Bruk av komponenter i ADDML

Komponenter i additionalElements og dataObjects

Innledning

Ved innføringen av additionalElements og dataObjects i ADDML ble det gitt en mye større anledning til at brukerne selv kunne bygge opp strukturer etter eget ønske. I Arkivverket ble dette gjort for uttrekk i Noark 5-format. Imidlertid ble det ikke vedlagt noen form for maler eller eksempler som viser hvordan man bør eller kan bygge opp en slik struktur. Dermed har det i ettertid kommet ønske om å lage eksempler på hvordan dette kan gjøres.

I dette dokumentet er det gått ennå et skritt videre, siden det også kommer med anbefalinger for hvordan det skal gjøres (i Norge). I dokumentet blir det vist forskjellige grunnstrukturer (kalt komponenter) som man så kan slå sammen til større strukturer i form av byggeklosser.

Komponentene vil variere fra de mest grunnleggende til mer kompliserte. Det er dog viktig å huske på at alle komponentene kun inneholder et minimum av hva de må inneholde. Det er fortsatt fullt mulig å legge på andre elementer selv på de enkleste komponentene dersom det er behov for det og det følger på en naturlig plass. Derfor vil enkelte komponenter inneholde relativt få elementer, fordi det nærmest er en forutsetning at de skal bygges ut.

Hvordan bruke komponentene

Som nevnt over er komponentene ment som byggeklosser i en struktur. Og dermed vil man ved hjelp av komponenter som i utgangspunktet er enkle kunne bygge mer kompliserte strukturer.

Som et enkelt eksempel kan nevnes at når man skal beskrive en xml-fil vil man gjerne ha med koblingen til filens skjema, altså xsd-filen, sammen med xml-filen. Dermed kan man lage en struktur som består av komponenten fil for xml-filen, samt ennå en komponent fil for skjema-filen og knytte disse sammen. (En slik xml-fil er senere i dokumentet beskrevet som en grunnkomponent. For ytterligere beskrivelse, se under komponenten xml-fil.)

Hvorfor bruke komponenter?

Ved å bruke komponenter vil det bli enklere for alle som er vant med komponentene til å forstå strukturene. Da har man allerede en grunnleggende forståelse som det kan bygges videre på.

I tillegg vil komponenter også gjøre det enklere å teste additionalElements og dataObjects. I og med at dette da er kjente strukturer kan man legge på tester og kontroller i henhold til hva elementene i komponentene er ment å skulle inneholde. F.eks. vil det bli enklere å teste lenken til en fil i en fil-komponent når man vet hvilket element som faktisk inneholder lenken.

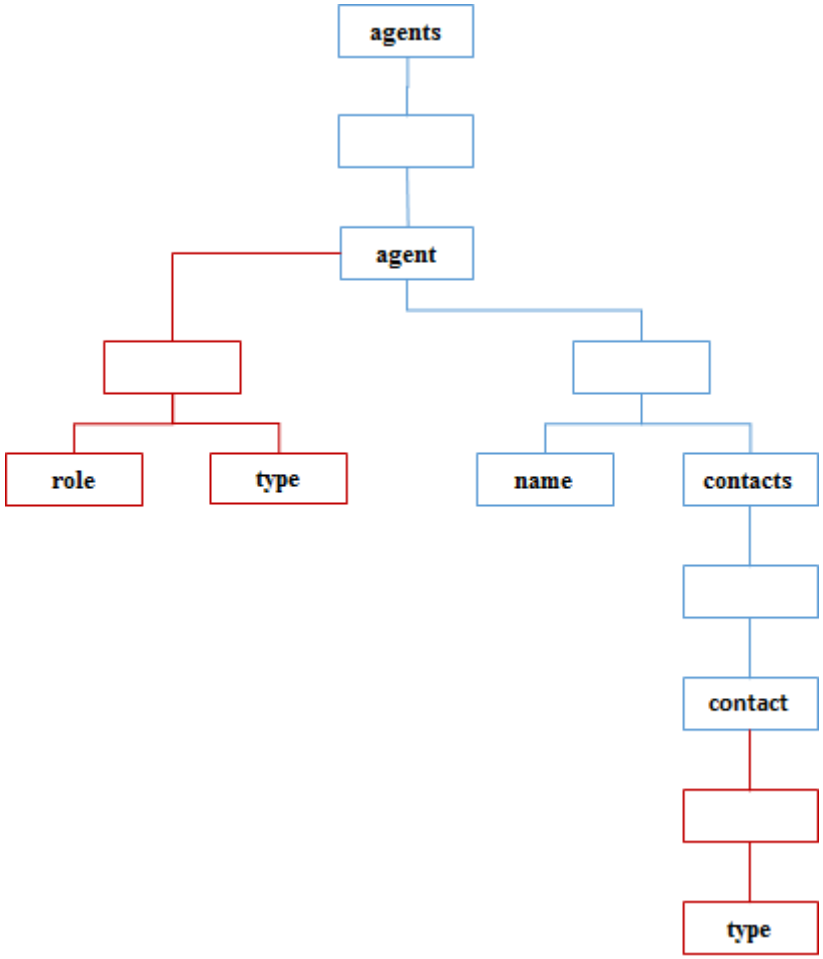
Komponenter

Dette dokumentet er bare første utgaven og vil følgelig foreløpig bare ha et begrenset antall komponenter som blir beskrevet. Målet må være at dokumentet revideres med jevne mellomrom slik at nye komponenter blir beskrevet og at antallet komponenter derved økes gradvis.

Komponenter i additionalElements

I additionalElements vil det først og fremst være behov for komponenter i sammenheng med informasjon om aktører, det opprinnelige systemet og uttrekket.

Aktør



Figur A. Grunnoversikt over aktør.

Grunninformasjonen for en aktør vil være:

- role:
  aktørens rolle i sammenheng med materialet.
  Følgende verdier er gyldige verdier:
  - recordCreator – i betydningen arkivskaper
  - deliver – i betydningen avgiver, dvs avleverende institusjon eller person

    - producer – i betydningen den som har produsert selve arkivuttrekket
    - owner – i betydningen eier av selve arkivmaterialet
    - archive – i betydningen depot som mottar arkivuttrekket

- type (til aktør):
  hva slags type aktør det er snakk om.
  Følgende verdier er gyldige verdier:
    - individual – om aktøren er en person
    - institution – om aktøren er en institusjon eller et firma
    - system – om aktøren er en applikasjon

- name:
  aktørens navn

- contact:
  kontaktinformasjon til aktøren

- type (til contact):
  hva slags type informasjon som oppgis i contact.
  Følgende verdier er gyldige verdier (listen kan utvides):
    - address
    - email
    - telephone
    - mobile

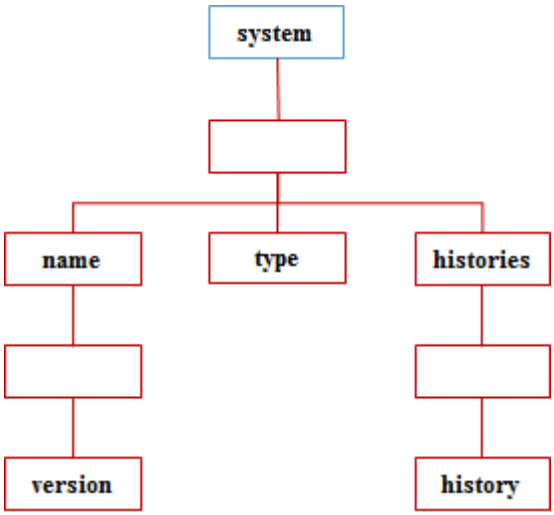
Eksempel:

```
<additionalElements>
  <additionalElement name=”agents”>
    <additionalElements>
      <additionalElement name=”agent”>
        <properties>
          <property name=”role”>
            <value>recordCreator</value>
          </property>
          <property name=”type”>
            <value>institution</value>
          </property>
        </properties>
```

```
            <additionalElements>
              <additionalElement name=”name”>
                <value>Aktørens navn</value>
              </additionalElement>
              <additionalElement name=”contacts”>
                <additionalElements>
                  <additionalElement name=”contact”>
                    <value>Aktørens adresse</value>
                    <properties>
                      <property name=”type”>
                        <value>address</value>
                      </property>
                    </properties>
                  </additionalElement>
                  <additionalElement name=”contact”>
                    <value>Aktørens e-post addresse</value>
                    <properties>
                      <property name=”type”>
                        <value>email</value>
                      </property>
                    </properties>
                  </additionalElement>
                </additionalElements>
              </additionalElement>
            </additionalElements>
          </additionalElement>
          <additionalElement name=”agent”>
            <properties>
              <property name=”role”>
                <value>deliver</value>
              </property>
              <property name=”type”>
                <value>institution</value>
              </property>
            </properties>
            <additionalElements>
              ...
            </additionalElements>
          </additionalElement>
            ...
        </additionalElements>
      </additionalElement>
    </additionalElements>
```

System

Under system er det tenkt å beskrive det opprinnelige systemet et arkivuttrekk ble tatt fra.



Figur B. Grunnoversikt over system.

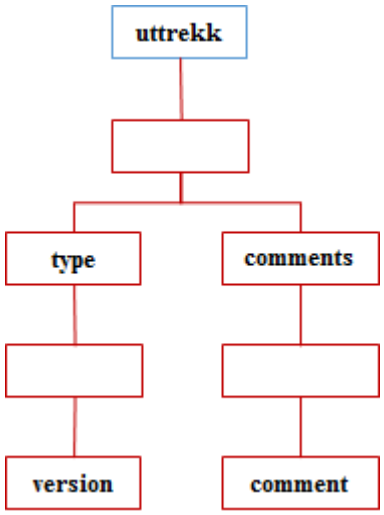
Eksempel:

```
<additionalElements>
  <additionalElement name="system">
    <properties>
      <property name="name">
        <value>Det opprinnelige systemets navn</value>
          <properties>
            <property name="version">
              <value>Versjon av system</value>
            <property>
          <properties>
      <property>
      <property name="type">
        <value>Noark 5</value>
      <property>
      <property name="histories">
          <properties>
            <property name="history">
              <value>Beskrivelse av historikk, enkelthendelse</value>
            <property>
            <property name="history">
              <value>Beskrivelse av historikk, enkelthendelse</value>
            <property>
            ...
```

```
            <properties>
        <property>
      </properties>
   </additionalElement>
</additionalElements>
```

Uttrekk

Under uttrekk beskrives det aktuelle uttrekket.



Figur C. Grunnoversikt over uttrekk.

Eksempel:

```
<additionalElements>
   <additionalElement name=”uttrekk”>
      <properties>
         <property name=”type”>
            <value>Noark 5</value>
               <properties>
                  <property name=”version”>
                     <value>Versjon av Noark 5</value>
                  <property>
               <properties>
         <property>
         <property name=”comments”>
               <properties>
                  <property name=”comment”>
                     <value>Kommentar</value>
```

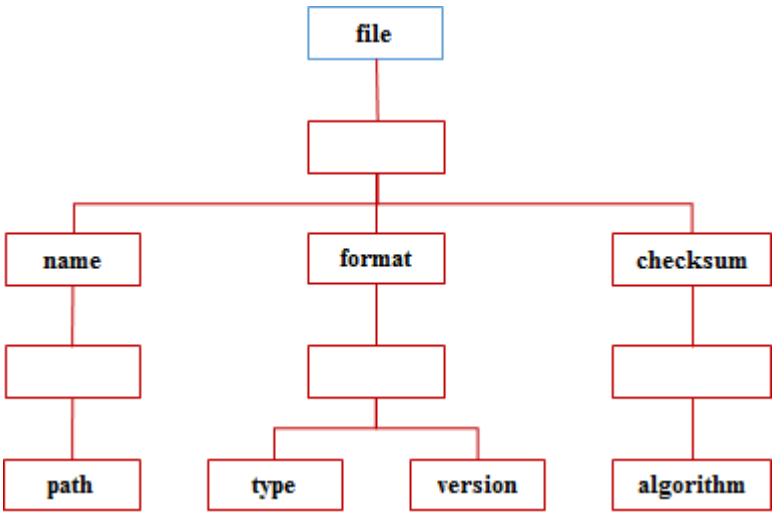
```
            <property>
            <property name="comment">
              <value>Kommentar</value>
            <property>
            ...
          <properties>
        <property>
      </properties>
    </additionalElement>
</additionalElements>
```

Her er det mulig å bygge komponenten videre ut dersom man f.eks. ønsker å vite hvem som har gitt kommentaren, ved å lage en *property* til hver enkelt kommentar med navn og dato.

Komponenter i dataObjects

I motsetning til additionalElements hvor det er et begrenset antall komponenter, vil det for dataObjects kunne være et uendelig antall komponenter og varianter. Dermed vil det først og fremst her være behov for å utvide med nye komponenter over tid. I denne omgang er det tenkt på følgende strukturer: fil, xml-fil og katalog.

Fil



Figur D. Grunnbeskrivelse av en fil.

Som grunninformasjon som alltid skal være tilstede for en hvilken som helst fil, gjelder følgende:

- Filens navn inkludert relativ sti fra utgangspunktet.

- Hva slags type format denne filen er. Og herunder følger da informasjon om typen og hvilken versjon av denne. (Typer er f.eks. PDF/A, TIFF, JPG, osv)
- En sjekksum for filen. Og herunder følger da selve sjekksummen og hvilken algoritme som er benyttet for å beregne sjekksummen. Pr. i dag er det SHA-256 som skal benyttes som algoritme.

Alle disse informasjonene registreres som egenskaper til filen.

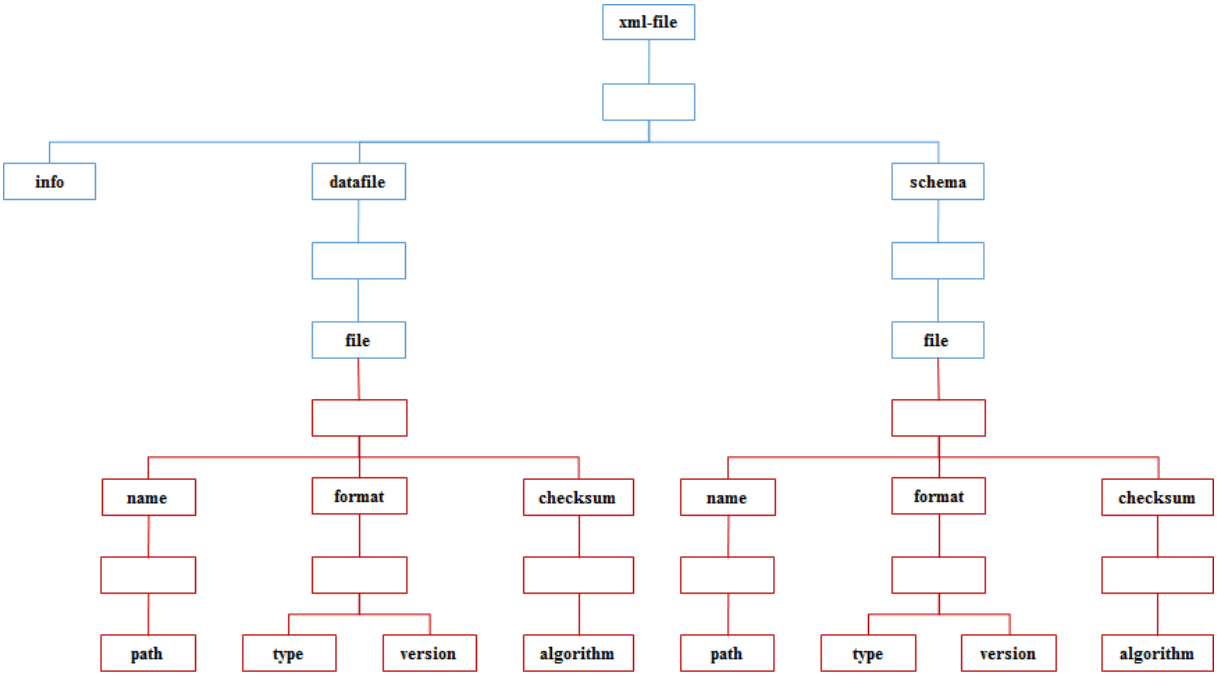
Også andre informasjoner kan det i enkelte tilfeller være behov for, f.eks. Krypteringsinformasjon.

Eksempel:

```
<dataObjects>
  <dataObject name=”file”>
    <properties>
      <property name=”name”>
        <value>Selve filnavnet</value>
        <properties>
          <property name=”path”>
            <value>Relativ sti til filen</value>
          <property>
        </properties>
      <property>
      <property name=”format”>
        <properties>
          <property name=”type”>
            <value>Filens type format</value>
          <property>
          <property name=”version”>
            <value>Versjon av formatet</value>
          <property>
        </properties>
      <property>
      <property name=”checksum”>
        <value>Selve sjekksummen</value>
        <properties>
          <property name=”algorithm”>
            <value>Sjekksummens algoritme</value>
          <property>
        </properties>
      <property>
    </properties>
  </dataObject>
</dataObjects>
```

xml-fil

En xml-fil følger naturlig nok samme prinsipper som en hvilken som helst annen fil. I tillegg skal den knyttes opp mot sin beskrivelse, enten i form av et skjema (xsd) eller flere, eller en beskrivelse (dtd). Og disse igjen er også filer. Slik at det spesielle her er den nevnte koplingen mellom to eller flere filer.



Figur E. Grunnoversikt over en xml-fil med en beskrivende fil (skjema).

I tillegg vil det også være naturlig med en info-komponent i de fleste tilfellene.

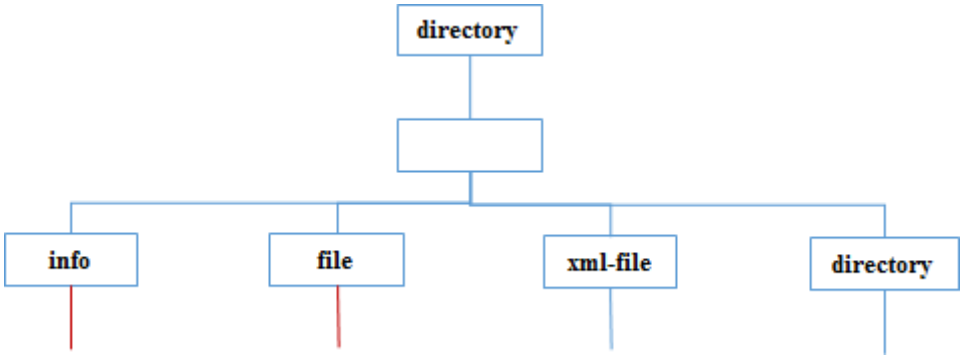
Eksempel:

```
<dataObjects>
  <dataObject name=”xml-file”>
    <dataObjects>
      <dataObject name=”info”>
        ...
      </dataObject>
      <dataObject name=”datafile”>
        <dataObjects>
          <dataObject name=”file”>
            ...
          </dataObject>
        </dataObjects>
```

```
        </dataObject>
        <dataObject name=”schema”>
          <dataObjects>
            <dataObject name=”file”>
              ...
            </dataObject>
          </dataObjects>
        </dataObject>
      </dataObjects>
    </dataObject>
</dataObjects>
```

Katalog

En katalog som komponent er tenkt benyttet for en samling av filer i en katalogstruktur.



Figur F. Grunnoversikt over katalog.

Eksempel:

```
<dataObjects>
  <dataObject name=”directory”>
    <dataObjects>
      <dataObject name=”info”>
        ...
      </dataObject>
      <dataObject name=”file”>
        ...
      </dataObject>
      <dataObject name=”xml-file”>
        ...
      </dataObject>
      <dataObject name=”directory”>
```

```
        ...
      </dataObject>
      ...
    <dataObjects>
  </dataObject>
</dataObjects>
```

Info

Figur G. Grunnoversikt over info.

Eksempel:

```
<dataObjects>
  <dataObject name=”info”>
    <properties>
      <property name=”filetype”>
        <properties>
          <property name=”name”>
            <value>PDF/A</value>
          </property>
          <property name=”version”>
            <value>1A</value>
          </property>
        </properties>
      </property>
      ...
    </properties>
  </dataObject>
</dataObjects>
```

NB! Eksemplet er kun et eksempel på hvordan man kan define egenskaper under info. Den egenskapen som er vist hører egentlig hjemme under Fil som format.