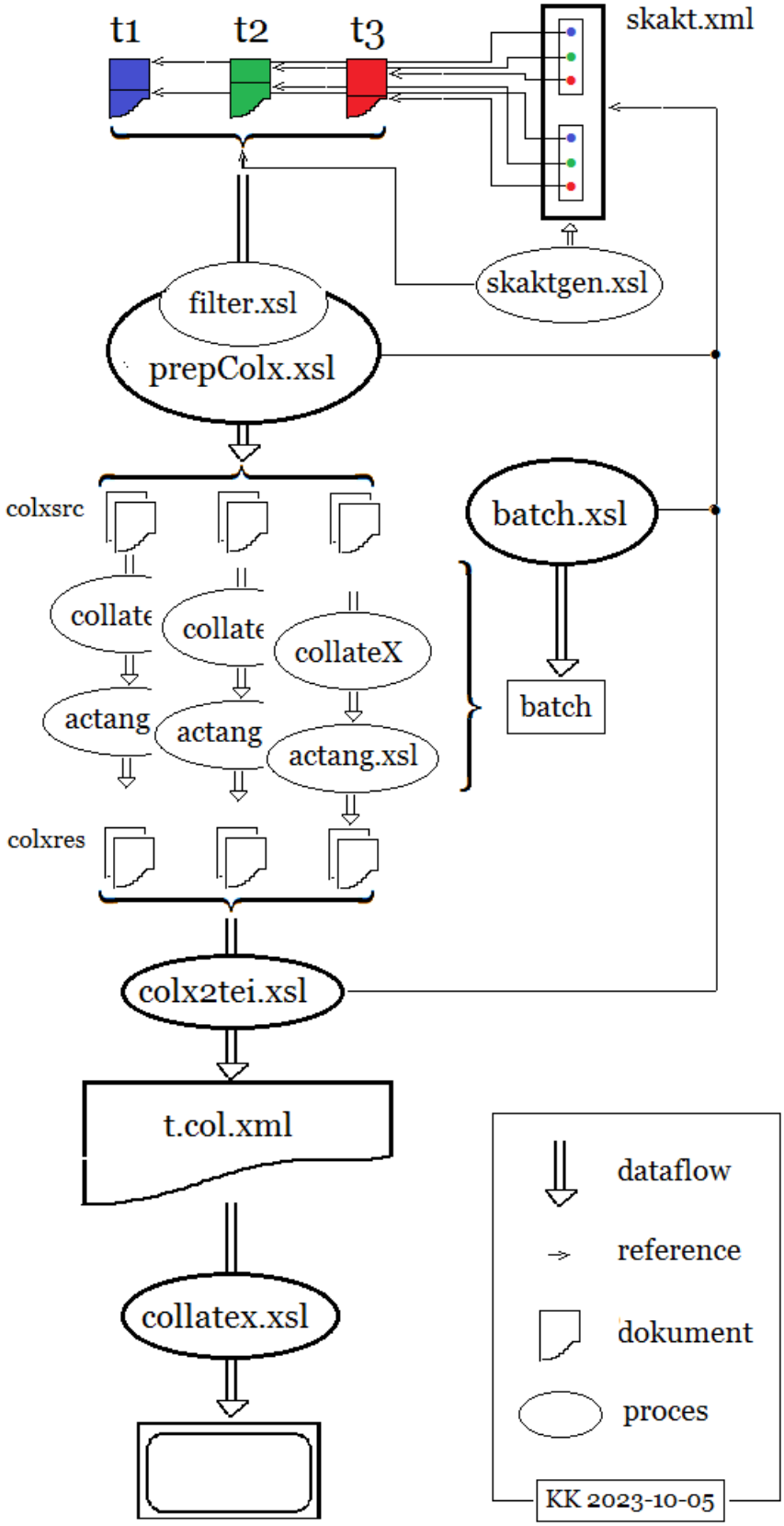
Fire trin

til tekstkollation med *collateX*

Karsten Kynde



Dette er en beskrivelse af en anvendelse af kollationsprogrammet *CollateX* og den dertil hørende gøteborgmodel.

Forklaring til *flowdiagrammet* side 1:

Udgangspunktet er tre sammenlignelige tekster, *t1*, *t2* og *t3*, hvis forskelle vi vil vise i et synoptisk apparat i tre spalter. Diagrammet illustrerer processen gennem fire trin, som delvist kan passes ind i Gøteborgmodellens fem anliggender [1].

En skakt er en datastruktur som sammenkæder sammenlignelige passager i forskellige tekstfiler. Ideen er at tilvejbringe en 'lodret' forbindelse som udpeger de sammenlignelige punkter i de tre udgavers 'vandrette' tekst. Som man vil kunne se er det skakterne der styrer de enkelte dele af processen.

Som skakterne rent praktisk er udformet, kan de maskinelt udtrækkes af de indgående filer. Dette er beskrevet i vejledningen til skaktgen.xsl [2]. Hvordan skakterne i øvrigt er tilvejebragt er dog irrelevant for det følgende.

Trin 1

prepColx.xsl[1]

filter.xsl

Indledningsvist vil vi *filtrere* de elementer som vi ikke vil have med. Det kan dreje sig om tekstrettelser som udgiver har foretaget, som vi gerne vil have i læseteksten men ikke i kollationeringen med andre tekster. Det gælder formentlig også interne varianter, hvor man primært er interesseret i den færdige version. Denne del er den mest sandsynlige kandidat for særhensyn til den aktuelle teksttype. Det er derfor henlagt til et særligt transformation *filter.xsl*, som importeres af *prepColx.xsl* og hvor man kan tilføje sine egne relevante filtreringer.

Bortset fra det, handler første trin om *tokenisering* (eng. *tokenization*, se [1]), dvs. hvilke tekstbidder (udtryk, tegn, eng. *tokens*) der som mindste enhed skal sammenlignes. Det foregår i vores tilfælde ved at xml-mærke hele ord med foranstillet mellemrum eller punktueringstegn, fx

```
<w> Jodlen</w><pc>,</pc>
```

Xml-mærkningen indgår ikke i sig selv i kollationeringen, det gælder også den opmærkning som findes i de originale TEI-opmærkede tekster. Men vil vi have noget af det med i kollationeringen, kan vi kamuflere det som tekst og tokenisere det på samme måde som det øvrige. Det gælder fx teksternes titel som anført i *teiHeader* eller linjebrud mellem afsnit som fx kan omdannes til dette token

```
<tag>&lt;lb/&gt;</tag>
```

I dette stykke tekstvolapyk[2] står < og > for hhv. < og >, som i et senere trin (2) kan 'aktiveres' og – *voila* – har vi TEI-elementet <lb/> (*line begin*).

Trin 2

---

1 *Prepcolx* er en xml-transformation, også sommetider omtalt som et stilark: *xslt* som står for *Xml Stylesheet language transformation*, Clark, James, *XSL Transformations (XSLT)* (1999), <https://www.w3.org/ TR/1999/REC-xslt-19991116>

2 Tingen mellem & og ; er en såkaldt 'tegnentitet' og bruges i denne sammenhæng som en art eufemisme hvor man ikke vil nævne tegnene ved deres rette navn. < og > om giver jo normalt xml-mærker, men skal her opfattes som almindelig tekst.

batch.xsl

Det andet trin består af selve udførelsen af kollationeringsprogrammet *CollateX*. Det styres af en *batchfil* som – igen – dannes udfra skakterne. Årsagen til dette yderligere omsvøb er, at det forrige trin munder ud i et antal sammenlignelige filer svarende til antallet af tekster gange antallet af skakter foretaget for hver tekst, i eksempeltilfældet 3×2 = 6 filer. Kollationsprogrammet skal altså kaldes seks gange, og resultere i seks nye filer.

Der dannes en temporær batchfil[3] som altså seks gange kalder to programmer i forlængelse af hinanden, nemlig

collateX

Dette er selvsagt den centrale del af processen.

Programmet kaldes i kommandopropten som angivet i dokumentationens afsnit 6: The Command Line Interface [1]. Her oplistes en række tilvalgskoder (eng. *options*) som styrer processen i detaljer og som her skal beskrives i udvalg, nemlig

- `-xml` : Kildeteksten, dvs. de prebehandlede filer, er xml-opmærkede (her: tokeniserede)
- `-f tei` : Resultatet formidles i (en variant af) TEI
- `-xml -xp "//w|//pc|//tag"` : Angiver en *xpath*[4] til de relevante *tokens*
- `-s \skt\nonorm.js` : Et javaScript der angiver at der *ikke* skal ske nogen normalisering af kildeteksten (hverken af store/små bogstaver eller mellemrum)

og

actang.xsl

Når kollationen er foretaget, er det tid at gendanne de xml-elementer som vi camouflerede i trin 1. Det er det eneste formål med denne transformation *activate angular brackets*.

Trin 3

colx2tei.xsl

Resultatet afleveres som en specielt collatex-variant af TEI, dvs. som læsninger (`<rdg>`) i et tekstapparat (`<app>`). Denne transformation samler de seks filer til én valid TEI-fil.

Trin 4

collatex.xsl

Vi er nu klar til at vise den færdige kollation. Den findes som en TEI-opmærket fil og gengives i et antal spalter svarende til antallet af indgående tekstkilder.

---

3 *Batch* betyder en samlet fremstilling eller behandling af et eller andet, her en kørsel af flere programmer. I *Mac* slang kaldes det et *script*.

4 *xpath* er endnu en xml-specialitet, her forenklet gengivet, `//w` betyder ord (`<w>`) over alt i teksten, `|` betyder 'eller også', *XML Path Language (XPath)* (1999), <https://www.w3.org/TR/1999/REC-xpath-19991116/>

Referencer

[1] The Interedition Development Group, *CollateX – Software for Collating Textual Sources, Documentation*, (2010-2019), <https://collatex.net/doc/>

[2] Karsten Kynde, *Automatisk generering af skakter, skaktgen.xsl vers. 02*, 2020-04-22, <http://etxt.dk/skt/dok/skaktgen.pdf>

En grundigere manual for de fire trin findes i

Karsten Kynde, *H.C. Andersens Eventyr og Historier. Den digitale manuskriptudgave. Kollationering af filer, Revideret vers. 3*, 2023-10-24, <http://etxt.dk/skt/dok/collatex3.pdf>

Den samlede programpakke kan downloades fra http://etxt.dk/skt/.
Anden dokumentation findes i

KK, *Skakter*, 2019.3.15, <http://etxt.dk/skt/dok/skakter.pdf>

KK, *Elektronisk udgivelse af Henrik Pontoppidans tre store romaner. Kodebog*, 2020-05-01, sidste afsnit, s. 11, <http://etxt.dk/skt/dok/kodebog.2020.pdf>