folio en maakt een aantekening van alle afwijkingen
  - *Waarom?* Het Homer Multitext-project maakt een diplomatieke editie waarbij de tekst van alle manuscripten precies zo wordt weergegeven zoals het er staat.
- **xml-editen**
  - Je werkt in de *virtual machine*, in de directory Desktop/vagrant/leiden/editions/Iliad
  - Je voert de resultaten van het hardcopy-editen in in xml-format

- Je tagt alle eigennamen, plaatsnamen en ethnonymen
  - ✓ bij het taggen wijs je ook ID's toe aan items; zoek de ID-nrs op in de authoritylists (te vinden op <https://github.com/homermultitext/hmt-authlists>)
  - ✓ staat jouw eigennaam of plaatsnaam nog niet in de authoritylist? Dien dan een "issue" in via de issuetracker. Check van te voren goed het format van de issues en wat voor informatie je dient aan te leveren.
  - ✓ twijfel je over de weergave van interpunctie, tekstkritische tekens of tags? Raadpleeg <http://homermultitext.github.io/hmt-editors-guide/editorial-policies/>
- Je tagt bijzonderheden van de hand van het manuscript:
  - ✓ doorhalingen (<del>) en verbeteringen (<corr>)?
  - ✓ gaten en vlekken (<gap>)
  - ✓ Check: <http://homermultitext.github.io/hmt-editors-guide/editorial-policies/>
- *Waarom?* Door teksten van *markup* te voorzien maak je hem "semantisch" doorzoekbaar voor de toekomstige gebruiker.
- **Add & Commit** (in de Virtual Machine, dus d.m.v. Command Lines; zie hieronder)

#### SCHOLIA:

- **Paleografisch sample** maken van de eerste 40 glyphen van een Main Scholion
  - *waar?* In dezelfde csv-tabel als je andere paleografische observaties.
  - *let op:* scholia hebben wel een andere urn: bv. "urn:cts:greekLit:tlg5026.msA.hmt:13.A1" = "de eerste paleografische observatie uit de scholia op boek 13 door team A"
  - **Commit** (in de github, dus gewoon klikken op "Commit Changes")
- **Scholia indexeren**
  - Je werkt via de browser, in: <https://github.com/hmteeditors/leiden> in de directory "collections"; werk door in de bestaande csv-bestanden, met een provisorische nummering (bv. 13.D1= eerste scholion van team D in boek 13, etc.)
  - Je maakt voor elk type scholion (mainScholia, exterior, interior, interlinear, intermarginal) een csv-tabel aan
  - *Waarom?* M.b.v. deze tabel wordt een scholion uit de Griekse literatuur gekoppeld aan een locatie op een afbeelding en aan een folio. Daarmee zeg je eigenlijk "dit scholion is *hier* te vinden in de Venetus A"
  - De tabel bestaat uit 3 kolommen
    - (1) de locatie van het scholion in het corpus van de Griekse literatuur, als urn: urn:cts:greekLit:tlg5026.msAext:13.001 (dit is een Homerus-scholion (=tlg5026), en wel een "exterior" in VenetusA (msAext), in boek 13; het is het eerste "exterior" scholion in boek 13; omdat wij in verschillende teams werken, hanteren wij even een provisorische nummering per team ("A1", "B1" etc.) en trekken we die later recht)
    - (2) een citaat uit de afbeelding van de folio: urn:cite:hmt:vaimg.VA164RN-0335@0.829,0.3802,0.06,0.0195
    - (3) de ID van de folio waarop het scholion te vinden is: urn:cite:hmt:msA.164r
  - **Commit** (in de github, dus gewoon klikken op "Commit Changes")
- **Scholia transcriberen**
  - Ontcijfer de scholia vanaf de foto's van de folio!
- **xml-editen**
  - Je werkt in de *virtual machine*, in de directory Desktop/vagrant/leiden/scholia

- Er is per type scholia 1 xml-bestand; werk dus ook hier met provisorische nummering (dezelfde als bij je csv-tabel) [**→ hier is een workflow-probleem: we moeten niet allemaal tegelijk in dezelfde scholia-bestanden werken**]
- Je voert de resultaten van je transcriptie in in xml-format
- Je tagt alle eigennamen, plaatsnamen en ethnonymen
- doorhalingen (<del>) en verbeteringen (<corr>)?
- gaten en vlekken (<gap>)
- Je tagt alle verbeteringen, afkortingen (<choice><abbr>wat er staat</abbr><expan>de uitgepakte versie</expan></choice>), citaten (<q>), en gebruiken van “meta-taal” (<waw>)
- Check: <http://homermultitext.github.io/hmt-editors-guide/editorial-policies/>
- **Add & Commit** (in de Virtual Machine, dus d.m.v. Command Lines; zie hieronder)

## VALIDATIE

Onderstaand scenario is nog een beetje toekomstmuziek, maar dit is uiteindelijk de bedoeling:

- de VM zal ons werk **valideren**: d.w.z. door de morfologische parser halen, checken op syntactisch correcte xml, op zinvol Grieks en op integriteit van verwijzingen.
- de VM geeft ons feedback waar we grofweg op 2 manieren op kunnen reageren:
  - wij hebben iets *fout* gedaan en moeten de fout herstellen
  - wij hebben iets *nieuws* gedaan en moeten de VM dit nieuwe leren (bv. dat een bepaalde orthografische variant heus goed Grieks is); dit zullen we doen d.m.v. Issue Tracker, een digitale vraagbaak/meldpunt van HMT waaraan wij dit soort gevallen kunnen melden

Voordat dit kan zal eerst nog een aantal zaken geconfigureerd worden, dus dit moet even wachten. Wie benieuwd is vindt alvast meer informatie op: <http://homermultitext.github.io/hmt-docs/totald/validation/>

## II. ROUTINE IN DE VIRTUAL MACHINE

Elke keer als je in de Virtual Machine werkt, moet je:

- **VM opstarten:**
  - in de gitbash:
    - ✓ cd desktop (**c**hange **d**irectory naar desktop)
    - ✓ cd editors-vm
    - ✓ vagrant up
  - in de terminal in de VM:
    - ✓ cd /vagrant/leiden
- **git pull**: hiermee zorg je dat jouw lokale versie van de Virtual Machine up to date is.
  - ✓ N.B. zorg dat je in de juiste directory (d.w.z. "leiden") zit te *pullen*, anders gebeurt er niks!
- Je werkt wat.
- Je **commit** je bestand:
  - **git add** "bestandsnaam" (bv. git add "164v.xml")
  - **git commit -m** "message" (bv. git commit -m "de persoonsnamen getagd")
    - ✓ als je vergeet -m in te typen kom je op een "gek" scherm uit; toets dan: ctrlO enter ctrlX
  - **git push**

- ✓ soms krijg je een conflict van versies als je hebt zitten werken in een versie van het bestand die niet de laatste is;
- ✓ sla dan je werk even provisorisch op buiten de virtual machine (als een tekstbestandje of als xml-bestand in Notepad++)
- ✓ verwijder in de directory het betreffende bestand
- ✓ pull opnieuw zodat de meest recente versie van het bestand verschijnt
- ✓ plak jouw werk in het bestand
- VM uitzetten:
  - in de gitbash: `vagrant halt`

**ONDERHOUD:**

- run eens in de zoveel tijd het command `refresh-hmt.sh` zodat jouw versie van de authoritylists en validatieschema's geupdate wordt