

# **Redigerbare dokumentformater**

## **Anbefaling om sammensetning**

Lars Marius Garshol, Bouvet ASA

# Innhold

Side

1	Sammendrag av rapporten.....	5
2	Bakgrunn.....	6
2.1	Avgrensninger og terminologi.....	6
2.2	Metodikk.....	6
2.3	En eller to standarder?.....	7
2.4	Internasjonale hensyn.....	7
2.5	Takk til.....	9
3	Vurdering av standardene.....	10
3.1	Om å vurdere en spesifikasjon.....	10
3.2	ODF.....	11
3.2.1	Problemer med standarden.....	11
3.2.1.1	Versjonering.....	12
3.2.2	Grunnleggende formatering.....	12
3.2.3	Endringssporing.....	12
3.2.4	Kommentarer.....	13
3.2.5	Formler i regneark.....	13
3.2.6	Makroer i regneark.....	14
3.2.7	Skjemaer.....	14
3.3	OOXML.....	15
3.3.1	Problemer med spesifikasjonen.....	16
3.3.1.1	Datoer i regneark.....	16
3.3.1.2	Selvmotsigelser.....	16
3.3.1.3	Versjonering.....	17
3.3.1.4	Diverse.....	17
3.3.2	Endringssporing.....	18
3.3.3	Kommentarer.....	18
3.3.4	Formler i regneark.....	18
3.3.5	Makroer i regneark.....	18
3.3.6	Skjemaer.....	19
3.4	Interoperabilitet.....	19
3.4.1	Grunnleggende formatering.....	19
3.4.2	Endringssporing og kommentarer.....	19
3.4.3	Formler i regneark.....	19
3.5	Oppsummering.....	19
4	Implementasjoner.....	21
4.1	OOXML.....	21
4.1.1	Grunnleggende formatering.....	23
4.1.2	Endringssporing.....	24

4.1.3	Kommentarer.....	24
4.1.4	Formler i regneark.....	24
4.1.5	Skjemaer.....	25
4.2	ODF.....	25
4.2.1	1 eller 4 verktøy?.....	26
4.2.2	Grunnleggende formatering.....	27
4.2.3	Endringssporing.....	28
4.2.4	Kommentarer.....	28
4.2.5	Formler i regneark.....	28
4.2.6	Skjemaer.....	29
4.3	Oppsummering.....	29
5	Anvendelsesområdene.....	29
5.1	Publisering på offentlige nettsider.....	30
5.1.1	Scenario #1: Skjemaer på nett.....	30
5.1.2	Scenario #2: Distribusjon av maler.....	30
5.1.3	Scenario #3: Standardkontrakter.....	31
5.1.4	Anbefalinger.....	31
5.2	Utteksling offentlig/privat via epost.....	32
5.2.1	Scenario #4: Utfylling av skjema.....	32
5.2.2	Scenario #5: Utarbeiding av kontrakt.....	32
5.2.3	Anbefalinger.....	32
5.3	Utteksling offentlig/offentlig via epost.....	33
5.3.1	Scenario #6: Samarbeidende skriving på tvers av etat.....	33
5.3.2	Scenario #7: Uttveksling av budsjettdata.....	33
5.3.3	Anbefalinger.....	33
5.4	Produksjonsformat.....	34
6	Ytterligere vurderinger.....	34
6.1	Fremtidig utvikling.....	34
6.2	Konsekvenser av anbefalingene.....	35
6.3	Om redigerbare dokumenter og standarder.....	35

## **OBS!**

Dette er ikke den endelige versjonen av rapporten, men kun versjon 0.9. Vi regner med å publisere den endelige versjonen innen utgangen av mai.

# 1 Sammendrag av rapporten

Denne rapporten er skrevet på oppdrag fra Difi, for å gi anbefalinger om hvilke av de to standardene OOXML og ODF som bør være med i neste versjon av referansekatalogen for IT-standarder [Referansekatalogen]. Helt konkret er vi blitt bedt om å komme med anbefalinger for to bestemte versjoner av standardene: ECMA-376:2006 for OOXML og ODF 1.1 for ODF. Videre er vi også kun blitt bedt om å komme med anbefalinger for følgende fire anvendelsesområder for redigerbare dokumenter:

- Publisering av tekstdokumenter på offentlige nettsider (revisjon),
- Utveksling av dokumenter som vedlegg til e-post mellom offentlig virksomhet og innbyggere/næringsliv (revisjon),
- Utveksling av dokumenter som vedlegg til e-post mellom offentlig virksomheter (nytt), og
- Produksjonsformater i fagsystemer (nytt).

Av tidshensyn er det siste anvendelsesområdet ikke vurdert i denne rapporten.

For å identifisere hvilken funksjonalitet offentlig sektor har behov for innen disse anvendelsesområdene har vi identifisert en rekke scenarier for praktisk bruk innen hvert anvendelsesområde (beskrevet i kapittel 5 side 29) og analysert disse for å se hvilken funksjonalitet som trengs innen hvert enkelt område.

Vi har så vurdert kvaliteten på beskrivelsen i standardene av den identifiserte funksjonaliteten, og testet utveksling mellom ulike kontorpakker av dokumenter som bruker disse funksjonene. Metodikken er beskrevet i mer detalj i kapittel 2.2 side 6.

Hvert enkelt anvendelsesområde er vurdert separat, men i og med at konklusjonene for de ulike områdene er de samme kan resultatene av undersøkelsene kort sammenfattes som følger:

- ECMA-376:2006 (OOXML) beskriver stort sett den nødvendige funksjonaliteten med tilstrekkelig presisjon til at standarden kan implementeres korrekt, men spesifikasjonen er stor, tung å lese, inneholder en del teknisk uheldige valg og er dertil belemret med en lang rekke småfeil. I tillegg til dette kommer at formatet i dag ikke er stabilt, på grunn av alle endringene som gjøres i formatet i ISO-prosessen. En mer detaljert redegjørelse finnes i kapittel 3.3 side 15.
- Enkel testing av utveksling av OOXML-dokumenter mellom utvalgte kontorpakker viser at dette fungerer svært dårlig. For mange typer funksjonalitet er det ingen andre verktøy enn Microsoft Office som kan åpne filer fra Office korrekt og lagre tilbake til OOXML. Testresultatene finnes i kapittel 4.1 side 21.
- ODF 1.1 beskriver ikke all den funksjonaliteten vi har identifisert behov for, og der funksjonaliteten er beskrevet er i flere tilfeller presisjonen på beskrivelsen for dårlig til at man kan være trygg på at den blir implementert korrekt. Det er også mange mindre feil i spesifikasjonen. Det blir gjort et stort antall endringer i formatet i versjon 1.2, slik at man kunne også ønske seg større stabilitet i formatet. En mer detaljert redegjørelse finnes i kapittel 3.2 side 11.
- Tilsvarende testing av utveksling av ODF-dokumenter viser at utveksling stort sett kun fungerer mellom OpenOffice.org og IBM Lotus Symphony, som er to produkter basert på samme opprinnelige kodebase. Testresultatene finnes i kapittel 4.2 side 25.

**Vår konklusjon er** at vi ikke finner det faglig forsvarlig å anbefale noen av disse standardene, fordi kvaliteten på begge spesifikasjonene er lav og fordi utveksling mellom verktøyene ikke fungerer. I praksis vil man ved valg av OOXML være tvunget til å bruke kun ett verktøy, mens ved valg av ODF vil man ha et valg mellom to svært like verktøy.

Det ser dog ut til at situasjonen vil bedre seg i fremtiden, slik at vi anbefaler at begge standardene settes under observasjon. Når spesifikasjonene har blitt utbedret og verktøyene har fått tid til å modnes kan det godt være at det vil bli mulig å anbefale enten en av standardene, eller begge.

Dersom standardiseringsrådet skulle følge disse anbefalingene i neste versjon av referansekatalogen vil konsekvensen mest sannsynlig være at Microsoft Office fortsetter å ha tilnærmet monopol i norsk offentlig sektor, og at den tette bindingen til dette produktet i fagsystemer fortsetter. Se kapittel 6.2 side 35 for en mer inngående diskusjon av dette.

## 2 Bakgrunn

Dette kapittelet gir litt bakgrunn for de vurderinger som gjøres i resten av rapporten.

### 2.1 Avgrensninger og terminologi

Denne rapporten vurderer to spesifikke versjoner av to dokumentformater. Formatene omtales i rapporten som henholdsvis ODF (Open Document Format) og OOXML (Office Open XML). Disse finnes i en rekke forskjellige versjoner, definert i forskjellige spesifikasjoner. Disse spesifikasjonene vil bli omtalt ved sine navn der det konkret er snakk om spesifikasjonene og ikke formatene.

Både ODF og OOXML er formater som omfatter mer enn bare tekstdokumenter. Begge har også støtte for regneark og presentasjoner. I denne rapporten har vi fokusert på tekstdokumenter, men også tatt hensyn til regneark der scenariene viser at det er naturlig. Presentasjoner har ikke blitt vurdert.

Denne rapporten begrenser seg til en rent teknisk vurdering av kvaliteten på de to standardene og deres implementasjoner for offentlig sektors behov innen de tre vurderte anvendelsesområdene. For andre anvendelsesområder er det mulig at man ville kommet til andre konklusjoner.

Politiske og markedsmessige hensyn er ikke tatt med i vurderingene.

Rapporten inneholder en rekke tabeller som oppsummerer resultatene av vurderinger og tester. Disse bruker en uformell karakterskala som kan kort oppsummeres slik:

- Ikke spesifisert: vil si at funksjonaliteten ikke er beskrevet i det hele tatt, og dermed strengt tatt ikke er standardisert.
- Dårlig: vil si at på dette området er standarden ikke bare under det som kan regnes som akseptabelt, men langt under.
- Ikke godt nok: vil si at på dette området er standarden under det som kan regnes som akseptabelt.
- Godt nok: vil si at på dette området er standarden akkurat på rett side av grensen for det som kan regnes som akseptabelt.
- Bra: vil si at på dette området er standarden ikke bare god nok, men faktisk ganske bra, selv om det fortsatt er noe å utsette på den.

Det burde kanskje vært en karakter til i skalaen over "Bra" for områder der en standard kommer meget godt ut og det egentlig ikke er noe å utsette på den, men i praksis har det ikke vært noe behov for en slik karakter.

### 2.2 Metodikk

Filosofien bak denne rapporten er at for at man skal kunne anbefale en standard til bruk innen et bestemt anvendelsesområde i norsk forvaltning må standarden støttes av flere verktøy som brukerne kan velge mellom. Videre må disse verktøyene være av en viss kvalitet, og utveksling av data mellom verktøyene må fungere nogenlunde pålitelig.

Sagt på en annen måte må kravet være at det må være et fungerende økosystem av implementasjoner rundt standarden.

For å verifisere om dette er tilfellet har vi valgt å basere oss på to kontroller:

- Vurdering av beskrivelsen av relevant funksjonalitet i standardene, for å se om denne dekker behovene og om beskrivelsen er av tilstrekkelig kvalitet til at man kan forvente interoperabilitet mellom implementasjonene. Vurderingene finnes i kapittel 3 side 10.
- Enkle tester av relevant funksjonalitet i verktøyene for å se om interoperabiliteten er tilstede i praksis. Resultatene av testene finnes i kapittel 4 side 21

For å identifisere relevant funksjonalitet innen hvert anvendelsesområde har vi valgt å trekke opp scenarier for bruk, og så analysere disse for å finne frem til funksjonaliteten. Scenariene er kvalitetssikret gjennom diskusjoner med aktører i offentlig sektor og mot fritt tilgjengelig informasjon på nett.

## 2.3 En eller to standarder?

Formålet med denne rapporten er å anbefale *sammensetninger* av de to standardene for de ulike bruksområdene, og å grunngi anbefalingene. En sammensetning vil i dette tilfellet si at man for hver av de to standardene sier at denne er enten obligatorisk, anbefalt, ikke anbefalt, under utfasing eller at man ikke sier noe om standarder for redigerbare dokumenter i det hele tatt. I og med at versjon 1.0 og 2.0 av referansekatalogen har hatt føringer for redigerbare dokumenter ville det siste være et ganske drastisk skritt, og ville vel i praksis måtte tolkes som at man ikke finner standarder for dette som kan anbefales.

Med to standarder og fem alternativer for hver standard har man i teorien ti muligheter for hvert bruksområde. Det er dog ikke helt klart hvordan alle disse mulighetene skal tolkes. Noen er enkle, som f.eks at man gjør bare en standard obligatorisk/anbefalt, eller at man ikke sier noe om noen av standardene.

Men hva om man gjør begge standardene obligatoriske? Hva betyr det? Dette kan gjøres på forskjellige måter. Man kan f.eks skille mellom ulike brukssituasjoner<sup>1</sup> for standardene og gi forskjellige anbefalinger basert på det. For eksempel kan man skille mellom hva som skal støttes ved mottak av dokumenter og hva som skal brukes ved publisering/utsendelse av dokumenter.

I praksis er det vanskelig å se noe grunnlag for å si at de to standardene egner seg for ulike brukssituasjoner innen et anvendelsesområde, slik at å skille dem på et dette grunnlaget blir svært vanskelig.

Dersom alle dokumenter skal gjøres tilgjengelige i begge formater vil dette i så fall medføre betydelige ulemper for de som publiserer dokumentene, som da vil måtte legge ut en ekstra versjon av alle dokumenter og også sørge for å ha verktøy som støtter begge standardene. For mottakerne vil dette øke sjansen for at de har et verktøy som kan lese en av versjonene, men det vil også kreve at mottakerne er i stand til å forstå forskjellen på de to formatene. Svært mange vil sannsynligvis bli forvirret av å bli tvunget til å gjøre et slikt valg.

Dersom mottakere i offentlig sektor skal pålegges å kunne motta dokumenter i begge formater vil dette igjen kreve at man installerer programvare som kan lese begge formater (enten ett program eller to). Hvor store ulempene med dette vil være er ikke klart.

Dersom man gjør en standard obligatorisk og den andre anbefalt betyr vel det i praksis at man krever at alle dokumenter foreligger i det obligatoriske formatet, men at man anbefaler at det også gjøres tilgjengelig i det anbefalte formatet. Dette blir omtrent som når begge standardene er obligatorisk, med den forskjell at offentlige etater gis anledning til å publisere kun én fil. Man må anta at i praksis vil de aller fleste kun legge ut en fil dersom dette skulle bli anbefalingen.

I praksis betyr dette at dersom man skulle gjøre to standarder anbefalt/obligatorisk vil dette mest sannsynlig ha ikke ubetydelige kostnader. Det vil derfor måtte være gode grunner for en slik anbefaling, og det fremstår i utgangspunktet som bedre å velge en standard enn å velge to.

## 2.4 Internasjonale hensyn

Et viktig hensyn når Norge skal ta en beslutning om hvilke formater som skal brukes i norsk forvaltning er hva andre land har bestemt seg for, slik at man unngår en situasjon der Norge legger seg på en helt annen linje enn resten av verden. Særlig viktig er valg gjort av andre land i Norden, og hva EU bestemmer seg for, men det bør også tas hensyn til andre aktører rundt om i verden.

Generelt ser det ikke ut til å være spesielt vanlig internasjonalt at det settes krav til hvilke dokumentformater som brukes i offentlig forvaltning, slik at for mange land er det ikke mulig å finne noen klare føringer i det hele tatt.

Ett eksempel på dette er Sverige, der valg av dokumentformat var oppe til het debatt i 2007, uten at det ble tatt noen beslutning. I dag er temaet mer eller mindre forsvunnet fra den offentlige debatt, og den eneste

---

<sup>1</sup> Dette må ikke forveksles med anvendelsesområde. Vi snakker her om hva som i teorien kan gjøres *innen* et bestemt anvendelsesområde.

formelle føring som ligger fast er det svenske Riksarkivets pålegg om at dokumenter skal arkiveres i PDF/A-format.

Tilstanden i EU er omtrent tilsvarende. Det har vært noen rapporter og høringer om problemstillingen (som f.eks [Valoris2003] og [TAC2004]), men det er per i dag ingen konkret anbefaling eller krav fra EU sentralt om hvilke formater som skal eller bør brukes.

Det samme gjelder Japan, der det ikke eksisterer noen offentlige krav til bruk av dokumentformater. Kommunestyret i Aizuwakamatsu (i Fukushima-prefekturet) promoterer ODF ivrig, men ellers har det offentlige ikke tatt noen offisielle avgjørelser rundt formatvalg. Situasjonen i USA er mye den samme: den føderale regjering og de fleste lokale administrasjoner har ikke satt noen krav. Unntaket er staten Massachusetts som opprinnelig krevde ODF, men tillater nå også OOXML [WikiODFAdoption]. Situasjonen er tilsvarende også i Sør-Korea [WikiODFAdoption], til tross for at dette er et land der myndighetene involverer seg svært aktivt i IT-utvikling og -forskning.

Kina er i denne sammenhengen et særtilfelle. Her har man valgt å utvikle UOF (Uniform Office Format), et helt nytt format med funksjonalitet spesielt tilrettelagt for kinesiske behov [OpenMalaysia2007b]. Denne ble vedtatt som kinesisk nasjonal standard GB/T 20916-2007 i 2007 [WikiUOF], hvilket vil si at den er utgitt av det kinesiske nasjonale standardiseringsorganet som en offisiell standard, men ikke nødvendigvis at det er noe formelt krav om at den må brukes. Hvor utbredt denne standarden faktisk er i Kina i dag er uklart. Noen hevder at den er dominerende i dag (se [OpenMalaysia2007b]), mens mer pålitelige kilder ser ut til å antyde at den er mindre utbredt, men promoterer aktivt (se [Smith2008] og [Xinhua2009]).

I Danmark har regjeringen involvert seg mer direkte i valg av formater, og det ble i 2007 utarbeidet en liste over standarder som inneholdt både ODF og OOXML. Dette ble så tatt opp til ny diskusjon i 2009, og 29. januar 2010 ble det inngått en avtale mellom IT-ordførerne til en rekke partier i det danske Folketinget. Denne vedtar at kun formater på en offisiell standardliste kan brukes, og dokumenter som ikke skal redigeres skal publiseres i PDF/A. For at formatene skal kunne komme på listen må de fylle visse generelle kriterier, anbefales av en ekspertgruppe, og utvelges av vitenskapsministeren [VTU2010]. Listen er i skrivende stund ikke publisert, men det er en generell forventning om at ODF 1.0 og OOXML Strict vil komme på listen.

I Finland har det så langt ikke vært satt noen krav til bruk av standarder i offentlig forvaltning, men regjeringen arbeider med en ny lov som vil gi Finansministeriet autoritet til å kreve bruk av bestemte IT-standarder i offentlig forvaltning. Hvilke områder man i så fall vil sette krav for er ikke klart.

I Tyskland har man også en egen oversikt over IT-standarder som skal brukes i det offentlige, kalt SAGA. Denne er svært detaljert og tar for seg en lang rekke standarder innen forskjellige anvendelsesområder. Foreløpig siste versjon er SAGA 4.0 fra mars 2008, som anbefaler ODF, mens OOXML er under observasjon [SAGA 4.0]. SAGA 5.0 er under utarbeidelse, men er ikke ferdig.

Frankrike har en lignende oversikt, kalt RGI, der siste versjon er 1.0 fra mai 2009. Denne har oppført både ODF og OOXML som under observasjon [RGI 1.0].

Det er også en del andre europeiske land der det offentlige har satt krav til bruk av åpne dokumentformater. Belgia, Nederland og Slovakia, som krever bruk av ODF, samt Sveits, ser ut til å være de viktigste av disse. Utenfor Europa har Uruguay anbefalt og Malaysia og Venezuela krevd bruk av ODF [WikiODFAdoption].

Land	ODF	OOXML	Kommentar	Kilde
Danmark	Tillatt	Tillatt	Antagelser	Jesper Lund Stocholm
Sverige	Ingen krav	Ingen krav		Stefan Görling
Finland	Ingen krav	Ingen krav	Under endring	[WikiODFAdoption] og Tommi Kartaavi
Tyskland	Anbefalt	Under observasjon		[SAGA 4.0]
Storbritannia	Ingen krav	Ingen krav		



Land	ODF	OOXML	Kommentar	Kilde
Frankrike	Under observasjon	Under observasjon		[RGI 1.0]
EU	Ingen krav	Ingen krav		[WikiODFAdoption]
Japan	Ingen krav	Ingen krav		Murata-san
Sør-Korea	Ingen krav	Ingen krav		[WikiODFAdoption] og Jaeho Lee
USA	Ingen krav	Ingen krav		[WikiODFAdoption]

Kort oppsummert kan man vel si at det er kun et fåtall land der det offentlige har satt sammen en standardkatalog i det hele tatt. Av de mest sentrale landene sett med norske øyne (se tabell) ser det ut til at man stort sett er svært forsiktig med å sette strenge krav til standarder for redigerbare dokumenter. Det landet som er strengest er vel antagelig Tyskland, som altså anbefaler ODF og har OOXML under observasjon.

Situasjonen i Norge i dag er at ODF er påkrevd og OOXML under observasjon for ett anvendelsesområde, samt at det samme skal innføres for nok et anvendelsesområde fra 1. januar 2011. Dette er litt uvanlig internasjonalt. De eneste eksemplene vi har klart å finne på land som har valgt en lignende linje er Nederland, Belgia, Slovakia, Malaysia og Venezuela. Med Norge blir dette altså 6 land av ca 200, som er en nokså liten minoritet.

Det må i parentes bemerkes at informasjonen om landene som ikke står i tabellen over kommer fra [WikiODFAdoption], som er en kilde der vi har funnet en del feil, slik at den virkelige situasjonen kan være noe annerledes. Det virker dog usannsynlig at det skal være betydelige avvik.

## 2.5 Takk til

I arbeidet med denne rapporten har mange hjulpet til bak kulissene på ulike måter. En hjertelig takk til:

- Stian Danenbarger, Bouvet, for svar på spørsmål om scenarier.
- Doug Mahugh, Microsoft, for svar på spørsmål versjoner av OOXML og identifisering av OOXML-versjon i dokumenter.
- Alex Brown, Griffin Brown, for svar på spørsmål om identifisering av ODF-versjon og informasjon om selvmotsigelser i ODF og OOXML.
- Knut-Erik Gudim, Software Innovation, for informasjon om bruk av OOXML og ODF i fagsystemer.
- Gareth Horton, Datawatch, for informasjon om implementasjon av OOXML og ODF fagsystemer.
- Jesper Lund Stocholm, Ciber, for informasjon om implementasjon av OOXML og ODF fagsystemer, samt informasjon om status på OOXML og ODF i dansk offentlig sektor.
- Stefan Görling for informasjon om status på OOXML og ODF i svensk offentlig sektor.
- MURATA Makoto (ETTERNAVN Fornavn) for informasjon om status på OOXML og ODF i japansk offentlig sektor.
- Jaeho Lee, universitetet i Seoul, for informasjon om status på OOXML og ODF i koreansk offentlig sektor.
- Tommi Kartavi, finske kommuneforbundet (Suomen Kuntaliitto), for informasjon om status på OOXML og ODF i finsk offentlig sektor.
- Morten Haug Frøyen, Utdanningsdirektoratet, for tilbakemelding på scenariene og behov i offentlig sektor.
- Ellen Weberg, Utdanningsdirektoratet, for tilbakemelding på scenariene og behov i offentlig sektor.
- Dag Sverre Henriksen, Utdanningsdirektoratet, for tilbakemelding på scenariene og behov i offentlig sektor.
- Ola Thoen, Departementenes Servicesenter, for tilbakemelding på scenariene og behov i offentlig sektor.
- Torgeir Strypet, Ergo Group, for informasjon om bruk av OOXML og ODF i offentlig sektor generelt og spesielt i fagsystemer.
- Morten Græsby, Bouvet, for tilbakemelding på metodikk og tilnærming, samt tilbakemelding på nest siste versjon.

— Geir Ove Grønmo, Bouvet, for tilbakemelding på metodikk og tilnærming.

### 3 Vurdering av standardene

I dette kapittelet vurderer vi den tekniske kvaliteten på de forskjellige spesifikasjonene. Helt konkret har vi sett på hvordan funksjonalitet som scenariene har identifisert som viktig er spesifisert i standardene for å vurdere i hvilken grad man kan anta at verktøyene har implementert dette på samme måte. Dette er avgjørende for hvorvidt manglende interoperabilitet mellom verktøy kan antas å bli bedre i fremtiden.

#### 3.1 Om å vurdere en spesifikasjon

En spesifikasjon kan vurderes på et utall forskjellige måter avhengig av ståstedet til den som skal gjøre vurderingen. For denne rapporten er det vesentlige i hvilken grad spesifikasjonen gir grunnlag for interoperabilitet mellom ulike implementasjoner av standarden. Alle elementer ved en spesifikasjon som kan beskrives som svakheter er dermed kun relevante i den grad de svekker muligheten for interoperabilitet mellom implementasjonene, eller i den grad de vanskeliggjør nye implementasjoner og på den måten reduserer det sannsynlige antallet implementasjoner.

En svakhet kan være **manglende funksjonalitet**. For eksempel spesifiserer ikke OOXML databasefunksjonalitet som den man finner i Microsoft Access og FileMaker. Det betyr at implementasjoner av OOXML som velger å legge inn databasefunksjonalitet ikke vil ha noen mulighet til å oppnå interoperabilitet via OOXML<sup>2</sup>, selv om de vil kunne oppnå det ved å knytte seg til andre standarder eller å kommunisere seg imellom. Manglende funksjonalitet fører med andre ord direkte til manglende interoperabilitet om ikke implementasjonene aktivt kompenserer for svakheten i spesifikasjonen.

En annen svakhet er at funksjonaliteten er der, men at den er **utilstrekkelig spesifisert**. Dette betyr at man gjør det klart at en bestemt funksjonalitet er med i standarden, men at beskrivelsen av denne ikke er tilstrekkelig presis til at ulike implementasjoner vil velge samme løsning uavhengig av hverandre. Et klassisk eksempel er når ODF 1.1 sier at passord skal lagres som et hash uten å si hvordan man skal beregne hashet. Dersom ulike implementasjoner gjør dette likt uten å kommunisere seg imellom vil dette være ren flaks. Igjen er det mulig for implementasjoner å kompensere for svakheten.

En svært vanlig svakhet er at standarden er **vanskelig å forstå**, slik at selv om funksjonaliteten er spesifisert med tilstrekkelig presisjon er det svært krevende for utviklere å forstå spesifikasjonen. Ett eksempel kan være RELAX-NG, som er formulert helt og holdent ved hjelp av en matematisk formalisme basert på logikk. Denne er svært presis, men samtidig meget krevende å forstå. Andre problemer kan være at spesifikasjonen er dårlig organisert, f.eks ved at informasjon om ett tema er spredd rundt i spesifikasjonen, eller at spesifikasjonen rett og slett er veldig stor og omfangsrik. Igjen er dette problemer som kan kompenseres for.

I teorien skal ikke en standard inneholde **tekniske feil**, men realiteten er at slikt forekommer. Det kan dreie seg om alt fra syntaksfeil i formelle språk som brukes for å definere standarden (som f.eks XML Schema), selvmotsigelser i forskjellige deler av spesifikasjonen, manglende samsvar med virkeligheten, eller lignende. Ikke alle feil i en spesifikasjon vil nødvendigvis gå ut over interoperabiliteten, men ofte gjør feil i praksis det.

En standard kan også velge **teknisk uheldige løsninger**, som for så vidt gir den funksjonaliteten man ønsker, men på en teknisk lite elegant måte. Dette kan føre til at standarden blir vanskelig å forstå, vanskeligere å implementere enn nødvendig og at transformasjon til andre standarder på samme område blir vanskeligere enn det trenger å være. Igjen kan dette altså føre til dårligere interoperabilitet, men det skal bemerkes at på dette punktet er det langt fra alle svakheter som vil ha konsekvenser for interoperabiliteten. En veldokumentert, men dårlig, løsning kan godt ha utmerket interoperabilitet. Får slike valg først konsekvenser er det ofte vanskelig å kompensere for svakhetene, selv om det fortsatt er mulig.

En annen problemstilling er hvor **stabil** standarden er. Hvis standarden er i stadig endring vil man fort kunne oppleve at interoperabiliteten lider fordi forskjellige verktøy implementerer forskjellige versjoner av standarden, slik at selv om alle følger standarden til punkt og prikke fungerer utveksling likevel ikke helt.

---

<sup>2</sup> Merk at det ikke er noen spesiell grunn til at OOXML *burde* ta med denne funksjonaliteten. Dette er bare et eksempel for å illustrere.

## 3.2 ODF

ODF, OpenDocument Format, skriver seg opprinnelig fra StarOffice, en kontorpakke utviklet av StarDivision. Dette firmaet begynte å utvikle et XML-format for StarOffice i 1999, rett før firmaet ble kjøpt opp av Sun Microsystems. Sun ga ut en åpen kildekode-versjon av StarOffice kalt OpenOffice.org, og i 2002 kom begge pakkene med støtte for et XML-format som den gangen het SXW. Dette formatet utviklet seg med tiden til ODF, og ble standardisert av OASIS som ODF 1.0 i mai 2005 [Macnaghten2007]. ODF 1.0 ble utgitt som ISO-standard ISO/IEC 26300 i mai 2006.

Tabellen under viser en oversikt over de forskjellige ODF-spesifikasjonene som har blitt utgitt av OASIS og ISO. Spesifikasjoner på samme linje har samme innhold.

OASIS	ISO
ODF v1.0 (1st ed)	
ODF v1.0 (2nd ed)	ISO/IEC 26300:2006
ODF v1.1	
ODF v1.2 (uferdig)	

ODF 1.1 ble utgitt av OASIS i februar 2007 og er foreløpig ikke godkjent av ISO, men er sendt til ISO for innarbeidelse i ISO 26300 som amendment 1 [SC34n1404]. ODF 1.1 er publisert som ett enkelt dokument på 738 sider (i PDF-format). Selve ODF-formatet er egentlig en zip-fil som følger visse konvensjoner spesifisert i seksjon 17 av standarden. Zip-filen må inneholde et manifest, som lister opp alle dokumentene i zip-filen og deres filtyper. Seksjon 2.1 beskriver hvilke filer som skal finnes i zip-filen og hva de skal inneholde. Strengt tatt er det mulig å representere et ODF-dokument kun som ett enkelt XML-dokument også, men dette er lite vanlig.

I praksis spesifiserer ODF 1.1 fem XML-formater som er inneholdt i zip-filen: manifest-formatet, selve dokument-formatet, et format for dokumentstiler, et format for metadata og et format for innstillinger. Dokument-formatet omfatter tekstbehandlingsdokumenter, regneark og presentasjoner, og felles innhold er i stor grad delt mellom disse og representert på samme måte, hvilket gjør dem langt enklere å forstå og implementere.

Spesifikasjonen spesifiserer 527 XML-elementer og 1135 attributter, fordelt på 22 forskjellige namespaces i selve dokumentformatet. (Manifest-formatet er her ikke medregnet.) Noen av disse XML-elementene er hentet fra W3C-spesifikasjoner som XLink, SMIL, XSL-FO og SVG, og dette er noe av grunnen til at det brukes så mange forskjellige namespaces.

Der kilder for påstander ikke er oppgitt er kildene ODF 1.1-spesifikasjonen. Denne rapporten vurderer kun ODF 1.1, men ser også på ODF 1.2 på områder der funksjonaliteten i ODF 1.1 vurderes som svak. ODF 1.2 er som det står i tabellen fortsatt ikke ferdig, og det er uklart når den vil bli ferdig.

### 3.2.1 Problemer med standarden

Til forskjell fra OOXML har ikke ODF vært gjenstand for kritisk gjennomgang av utenforstående som høylytt publiserer det de finner av feil og problemer i ODF. Dette gjør det vanskeligere å finne eksempler på feil i standarden annet enn ved å grave dem frem selv, noe som er et tidkrevende og omfattende arbeid. Det at denne seksjonen er kortere og mindre innholdsrik enn den tilsvarende seksjonen for OOXML (kapittel 3.3.1 side 16) betyr dermed ikke nødvendigvis at ODF er en bedre spesifisering.

OASIS har en åpen feildatabase for ODF som inneholder en enorm mengde med feilrapporter. Et enkelt søk i denne på kjente innrapporterte feil i ODF 1.1 gir 131 treff. Listen inneholder uklarheter, småfeil, skrivefeil, tvetydigheter, manglende definisjoner, ting som ikke egentlig er feil i det hele tatt, og noen duplikater.

Under er noen stikkprøver av de virkelige feilene som er innrapportert, slik at man kan danne seg et bilde av hva det er snakk om:

- Attributtet `startrealod` som brukes til å angi en event som skal utløses når et skjema oppfriskes mot databasen skal egentlig hete `startreload`. (Seksjon 13.6, feilnummer 2388.)
- Datatypen `namespacedToken` som brukes på fire forskjellige steder i ODF 1.1 for identifikatorer er feildefinert, det vil si at definisjonen ikke stemmer overens med W3C-spesifikasjonen for XML Namespaces. (Seksjon 17.3.19, feilnummer 2157.)
- RELAX-NG-skjemaene for ODF har en relativt subtil feil i definisjonen av ID-attributter, som gjør at verktøy ikke vil lese skjemaene. (Feilnummer 1352 og 1354.)
- I tegninger er det mulig å oppgi formler som skal brukes for å regne ut visse verdier i tegningene, og ODF 1.1 definerer en syntaks for disse formlene. I syntaksen nevnes 10 funksjoner ved navn, men det er ingen definisjon av disse funksjonene noe sted. (Seksjon 9.5.5, feilnummer 1863.)

Feildatabasen inneholder massevis av slike feil i ODF 1.1, og så vidt vi kan anslå er de fleste av omtrent samme alvorlighetsgrad. Det er altså stort sett snakk om småfeil som gjør en liten del av funksjonaliteten i ODF enten umulig å implementere korrekt fra spesifikasjonen, eller som strider mot andre spesifikasjoner.

Det er slik sett ingen tvil om at den publiserte utgaven av ODF 1.1 er rik på feil og problemer. De enkle stikkprøvene som er gjort i arbeidet med denne rapporten har jo også avdekket alvorlige problemer med spesifikasjonen, noe som jo tyder på de feilene som påpekes i denne rapporten bare er toppen av isfjellet.

### 3.2.1.1 Versjonering

I ODF 1.0 og 1.1 finnes det et attributt `office:version` på det ytterste XML-elementet som brukes for å angi hvilken ODF-versjon dokumentet følger. Dette attributtet er ikke påkrevd, og noen verktøy som kan skrive ODF fyller ikke ut denne verdien, slik at man ikke kan se hvilken versjon av ODF dokumentet følger.

Dette er problematisk fordi mottager av et ODF-dokument dermed ikke kan se hvilken versjon av ODF dokumentet følger uten å gjennomføre en analyse av dokumentet. Dermed kan programvare som mottar dokumentet feiltolke innholdet.

Denne feilen ble innrapportert fra ISO SC34 til OASIS-komiteén som skriver ODF-spesifikasjonen, og i ODF 1.2 er attributtet gjort påkrevd, slik at problemet er løst der [Office1836].

### 3.2.2 Grunnleggende formatering

Beskrivelsen av dette i ODF er så omfattende at vi ikke har gjort noe forsøk på å lese og vurdere denne. Fraunhofer-rapporten [Fraunhofer] beskriver dog i seksjon 5.2.5.1 et problem med lister i ODF 1.0. Listenummerering er tvetydig beskrevet, slik at man kan ende opp med forskjellig nummerering avhengig av hvordan standarden tolkes. Så vidt vi er i stand til å se er problemet også til stede i ODF 1.1, og antagelig også ODF 1.2.

### 3.2.3 Endringssporing

Sporing av endringer er beskrevet i seksjon 4.6 av ODF 1.1-spesifikasjonen. Helt konkret spores endringer ved at en serie elementer av type `text:changed-region` føres opp i en egen liste i dokumentet. Hver endring har en ID som sammenholdes med markører (`text:change-start` og `text:change-end` eller kun `text:change`) som viser hvor i teksten endringen ble gjort. Endringer kan være innsetting, sletting (slettet innhold lagres i listen, ikke der endringen ble gjort), eller formateringsendring.

Hele beskrivelsen av dette dekker kun 4-5 sider i spesifikasjonen, og det er mye som ikke er beskrevet her. Endringer i løpende tekst er greit håndtert, men endringer i tabeller, i matematiske formler, i sidemaler osv er ikke beskrevet i det hele tatt, og delvis ikke støttet. Endring av formatering er også svakt, i og med at hva som ble endret faktisk ikke lagres. Det er heller ikke klart hvordan slettede fragmenter i lister kan gjenopprettes.

Siste utkast til ODF 1.2 (2009-07-03) har ikke endret vesentlig på dette. Faktisk er seksjon 4.5 i ODF 1.2 kortere enn i 1.1, i og med at eksemplene er tatt ut. Det ser dermed ikke ut til at dette vil bli vesentlig forbedret i ODF 1.2.

**Vår konklusjon er** at støtten for dette finnes i ODF, men at den ikke håndterer mer kompliserte tilfeller, og at den generelt må regnes som svak.

### 3.2.4 Kommentarer

Kommentarer er beskrevet i seksjon 12.1 i ODF 1.1-spesifikasjonen. En kommentar består av et `office:annotation`-element som inneholder teksten (en serie avsnitt og lister), valgfri informasjon om plassering, navn på den som lagde kommentaren og tidspunkt. Det hele er beskrevet på knapt 1.5 side, men det virker for så vidt som om funksjonaliteten er vel beskrevet.

Det sies ikke eksplisitt, men indirekte virker det som om plasseringen av kommentaren i teksten gjøres ved å plassere `office:annotation`-elementet i løpende tekst der man vil ha kommentaren. Ut fra teksten i 12.1 ser det ut til at dette kun er mulig i avsnitt og lister (men at disse igjen kan være inneholdt i andre elementer).

Plasseringen av kommentarer gjenbraker attributter fra `draw:caption`-elementet som er beskrevet i seksjon 9.2.10 og for tolkningen av disse henvises man dit. Det er litt uklart hvor godt dette egentlig er beskrevet, og i hvilken grad implementasjoner av dette virkelig vil være interoperable. På den annen side er det viktigste at kommentarene faktisk tas vare på og knyttes til det punktet i dokumentet de gjelder, og det vil være mulig uavhengig av denne informasjonen. Faktisk lagrer ikke OpenOffice.org posisjoneringsinformasjon på kommentarer i det hele tatt, antagelig fordi det ikke er mulig å plassere disse manuelt uansett.<sup>3</sup>

**Vår konklusjon er** at denne funksjonaliteten er vel ivaretatt i ODF 1.1.

### 3.2.5 Formler i regneark

Regnearkformler er spesifisert i seksjon 8.1.3 av ODF 1.1-spesifikasjonen. Det vil si: denne seksjonen spesifiserer ikke noe konkret format på formler. Spesifikasjonen sier: "every formula should begin with a namespace prefix specifying the syntax and semantics used within the formula" (side 190), men den gir ingen indikasjoner om hvordan dette prefikset skal se ut, eller hvilke alternativer som finnes.

Teksten sier videre at det forventes at formler skal begynne med "=" og legger til litt om hvordan henvisninger til celler skal se ut. Dette er hele spesifikasjonen av formler i ODF 1.1. Det er ingen formell grammatikk, ingen oversikt over funksjoner og operatører, bare to enkle eksempler. I det hele tatt gis det ingen ordentlig definisjon av formelspråket.

Det er åpenbart at det vil bli avvik mellom implementasjonene siden formelspråket ikke er dokumentert noe sted, og i tilfelle avvik vil det være umulig å si hvilken implementasjon som er riktig og hvilken som er gal. Dette er jo tross alt en standards funksjon og formål, og på dette punktet finnes det altså et stort hull i ODF-standardens som rett og slett ikke beskriver formelspråket.

Doug Mahugh (fra Microsoft) gir noen gode eksempler på hva dette har ført til på sin blogg [Mahugh2009]. Rob Weir (ODF Architect hos IBM og co-chair for ODF-arbeidsgruppen i OASIS) skriver på sin blogg [Weir2009b] at formelspråket som brukes av OpenOffice.org er basert på Excels formelspråk og viser også gjennom testing at det er problemer med interoperabilitet mellom implementasjonene [Weir2009a].

For å løse dette har OASIS opprettet en "Formula"-underkomité av OpenDocument-komiteén, ledet av David Wheeler, som arbeider med å utvikle et formelspråk kalt OpenFormula. I skrivende stund er det uklart når dette vil være ferdig, men det ser ut til at dette skal bli en del av ODF 1.2. Arbeidet pågår for fullt: siste utkast er datert 3. mars 2010 og er på 241 sider. (Kilde: OASIS hjemmesider.)

Denne spesifikasjonen [OpenFormula] inneholder alt man ville forvente av en slik spesifikasjon: formell grammatikk for syntaksen, definisjoner av datatyper, prosesseringsmodell og definisjoner av operatører og funksjoner. Definisjonene av funksjonene er ganske grundige, med fullstendig definisjon av signaturer (parametre, datatyper osv) og eksempler som viser korrekte verdier for forskjellig input. Den kunne nok likevel vært grundigere på en del punkter, siden mange av funksjonsdefinisjonene avhenger av informasjon som ikke gis i teksten.

---

<sup>3</sup> Egen test med OpenOffice 3.1.1 for Mac 2010-02-04.

**Vår konklusjon er** at ODF 1.1 faller fullstendig gjennom og ikke spesifiserer denne funksjonaliteten i det hele tatt.

Det ser ut til at ODF 1.2s spesifisering av formelspråket vil være langt på vei tilfredsstillende. Rick Jelliffes oppsummering på O'Reilly Community våren 2009 var at det vil være realistisk å forvente interoperabilitet mellom regnearkverktøy med ODF en gang i 2011 [Jelliffe2009]. I dag fremstår det som optimistisk, men fortsatt teoretisk mulig.

### 3.2.6 Makroer i regneark

ODF 1.1 kaller makroer "scripts" og definerer dem i seksjon 2.5, som dekker nøyaktig en side (side 54). Et element kalt `office:scripts` defineres, som inneholder en liste av `office:script`-elementer som har de konkrete makroene. Et attributt på `office:script` angir hvilket makrospråk som brukes, men som standarden sier "since script language names are application specific, the name should be preceded by a namespace prefix." Standarden definerer ingen makrospråk eller navn på disse og har ingen referanser til makrospråk definert andre steder.

Hvordan makroer får tilgang til innholdet i dokumentet er helt åpent. Konkret sier standarden: "Scripts do not imply a scripting language or an object model. A script can for instance operate on the Document Object Model (DOM) composed from the XML representation of a document in OpenDocument format (see [DOM2]), or on an application specific API."

I tillegg finnes det noe som kalles "event listeners" som er definert i seksjon 12.4 over en 3-4 sider. Her gis det navn på konkrete events og det er spesifisert når disse skal utløses. Koden som skal kjøres blir dog definert på samme måten som makroer, slik at selv om eventmodellen er beskrevet i noen grad er makroene som utløses overhodet ikke spesifisert.

OpenOffice.org støtter makroer skrevet i OpenOffice.org Basic, Java, BeanShell og JavaScript, og tilbyr et eget API definert av OpenOffice.org for tilgang til innhold og struktur i dokumenter [OODevGuide]. Hva andre implementasjoner gjør er mer uklart.

**Vår konklusjon er** at ODF 1.1 ikke spesifiserer denne funksjonaliteten i det hele tatt.

ODF 1.2-spesifikasjonen er enda mer kortfattet i sin omtale av makroer enn det ODF 1.1 var, men selve formatet ser ut til å være det samme. Komitéen har besluttet å ikke spesifisere denne funksjonaliteten i ODF 1.2, og det er ikke klart om den noensinne vil bli spesifisert [Office1277].

### 3.2.7 Skjemaer

Skjemafunksjonaliteten er beskrevet i seksjon 11, og mye av spesifiseringen tar for seg funksjonalitet som lar skjemaene kobles mot en datakilde og eventuelt postes til en web services over HTTP. Dette er funksjonalitet som ikke er særlig relevant for den bruk som er identifisert her. Selve spesifiseringen av skjemaer ser ut til å være god nok, og XML-formatet har en oppbygning som fremstår som naturlig og normal.

Seksjon 4.4.1 beskriver skrivebeskyttelse av seksjoner av dokumenter. Noen måte å beskytte et helt dokument på finnes ikke, men i praksis kan man løse dette ved å enten bare ha en seksjon, eller å beskytte alle seksjonene.

Et attributt på seksjons-elementet skruer beskyttelsen av/på, mens et eget attributt (`text:protection-key`) holder det spesifiseringen kaller "a hash value of the password". Dette er for så vidt en vanlig måte å lagre passord på, men ODF 1.1 sier ingenting om hvordan denne verdien produseres. Man kan derfor ikke regne med interoperabilitet mellom ulike implementasjoner på dette punktet.

I ODF 1.2 rettes dette opp, med et eget attributt som brukes for å spesifisere hash-algoritmen. SHA256 anbefales sterkt, selv om andre alternativer tillates.

**Vår konklusjon er** at støtte for skjemaer er godt spesifisert, men at skrivebeskyttelse av dem er altfor dårlig spesifisert.

### 3.3 OOXML

OOXML har sitt utspring i et XML-format for Microsoft Excel, som ble utgitt i 2000. Dette ble siden fulgt av et XML-format for Word i 2002, og begge ble gitt ut som en del av Microsoft Office i 2003. Disse formatene ble siden videreutviklet til det som i dag heter Office Open XML, eller OOXML, og ble standardisert i ECMA i 2005 og 2006.

Den første spesifikasjonen, ECMA-376:2006, består av fem ulike deler på tilsammen 6045 sider:

1. Fundamentals (178 sider).
2. Open Packaging Conventions (131 sider). Beskriver hvordan dokumenter pakkes inn i zip-filer, og tar for seg ting som filnavn og formater, kryptering og signaturer, thumbnails, osv. Denne delen er strengt tatt uavhengig av OOXML-formatet.
3. Primer (473 sider). Denne delen er en innføring i innholdet i del 4, og har faktisk ikke noe normativt innhold (dvs: den standardiserer ingenting, den inneholder bare forklarende tekst). Så godt som alt innholdet ser ut til å være duplisert fra del 4.
4. Markup Language Reference (5220 sider). Dette er selve definisjonen av OOXML, inndelt i WordprocessingML (seksjon 2), SpreadsheetML (seksjon 3), PresentationML (seksjon 4), DrawingML (seksjon 5), VML (seksjon 6) og delte XML-språk (seksjon 7).
5. Markup Compatibility and Extensibility (43 sider). Denne delen beskriver hvordan man kan inkludere egne XML-elementer i OOXML-dokumenter, slik at disse kan håndteres uten misforståelser av mottagerne. Hensikten er å tillate private utvidelser og fremtidige utvidelser av standarden med minst mulig påvirkning på programvare som ikke kjenner utvidelsene.

I tillegg defineres formatet av formelle skjemaer for OOXML distribuert elektronisk i 87 egne filer, i både RELAX-NG- og W3C XML Schema-format. Disse definerer ikke mindre enn 2301 elementer og 1988 attributter fordelt på 29 navnerom.

Denne rapporten vurderer ECMA-376:2006, men det har siden kommet flere nye versjoner av OOXML fra både ECMA og ISO. Tabellen under gir en kort oversikt. Spesifikasjoner på samme rad har samme innhold.

ECMA	ISO
ECMA-376:2006	
ECMA-376:2008	ISO/IEC 29500:2008
	ISO/IEC 29500:2008 corrigendum 1 (uferdig)
	ISO/IEC 29500:2008 amendment 1 (uferdig)

Merk at offisielt regner man ikke ECMA-376:2006 og :2008 som to forskjellige versjoner av formatet OOXML. I stedet regner man OOXML Transitional (spesifisert i ECMA-376:2006, og med små endringer også i ECMA-376:2008) og OOXML Strict (spesifisert i ECMA-376:2008, ISO 29500:2008, og amendment 1) som forskjellige versjoner av formatet. Tanken bak er at Transitional skal spesifisere "oppførselen til eksisterende implementasjoner" (det vil i praksis si Microsoft Office, og programvare basert på "reverse engineering" av formatet), mens Strict skal være en mer fremtidsrettet versjon som utelater mye av den historiske ballasten i formatet.

Dette skillet ble introdusert som et kompromiss for å gjøre OOXML-formatet akseptabelt for ISO. I første avstemning ble OOXML avvist av ISO fordi det var for mye inkonsistenser og duplisering av funksjonalitet i formatet. Strict-versjonen fjerner dette, samtidig som Transitional-versjonen følger eksisterende implementasjoner. Dette kompromisset var det som gjorde at OOXML dermed ble akseptabelt for ISO og ble akseptert i andre forsøk [Brown2010].

Den siste utgaven, amendment 1, er et sett med endringer til ISO 29500:2008, og ikke en ny versjon av selve dokumentet. Denne versjonen endrer XML-namespacet for Strict-dokumenter, slik at det blir mulig å se forskjell på dokumenter i Transitional-format og Strict-format. Denne endringen er fortsatt under avstemning i ISO og ikke formelt godkjent enda [Hamilton2010].

ECMA-376:2008 og ISO 29500:2008 bruker det gamle namespace for Strict, slik at det er egentlig amendment 1 som spesifiserer Strict-versjonen korrekt. Man kan dermed si at i dag er det egentlig bare Transitional som har en offisiell spesifisering.

Der ikke annet er oppgitt er kilden for påstandene her ECMA-376:2006-spesifikasjonen. Referanser til deler og seksjoner i spesifikasjonen er også til denne versjonen.

### 3.3.1 Problemer med spesifikasjonen

OOXML har nå vært en publisert ECMA-standard i fire år, og i løpet av denne tiden har den blitt gjenstand for nokså hard kritikk fra ulike hold. I tillegg har arbeidet med denne rapporten avdekket en del problemer. Dette kapittelet gir en kort oversikt over disse svakhetene.

#### 3.3.1.1 Datoer i regneark

Datohåndteringen i regnearkdelen av OOXML har vært gjenstand for til tider kross kritikk på nettet, og denne har fokusert særlig på det at OOXML ikke støtter datoer før 1900 [OpenMalaysia2007a]. Dette skyldes at OOXML bruker noe som kalles "serial dates", det vil si at en dato representeres som et tall. 1 er 1. januar 1900<sup>4</sup> og alle andre tall datoen som er så mange dager etter denne dagen, slik at 2. blir 2. januar 1900, 32 blir 1. februar 1900, osv. Negative tall er ikke tillatt, og datoer etter 31. desember 9999 er heller ikke lov.

Det kan enkelt verifiseres i Excel. Excel 2008 for Mac vil ikke formatere datoer tidligere enn 1904, og vil heller ikke akseptere disse som argumenter til f.eks `weekday`-funksjonen. Tilsvarende vil `=datevalue("1904-01-01")` returnere 0 i Excel 2008 for Mac, mens `=datevalue("1903-12-31")` vil gi en feilmelding.

Et annet problem er at spesifikasjonen eksplisitt sier at "an implementation using the 1900 date base system shall treat 1900 as though it was a leap year" (seksjon 3.17.4.1). I det gregorianske kalendersystemet var 1900 ikke et skuddår, slik at dette er faktisk galt. Implementasjoner må dermed kode seg rundt denne underligheten, og det blir mulig å representere 1900-02-29, som faktisk ikke var noen virkelig dato. Også på dette punktet har OOXML vært gjenstand for til dels kross kritikk [Weir2006] [Oram2007].

Når det gjelder datoer før 1900 er dette et åpenbart eksempel på manglende funksjonalitet i OOXML, mens dette med skuddår er mer å regne som et tilfelle av teknisk uheldige valg. Årsaken til at OOXML har valgt å gjøre dette slik er ganske enkelt at dette er datorepresentasjonen til Lotus 1-2-3, slik at dette har vært vanlig praksis i regneark helt siden 1980-tallet [Horton2009].

Til tross for alt oppstyret rundt dette problemet er det ikke klart at dette faktisk har noen konsekvenser for bruk av OOXML i norsk forvaltning. Ingen av scenariene i denne rapporten antyder noe behov for datoer før 1900, eller for korrekt håndtering av skuddår i 1900.

Det skal også bemerkes at ISO 29500:2008 endrer datorepresentasjonen til å bruke ISO 8601-formatet i både OOXML Strict og Transitional. Det vil si at 1. januar 1900 representeres som 1900-01-01 og 31. januar 1899 som 1899-01-31. Det er uklart om OOXML Transitional vil gå tilbake til seriedatoer i fremtiden [Horton2009].

#### 3.3.1.2 Selvmotsigelser

Spesifikasjonen inneholder interne selvmotsigelser. To slike har blitt identifisert i dag:

- OPC spesifiserer ulike egenskaper (properties) som en fil kan ha, og "core properties" er de egenskapene som er definert av OPC. Del 1, seksjon 15.2.11.1 sier at identifikatoren for core properties er <http://schemas.openxmlformats.org/officedocument/2006/relationships/metadata/core-properties>, mens del 2 annex F sier at den skal være <http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties>. Det er ikke mulig å bruke begge

<sup>4</sup> Seksjon 3.17.4.1 av del 4. Merk at denne også spesifiserer en alternativ startdato, som er 1. januar 1904. En egen egenskap på arbeidsbøker i OOXML-formatet angir hvilken av de to base-datoene som brukes i arbeidsboken.



identifikatorene, slik at så lenge man spesifiserer noen core properties vil man være i strid med en del av standarden. Alle versjoner av Microsoft Office bruker forøvrig den siste URL-en.

- Det andre problemet er også i OPC. Del 2 seksjon 8.1.1 sier at referanser til filer inne i zip-filen skal begynne med "/", mens seksjon 8.3.3.2 sier at den skal være en absolutt eller relative URL. Microsoft Office bruker det andre alternativet, mens Microsofts XML Paper Specification, som også bruker OPC, bruker det første.

Dette er reelle feil som høyst sannsynlig vil føre til problemer med interoperabilitet, men i og med at alt dette er på OPC-nivå snarere enn på OOXML-nivå er det lite trolig at disse feilene vil få noen særlige konsekvenser i norsk offentlig sektor.

### 3.3.1.3 Versjonering

Et annet ankepunkt mot OOXML, som har meldt seg etter at ISO har begynt å jobbe med spesifikasjonen, er at det nå finnes flere forskjellige versjoner av formatet. Disse har ikke offisielle navn, men det dreier seg kort fortalt om følgende:

- Transitional 1, dvs som spesifisert i ECMA-376:2006.
- Transitional 2, dvs som spesifisert i ECMA-376:2008 og ISO 29500:2008.
- Transitional 3, dvs som spesifisert i ISO 29500:2008 amendment 1. Det er uklart om det faktisk er gjort noen endringer i transitional i denne utgaven av standarden.
- Strict 1, som spesifisert i ECMA-376:2008 og ISO 29500:2008.
- Strict 2, som vil bli spesifisert i ISO 29500:2008 amendment 1.

Problemet er at gitt en OOXML-fil er det svært vanskelig å si ut fra innholdet om denne følger Transitional 1, Transitional 2, Transitional 3 (dersom denne finnes) eller Strict 1. Den eneste måten å sjekke dette på er å se etter verdier/elementer som er spesifikke for en av disse versjonene [Stocholm2009]. For Strict 2 er dette heldigvis løst, ved at XML-namespacene er endret, slik at man lett ser at det dreier seg om et Strict 2-dokument.

Dette er en svakhet som i praksis kan gi alvorlige problemer for interoperabiliteten, for det første fordi det finnes forskjellige varianter av formatet som ikke alle implementerer. For det andre fordi det er svært krevende for implementasjoner å se hvilken versjon som er i bruk i et gitt dokument. I dag er dette ikke så problematisk, ettersom de aller fleste implementasjoner kun forholder seg til ECMA-376:2006, men dette vil forandre seg etterhvert som implementasjonene oppdateres.

### 3.3.1.4 Diverse

Generelt er det mye rusk og slurv i ECMA-376:2006. Rob Weir har hatt stor glede av å grave frem mye av dette, og presenterer en lang liste på bloggen sin [Weir2008]. Ikke alt han nevner er virkelige feil, men det er en god del reelle problemer i listen.

Det viser seg også at skjemafilene som følger med ECMA-376:2006 har sine problemer. Det er to sett med skjemaer, i W3C XML Schema og i RELAX-NG. Følgende RELAX-NG-filer har syntaksfeil:

- dml-diagramColorTransform.rnc
- dml-diagramStyleDefinition.rnc
- pml-viewProperties.rnc
- wml.rnc (3 feil)
- sml-workbook.rnc
- sml-pivotTable.rnc (2 feil)
- sml-queryTable.rnc

Dette betyr at disse filene ikke kan leses med verktøy i det hele tatt før feilene blir rettet manuelt. Feilene er temmelig systematiske, antagelig forårsaket av dårlig konvertering fra XML Schema, og å gjette en rettelse er ganske enkelt, men det hele tyder på at arbeidet med standarden har vært temmelig slurvete.

Et annet problem som ofte nevnes er at OOXML inneholder fire forskjellige formater: WordprocessingML, SpreadsheetML, PresentationML og DrawingML. Disse har en del funksjonalitet felles, som f.eks enkel

formatering av løpende tekst, men hvert \*ML har definert sitt eget format for å håndtere dette. For eksempel finnes det forskjellige måter å uttrykke at tekst skal være med fet skrift i de fire forskjellige formatene. Dette gjør OOXML langt større og mer komplisert enn det trenger å være, og gjør det også mer kostbart å implementere.

### 3.3.2 Endringssporing

I OOXML kalles dette "revisions" og er spesifisert i seksjon 2.13.5 av del 5 over ca 120 sider. Beskrivelsen av funksjonaliteten er grundig, men ikke så omfattende som sidetallet kan gi inntrykk av, siden mye av informasjonen er duplisert (attributtene på de forskjellige elementene defineres på nytt for hvert element selv om attributtene er helt like; dette tar opp ca 1 side per element, osv).

Grovt sett kan man dele funksjonaliteten for endringssporing inn i følgende områder: tabeller (innskudd, sletting, sammenslåing, endring av egenskap), egendefinert XML (sletting, innskudd, flytting), endring i tekst (sletting, innskudd, flytting, endring av egenskap) og mer spesialiserte ting (f.eks endring av listenummerering).

Å analysere disse 120 sidene i detalj ville vært en ganske krevende øvelse, men et overfladisk studium tyder på at dette er godt spesifisert og at de fleste tilfeller er tilstrekkelig beskrevet.

**Vår konklusjon er** at dette er godt spesifisert.

### 3.3.3 Kommentarer

Kommentarer er spesifisert i seksjon 2.13.4 av del 4 og består av ca 30 sider tekst, igjen med mye duplisering av informasjon. Teksten er likevel ganske omfattende og detaljert.

En kommentar har to deler: en markør som viser hva kommentaren gjelder og selve kommentaren. Markøren er enten bare ett element som henviser til kommentaren (`w:commentReference`) hvis markøren bare er et punkt, eller to elementer som angir start og slutt på det kommenterte området (`w:commentRangeStart` and `w:commentRangeEnd`) pluss referansen.

Selve kommentaren representeres av et `w:comment`-element som har metadata om kommentaren (forfatter og tidspunkt) samt selve kommentarteksten. Denne ligger i en egen fil (`comments.xml`) inne i zip-filen som utgjør selve OOXML-dokumentet.

**Vår konklusjon er** at denne funksjonaliteten er grundig spesifisert og ser ut til å være godt ivaretatt i OOXML.

### 3.3.4 Formler i regneark

Formelspråket i OOXML er definert i seksjon 3.17 av del 4 av spesifikasjonen og definisjonen dekker litt over 300 sider. Den inneholder en formell grammatikk for språket, definisjon av operatører og datatyper, presisjon på datatypene, evalueringsmodell og definisjoner av innebygde funksjoner.

Det hele ser svært grundig og detaljert ut, men motstanderne av OOXML har funnet svakheter i definisjonene av ca 30 funksjoner [Weir2007]. Noen av disse påståtte svakhetene er ikke så alvorlige. Det står f.eks ikke eksplisitt hva slags grader som er input til de trigonometriske funksjonene, men eksemplene viser tydelig at det er radianer det er snakk om. Stort sett er det snakk om mindre svakheter, men de tyder klart på slurv, og en del av disse svakhetene vil nok i praksis påvirke interoperabiliteten.

**Vår konklusjon er** at dette er godt spesifisert, om enn ikke feilfritt.

### 3.3.5 Makroer i regneark

ECMA-376:2006 inneholder en rekke definisjoner av syntaks for å henvise til makroer ved navn, men spesifiserer ikke hvor i filene disse makroene skal defineres, eller hva slags språk de skal skrives i.

**Vår konklusjon er** at makroer ikke er spesifisert i OOXML i det hele tatt.

### 3.3.6 Skjemaer

Seksjon 2.16.18-23 av del 4 beskriver skjemafelter. Disse har en nokså eiendommelig XML-representasjon, ved at man bruker et element som heter `w:fldChar` som settes inn tre ganger for å representere f.eks et tekstfelt. Først kommer en `w:fldChar` av type "start", så en av type "separate", og til slutt en av type "end". Forskjellige typer parametre til feltet legges inn før og etter "separate"-elementet. Det hele blir ganske komplisert, og å forstå hvordan det fungerer fra spesifikasjonen er ikke lett, men det ser ut til å være tilstrekkelig spesifisert.

Seksjon 2.15.1.28 i del 4 beskriver hvordan man beskytter et dokument. Det finnes ulike typer beskyttelse (beskrevet i seksjon 2.18.22), der "forms" er den som gir beskyttelse av skjemaer. I tillegg er det en omfattende beskrivelse av kryptering, dvs ulike algoritmer, "cryptography providers", parametre til algoritmene med mer. Det er uklart hvorfor dette er så ekstremt komplisert, i og med at dette ikke egentlig gir noen sikkerhet, som spec-en også sier selv. Funksjonaliteten er dog tilstrekkelig beskrevet, skjønt kompleksiteten i denne funksjonaliteten gjør at det er langt fra sikkert at alle implementasjoner vil være fullt ut kompatible.

**Vår konklusjon er at dette er godt nok spesifisert.**

## 3.4 Interoperabilitet

Fraunhofer Institut har skrevet en omfattende rapport om interoperabilitet mellom ODF og OOXML som tar utgangspunkt i scenarier og vurderer interoperabilitet for forskjellig funksjonalitet ut i fra disse [Fraunhofer]. Rapporten vurderer ikke skjemaer og makroer, som vi har tatt for oss over, men dekker ellers den funksjonaliteten som vi har identifisert behov for.

Rapportens vurderinger av den funksjonaliteten som er interessant i henhold til scenariene i denne rapporten beskrives kort under.

### 3.4.1 Grunnleggende formatering

Fraunhofer-rapporten går gjennom dette i detalj i seksjon 5.2.1-5.2.6 i lange tabeller som vurderer hver enkelt funksjonalitet separat. Snittet av vurderingene ser ut til å ligge omtrent på "medium", ofte fordi det ene formatet støtter mer enn det andre (ofte er det ODF som har minst funksjonalitet).

### 3.4.2 Endringssporing og kommentarer

Interoperabilitet for akkurat denne funksjonaliteten er vurdert i seksjon 5.2.7 av Fraunhofer-rapporten, som finner at denne stort sett er "medium" for endringssporing og "high" for kommentarer. Vanskelighetene med interoperabilitet skyldes ifølge rapporten stort sett at OOXML har funksjonalitet som ODF ikke har, og som dermed ikke kan konverteres. Dette stemmer godt med min egen vurdering av denne funksjonaliteten.

### 3.4.3 Formler i regneark

Interoperabilitet for formler er vurdert i seksjon 5.3.3 av Fraunhofer-rapporten som konkluderer med at denne stort sett er "Low" skjønt to punkter ligger høyere. Problemet er at ODF ikke spesifiserer noe formelspråk. I tillegg er det andre problemer med manglende funksjonalitet i ODF, som f.eks navngitte celler, delte formler og automatisk filtrering av tabeller. Igjen stemmer konklusjonene godt med mine egne vurderinger.

## 3.5 Oppsummering

Generelt kan man vel ikke si at ECMA-376:2006 og ODF 1.1 fremstår som spesielt imponerende, som spesifikasjoner betraktet, men svakhetene deres er ganske forskjellige. ECMA-376 er først og fremst for stor og vanskelig å få grep på, og skjemmes av en del slurv og unøyaktigheter, samt at den inneholder mye som burde vært luket ut og ryddet opp i. For eksempel er det ingen grunn til å ha to formater for vektorgrafikk (VML og DrawingML) i samme standard. Men tross alt beskriver ECMA-376 formatet grundig og utfyllende og tilbyr svært omfattende funksjonalitet.

ODF 1.1 er til sammenligning langt lettere å sette seg inn i, og har en teknisk tilnærming som fremstår som langt mer naturlig. Selve standarden er også langt mindre, hvilket selvsagt bidrar til å gjøre den enklere å forstå. Ulempen er at mye funksjonalitet mangler, og at ikke all funksjonaliteten er tilstrekkelig beskrevet.

Litt forenklet kan man si at ECMA-376 først og fremst trenger en opprydning der småfeilene rettes og det overflødig skjæres vekk, mens ODF 1.1 trenger å fullføre arbeidet, for å fylle ut hullene. Begge prosessene er i gang i henholdsvis ISO og OASIS, og det er all grunn til å tro at kvaliteten på begge spesifikasjonene vil heves i tiden framover. Dette er dog et langt lerret å bleke, og internasjonal standardisering er en kvern som maler langsomt og ujevnt, så bare tiden vil vise hva som faktisk kommer ut av dette.

Hva angår det første kriteriet i kapittel 3.1 side 10 funksjonalitet, er det tydelig at OOXML tilbyr mer komplett funksjonalitet enn hva ODF gjør [Fraunhofer]. Av den funksjonaliteten som det er identifisert behov for i scenariene er de største manglene kanskje makroer (mangler i begge) og regnearkformler (mangler i ODF 1.1).

Når det gjelder hvorvidt den nødvendige funksjonaliteten er spesifisert med tilstrekkelig presisjon er svaret oppsummert i tabellen under:

Funksjonalitet	OOXML	ODF
Endringssporing	Bra	Ikke godt nok
Kommentarer	Bra	Godt nok
Formler regneark	Godt nok	Ikke spesifisert
Makroer regneark	Ikke spesifisert	Ikke spesifisert
Skjemaer	Godt nok	Ikke godt nok

Her kommer OOXML betraktelig bedre ut enn det ODF gjør, og bortsett fra makroer kan man vel si at OOXML faktisk kommer godt nok ut av det på dette punktet. Om ODF gjør det kan diskuteres. De største manglene gjelder regneark, som ikke er den viktigste anvendelsen.

Når det gjelder forståelighet er nok ODF hakket bedre enn OOXML, både fordi standarden er mindre, men også fordi selve formatet er enklere og mer i tråd med hva en utvikler skolert i XML ville forvente å finne i et slikt format.

Hva angår tekniske feil ser ODF noe bedre ut enn OOXML, men det er uklart om dette kun skyldes mindre oppmerksomhet rundt det som faktisk finnes av feil i ODF. Kvaliteten på de tekniske valgene som er gjort fremstår som bedre i ODF på en rekke punkter.

Når det gjelder stabiliteten imponerer OOXML ikke. OOXML Transitional har sett endringer på en rekke punkter (attributtverdier som er endret, nytt datoformat, ...), og OOXML Strict er ennå ikke ferdig definert. Mye av turbulensen her skyldes at man i ISO har forlangt å få rydde opp i OOXML, et arbeid som ECMA ikke prøvde seg på i det hele tatt.

Hva angår ODFs stabilitet er denne mer uklar. ODF 1.2-spesifikasjonen har en svært omfattende liste over endringer i formatet, men det aller meste ser ut til å være utvidelser, og ikke endringer av den typen som kan forventes å skade interoperabiliteten. Det virker dog som om det kan bli problemer med å lese ODF 1.2-dokumenter på ODF 1.1-implementasjoner.

Kriterie	OOXML	ODF
Funksjonalitet	Bra	Ikke godt nok
Presisjon	Bra	Ikke godt nok
Forståelighet	Godt nok	Godt nok

Kriterie	OOXML	ODF
Korrekthet	Ikke godt nok	Ikke godt nok
Tekniske valg	Godt nok	Godt nok
Stabilitet	Dårlig	Godt nok

Kort oppsummert kan man si at ODF feiler på det viktigste: funksjonaliteten, mens OOXML feiler på stabiliteten. I tillegg får begge strykkarakter på korrektheten. Det hefter alvorlige feil og mangler ved begge spesifikasjonene, men spesifikasjonene har altså problemer på ulike punkter.

## 4 Implementasjoner

Hvilke implementasjoner som finnes for de ulike standardene er avgjørende for hvilke som kan gjøres anbefalt og påkrevd for de forskjellige anvendelsesområdene. Dette kapittelet gir en liten oversikt over hvilken programvare som finnes innenfor de forskjellige kategoriene programvare, og hvilke formater (og versjoner av formatene) disse verktøyene støtter.

Denne oversikten er grunnlagsmateriale for de vurderinger som siden gjøres. Oversikten er stort sett innhentet fra engelsk Wikipedia, med noe egen utforskning av leverandørnettsteder (og tilhørende redigering på Wikipedia) for å avklare uklare punkter.

I tillegg beskrives resultatene av å teste utveksling mellom de verktøyene som virker relevante for norsk offentlig sektor. Her testes den funksjonaliteten som det er identifisert behov for ut i fra scenariene i kapittel 5 side 29. En detaljert oversikt over verktøyene vi har testet med finnes i appendiks 7 side 38.

### 4.1 OOXML

Her gir vi en liten oversikt over hvilken kontorprogramvare som støtter OOXML-formatet. Denne tabellen viser hvilke produkter som kan lese og/eller skrive OOXML.

Navn	Lese	Skrive	Internformat	Kommentar
AbiWord	x	x		Fra versjon 2.6
Adobe Buzzword	x	x		Web-basert
Apple iWork	x			
Apple iPhone	x			
Apple TextEdit	x	x		Begrenset støtte
Atlantis WP	x	x		Kun Windows
Bean OS X	x	x		Begrenset støtte. Bruker Apples konvertering.
Butler Office Pro X	x	x		En variant av NeoOffice.
Corel WordPerfect	x			Kun Windows
docx4all	x	x	x	Java-basert.
EIOffice WP	x			Kinesisk.

Navn	Lese	Skrive	Internformat	Kommentar
Go-oo	x	x		En versjon av openoffice.org
Google Docs	x			Web-basert
Hangul WP 2009	x	x		Koreansk. Ikke utgitt enda.
IBM Lotus Symphony	x			Basert på openoffice.org
Jarte 3.0+	x			
JustSystems Ichitaro	x	?		Japansk.
Kingsoft Office Writer	x			
Madcap Blaze		x		
Microsoft Office	x	x	x	
NeoOffice	x	x		Basert på openoffice.org
Nisus Writer Pro	x			
Openoffice.org	x			
QuickOffice	x	x		For mobile plattformer.
Schreiben	x	x		Kun MacOS
ThinkFree Office	x	x		Java-basert, men også web-utgave
Zoho Writer	x	x		Web-basert

Tabellen viser også hvilke verktøy som har OOXML som internformat, og hvilke som støtter det via konvertering. Det er en vesensforskjell mellom disse to ulike måtene å støtte et format på, i og med at når et format er internformat bygges den interne representasjonen rundt dette formatet, mens ved konvertering må man gjøre en transformasjon fra de interne strukturene til det eksterne formatet. Dette betyr i praksis at funksjonaliteten som støttes i eksterne formater gjerne bare blir den funksjonaliteten som finnes i både internformatet og det eksterne formatet. Det betyr også i noen tilfeller at enkelte strukturer kan forandre seg, fordi det ikke er mulig å få til eksakt samme representasjon i det eksterne formatet. Av disse grunnene er støtten for et format i et verktøy som har det som internformat som regel bedre enn i et verktøy som har det som eksternt format.

Ikke alle disse verktøyene er like kjente og utbredte, og det er heller ikke alle som egner seg like godt for bruk i norsk offentlig sektor. Vi har ikke oversikt over absolutt alle disse, men mener at i hvert fall verktøyene i tabellen under har tilstrekkelig utbredelse og funksjonalitet til at de kan være realistiske alternativer til bruk i norsk offentlig sektor. Tabellen viser hvilke plattformer de forskjellige verktøyene finnes på, for å gi en oversikt over dekningen på forskjellige plattformer. (Merk at tabellen kun tar med verktøy som kan skrive OOXML, siden verktøy som ikke kan dette kun unntaksvis er relevante.)

Name	Mac	Windows	Linux	Kommentar
AbiWord		x	x	2.6 finnes ikke for Mac.
Go-oo	x	x	x	
Microsoft Office	x	x		
NeoOffice	x			

Mac og Windows er de plattformene der det er flest alternativer å velge mellom (3), men selv på plattformen med dårligst utvalg (Linux) er det to alternativer. Dermed ser det i utgangspunktet ut til å finnes tilstrekkelig med programvare for OOXML-formatet til at det er mulig å anbefale det.

For at disse verktøyene skal kunne brukes i norsk offentlig sektor må det finnes oversettelser av dem til bokmål og nynorsk. Listen under oppsummerer kort status:

- **AbiWord**: oversettelser som er 91% og 93% komplette finnes. Det burde være mulig å fullføre disse dersom man ønsket å bruke AbiWord.
- **Go-oo**: har komplette oversettelser til både nynorsk og bokmål.
- **Microsoft Office**: har komplette oversettelser til både nynorsk og bokmål.
- **NeoOffice**: har komplette oversettelser til både nynorsk og bokmål.

I kapitlene under ser vi på hvordan støtten for spesifikk funksjonalitet er i de forskjellige verktøyene for å kunne vurdere hvilke som kan brukes i de forskjellige anvendelsesscenariene som kommer lenger ned. Her er det kun verktøyene fra den siste tabellen som blir vurdert.

Vi har ikke gjort noe for å sjekke hvilke versjoner av OOXML som implementeres av de forskjellige verktøyene, men i og med at OOXML Strict ennå ikke er ferdig publisert kan man trygt anta at det som implementeres er OOXML Transitional. Microsoft Office 2007 og 2010 implementerer begge OOXML Transitional som spesifisert i ECMA-376:2006 ([Brown2008] og [Brown2010]). Microsoft hevder at støtte for å skrive OOXML Strict vil komme i Office 15 (dvs neste versjon; Office 14 var kodenavnet for Office 2010), og at Office 2010 allerede kan lese OOXML Strict [Mahugh2010].

#### 4.1.1 Grunnleggende formatering

For å vurdere i hvilken grad grunnleggende formatering er skikkelig støttet lagde vi et test-dokument i Word som brukte følgende funksjoner:

- Vanlige avsnitt med tekst,
- Overskrifter (med stiler),
- Innholdsfortegnelse,
- Tabeller,
- Lister,
- Farger,
- Skifte av font,
- Header og footer,
- Kryssreferanser,
- Fotnoter,
- URL-er,
- Bitmap-bilder og
- Vektorgrafikk.

Dette dokumentet ble så lagret og åpnet igjen i verktøyet som skulle testes. Eventuelle feil ble rettet opp der før dokumentet ble lagret på nytt, og så forsøkt åpnet i Word.

I AbiWord ble innholdsfortegnelsen til vanlig tekst, headeren forsvant, listene fikk noe underlig nummerering og ble galt formatert og vektorgrafikken ble ødelagt. Ellers så det ut til å fungere bra.

NeoOffice gjorde det ganske bra i denne testen. Headeren fikk ødelagt formateringen og vektorgrafikken ble borte, men ellers åpnet dokumentet korrekt i NeoOffice. På vei tilbake til Word ble lister som var på flere nivåer flatet ut, men ellers åpnet dokumentet korrekt. I praksis er vel dette knapt godt nok, siden dette var et lite og enkelt dokument, men det tyder på at grunnleggende formatering i hvert fall delvis fungerer.

Go-oo åpnet Word-dokumentet omtrent like bra. Headeren så ikke helt bra ut, men bedre enn i NeoOffice. Noen listepunkter ble til rare hake-merker, og vektorgrafikken ble delvis ødelagt. Ellers åpnet dokumentet helt korrekt. Etter at dette var rettet opp klarte ikke Word å åpne dokumentet i det hele tatt lenger, slik at på vei tilbake feilet det totalt. Om feilen skyldes Go-oo eller Word skal være usagt.

Google Docs åpnet dokumentet stort sett korrekt, men sidetallet falt ut av headeren og ble satt inn i selve teksten i dokumentet. Innholdsfortegnelsen ble til vanlig tekst og ble nokså stygt formatert. Vektorgrafikken falt også ut. I det store og det hele var ikke dette resultatet blant de verste.

Konklusjonen er at dette egentlig bare virket i Microsoft Office, mens Google Docs kanskje kan brukes i noen tilfeller. Google Docs kan dog ikke skrive OOXML.

#### **4.1.2 Endringssporing**

Dette ble testet ved å ta et eksisterende dokument og gjøre endringer i det med endringssporing i Microsoft Office. Dokumentet ble så lagret og åpnet i verktøyet som skulle testes. Deretter ble det gjort endringer i dette verktøyet også, og dokumentet lagret og åpnet igjen i Microsoft Office.

I AbiWord var det ikke spor av endringene som var lagret i dokumentet, selv om AbiWord ser ut til å støtte revisjoner av dokumenter.

NeoOffice håndterte dette helt korrekt.

Go-oo støtter sporing av endringer, men dokumenter med endringer lot seg ikke åpne igjen i Microsoft Office, så Go-oo kan ikke sies å ha tilfredsstillende interoperabilitet.

Kort oppsummert fungerte dette kun i NeoOffice.

#### **4.1.3 Kommentarer**

Dette ble testet ved å legge inn noen kommentarer av forskjellig type på ulike steder i et dokument i Microsoft Office. Dokumentet ble så åpnet i forskjellige verktøy for å se om kommentarene ble bevart. Dersom dette lyktes ble kommentarene endret og flyttet og dokumentet lagret for så å åpnes i Word igjen.

Apple TextEdit, AbiWord og Nisus Writer Pro fjernet kommentarene uten å gjøre skade på dokumentet. I Go-oo ble kommentarene for så vidt bevart, men teksten ble summen av teksten i alle kommentarene, og informasjon om forfatter og tid gikk tapt. NeoOffice og Google Docs fanget ikke opp at kommentarene var kommentarer, og satt dem i stedet inn i teksten som vanlig tekst.

I Apple iWork (Pages) fungerte det helt perfekt, bortsett fra at punktkommentarer ble kommentarer på nærmeste hele ord.

Kort oppsummert kan man si at dette overhodet ikke fungerte i vår test, annet enn i Pages, som ikke kan skrive OOXML.

#### **4.1.4 Formler i regneark**

Vi gjorde en liten test av dette, med et enkelt regneark bestående av to arbeidsbøker, noen enkle formler med referanser på tvers og noen celler med feil datatype (text i stedet for number). Dette regnearket ble laget i Excel, åpnet i andre verktøy, tall fylt inn, og regnearket lagret, og så åpnet igjen i Excel.

I NeoOffice, Gnumeric, og Go-oo fungerte dette fint. Det åpnet også riktig i IBM Lotus Symphony og Google Docs, men disse kan ikke skrive .xlsx-formatet.



Apple Numbers åpnet regnearket nesten helt korrekt: et tall som var formatert som tekst falt ut, og dermed ble summene i formlene gale. Dette er i praksis ikke godt nok.

I utgangspunktet ser dette ut til å fungere fint i de verktøyene som faktisk støtter regnearkformatet fullt ut (unntatt Apple Numbers), i hvert fall for enklere formler og regneark.

#### 4.1.5 Skjemaer

Vi testet dette ved å lage et enkelt skjema i Word med noen forskjellige typer felter, og låste dokumentet for endringer. Dette ble så åpnet i andre verktøy, utfylt, og åpnet igjen i Word.

NeoOffice fikk ikke med seg at dokumentet var låst, og selve skjemafeltene forsvant, så her fungerte det ikke i det hele tatt. I Go-oo og AbiWord var resultatet det samme.

På dette punktet ser altså OOXML ikke ut til å fungere på tvers av implementasjoner i det hele tatt.

## 4.2 ODF

Tabellen under viser en oversikt over alle de kontorpakkene vi kjenner til som har støtte for ODF-formatet.

Navn	Lese	Skrive	Versjon	Internformat	Kommentar
AbiWord	x	x	1.0		
Apple TextEdit	x	x	1.0		Ufullstendig støtte.
Celframe Office	x	x?	?	?	
Go-oo	x	x	1.2	x	En versjon av OpenOffice.org
Google Docs	x	x	1.1		Web-basert
IBM Lotus Symphony	x	x	Uspesifisert	x	Basert på OpenOffice.org
JustSystems Ichitaro	x	x	?		
KOffice	x	x	?		
Microsoft Office	x	x	1.1		Office 2007 SP2 [Microsoft2008]
NeoOffice	x	x	1.2	x	Basert på OpenOffice.org (kun Mac)
Nisus Writer	x		?		Kun Mac
OpenOffice.org	x	x	1.2	x	
Schreiben	x	x	?		Kun Mac
SoftMaker TextMaker	x	x	?		
Wordperfect Office	x		?		

Navn	Lese	Skrive	Versjon	Internformat	Kommentar
Zoho Writer	x	x	?		

Også denne tabellen viser hvilke implementasjoner som har ODF som internformat, og de samme vurderingene gjelder her som for OOXML.

Under finnes en tabell som viser hvilke plattformer de viktigste verktøyene som kan skrive ODF finnes på.

Name	Mac	Windows	Linux	Kommentar
AbiWord	x	x	x	
Go-oo	x	x	x	
Google Docs	x	x	x	
IBM Lotus Symphony	x	x	x	
KOffice	?	x	x	
Microsoft Office		x		Mac-utgaven støtter ikke ODF.
NeoOffice	x			
OpenOffice.org	x	x	x	

Det finnes altså betydelig flere relevante implementasjoner av ODF enn av OOXML, og ingen av plattformene har mindre enn 6 implementasjoner. I realiteten er fire av disse (OpenOffice, Go-oo, NeoOffice og Symphony) basert på langt på vei den samme koden, men selv om man regner dette som bare ett verktøy er det altså fortsatt minst 3 forskjellige alternativer på alle plattformer. I utgangspunktet fremstår dermed verktøystøtten for ODF som meget god.

Som det fremgår av listen under finnes også mange av disse verktøyene på norsk:

- **AbiWord** har 91% og 93% komplette oversettelser til nynorsk og bokmål, som nok kan fullføres om det skulle være behov for det.
- **Google Docs** finnes kun på bokmål.
- **IBM Lotus Symphony** finnes kun på "Norwegian"; litt uklart hva det egentlig betyr.
- **KOffice**: har oversettelse til både bokmål og nynorsk.
- **OpenOffice** finnes på både bokmål og nynorsk.
- **Microsoft Office** finnes på både bokmål og nynorsk.
- **NeoOffice** finnes på både bokmål og nynorsk.

#### 4.2.1 1 eller 4 verktøy?

Et viktig spørsmål når man skal vurdere verktøystøtten for ODF er om OpenOffice.org og de fire verktøyene basert på det skal vurderes som fire separate verktøy eller som fire forskjellige versjoner av ett og samme verktøy. Alle fire skriver seg opprinnelig fra den samme kodebasen, og for brukeren fremstår de som svært like. De har f.eks den samme menystrukturen og stort sett den samme funksjonaliteten.

I og med at OpenOffice.org er grunnlaget for alle fire bruker vi den som utgangspunkt, og vurderer de andre tre opp mot den, for å se i hvilken grad de virkelig kan betraktes som egne verktøy.

Go-oo installerer seg selv som OpenOffice.org, og slik sett gir den seg altså ut for å være OpenOffice ved installasjon. Den beskrives som "a concentrated set of patches for ... OpenOffice.org" [WikiGo-oo]. Det beskrives med andre ord ikke som et selvstendig verktøy, men som en samling endringer til OpenOffice.org, der mange av disse etterhvert vil bli en del av OpenOffice. Den viktigste forskjellen ser ut til å være støtten for

OOXML i Go-oo. For brukeren fremstår programmene som noe bortimot identiske. Go-oo fremstår dermed som en versjon av OpenOffice, og ikke som et eget verktøy.

NeoOffice beskriver seg selv som "a free software port of OpenOffice.org to the MacOS X platform" [NeoOffice]. Med andre ord, som en oversettelse av OpenOffice til en annen plattform. Wikipedia beskriver den som "a fork of the free/open source OpenOffice.org that implements nearly all of the features of the corresponding OpenOffice.org version" [WikiNeoOffice]. For brukeren fremstår NeoOffice som svært lik OpenOffice.org i brukergrensesnitt og funksjonalitet. Versjon 3.0 var basert på OpenOffice 3.0.1 [NeoOffice2009], mens 3.0.2 er basert på OpenOffice 3.2 [WikiNeoOffice]. Det er med andre ord tydelig at NeoOffice kontinuerlig henter inn ny kode fra OpenOffice og tilpasser denne til MacOS X-plattformen, slik at det her er snakk om en oversettelse (en versjon), mer enn om et selvstendig verktøy.

IBM Lotus Symphony bruker IBM-rammeverk for brukergrensesnittet, mens kjernefunksjonaliteten er basert på OpenOffice.org 1.1.4. Denne versjonen var tilgjengelig under både GPL- og SISSL-lisens, slik at IBM ikke behøver å gjøre sin kode åpent tilgjengelig. Siden den gang har IBM utviklet koden selvstendig [WikiSymphony]. Dette er hva åpen kildekode-miljøet kaller en "fork", et nytt og uavhengig produkt som har blitt laget med utgangspunkt i en bestemt versjon av et annet produkt, men som siden utvikles uavhengig. Foreløpig siste stabile versjon er 1.3.

IBM har annonsert at Symphony 3.0 vil inkludere kode for ny funksjonalitet fra OpenOffice 3.0 samt nye moduler utviklet av IBM [IBM2008]. Dette bryter med den tidligere strikte uavhengigheten etter splitten ("the fork") fra OpenOffice, men samtidig er det tydelig at dette er et uavhengig produkt som kun deler noe kode med OpenOffice. For brukeren fremstår Symphony som den klart mest ulike av disse fire produktene, selv om menystrukturen fortsatt er nesten den samme. I sum må nok Symphony vurderes som et eget produkt.

**Vår konklusjon er** at det her er snakk om to produkter: OpenOffice.org og IBM Lotus Symphony, der førstnevnte også kommer i to andre versjoner: Go-oo, med litt nyere/mer funksjonalitet, og NeoOffice, tilpasset MacOS X.

Til tross for denne vurderingen har vi valgt å teste Go-oo, NeoOffice og Symphony separat. Vi har valgt Go-oo som vår OpenOffice-versjon for å få med støtten for å skrive OOXML.

#### 4.2.2 Grunnleggende formatering

Et dokument helt likt det beskrevet i kapittel 4.1.1 side 23 ble laget i OpenOffice.org og så åpnet i andre verktøy. Noen endringer ble gjort i disse verktøyene, dokumentet lagret, og resultatet åpnet i OpenOffice.org igjen.

I AbiWord for Mac overlappet ordene i vanlige avsnitt hverandre, lister på flere nivåer flatet ut til ett nivå, bakgrunnsfarge forsvant, bitmap-bilder forsvant og vektorbilder forsvant. AbiWord for Mac kan med andre ord ikke brukes med ODF-dokumenter. I AbiWord for Windows gikk det langt bedre, men listepunktene ble feil, formateringen av nummererte lister ble feil, vektorgrafikken ble ødelagt og havnet på feil sted og tabellen ble plassert oppå footer. Dermed kan heller ikke AbiWord for Windows få ståkarakter.

NeoOffice åpnet dokumentet helt perfekt, og endringer ble helt perfekt med tilbake. I IBM Lotus Symphony flatet lister på flere nivåer ut til ett, men ellers fant vi ingen feil.

I Apple TextEdit ble noe av formateringen borte, men dokumentet var fullt ut leselig. Fotnoter, header og footer, samt vektor- og bitmap-bilder forsvant.

I Google Docs ble innholdsfortegnelsen til vanlig tekst, punktlisten flatet ut til ett nivå, header og footer ble til vanlig tekst, og vektorgrafikken forsvant. Da vi prøvde å rette opp dokumentet viste det seg at man ikke kan tegne vektorgrafikk i Google Docs. Tilbake i OpenOffice.org ble innholdsfortegnelsen og header/footer til vanlig tekst igjen, og punktlisten ble rotet til.

I Microsoft Office 2007 SP2 så både punktlistene og nummererte lister flate ut rent visuelt, selv om den dypere strukturen faktisk var til stede. I tillegg ble tekstformateringen i vektorgrafikken ødelagt. Ellers åpnet Office dokumentet helt korrekt. Tilbake i OpenOffice ble punktene i ytterste nivå av punktlisten til stygge hakesymboler, men ellers så alt korrekt ut.

**Vår konklusjon er** at for verktøyene basert på OpenOffice.org fungerer interoperabiliteten svært bra, mens for andre verktøy er det betydelige problemer. Unntaket er noe overraskende Microsoft Word, som ikke var så verst.

#### 4.2.3 Endringssporing

Dette ble testet ved å ta et eksisterende dokument og gjøre endringer i det med endringssporing i OpenOffice.org. Dokumentet ble så lagret og åpnet i verktøyet som skulle testes. Deretter ble det gjort endringer i dette verktøyet også, og dokumentet lagret og åpnet igjen i OpenOffice.org.

Vi fant ingen problemer med utveksling i OpenOffice.org, NeoOffice, Go-oo, og IBM Lotus Symphony. Det viste seg dog fort at OpenOffice ikke støtter endringssporing i tabeller og lister i det hele tatt, hvilket betyr at man kan diskutere om støtten for endringssporing i ODF i det hele tatt er god nok. Det er også verdt å merke seg at alle disse fire er basert på kildekoden fra OpenOffice.org. Et grundigere studium av dette viser også at det er store problemer med utveksling av endringssporing mellom ODF-implementasjoner [Mahugh2009b].

AbiWord for Mac kunne ikke åpne helt vanlige ODF-dokumenter, og kan derfor ikke regnes med. AbiWord for Windows plukket ikke opp endringssporingen, og viste både gammel og ny tekst sammenblandet. Apple TextEdit støtter ikke sporing av endringer, og kan derfor heller ikke regnes med. Office 2007 støtter ikke endringssporing i ODF [Mahugh2009b].

**Vår konklusjon er** at dette kun fungerer i de fire verktøyene basert på OpenOffice, og at selv disse har begrensninger.

#### 4.2.4 Kommentarer

Vi gjorde en liten test av dette ved å lage et lite dokument i OpenOffice.org med kommentarer i løpende tekst, lister, overskrifter og tabeller, med litt enkel formatering i kommentarene. Dette ble så åpnet i andre verktøy. Hvis dette fungerte ble kommentarene endret og noen lagt til, og det hele lagret igjen for å bli åpnet i OpenOffice.org igjen.

I NeoOffice 3.0.1 patch 0, IBM Lotus Symphony og Microsoft Office 2007 SP2 fungerte dette helt prikkfritt, bortsett fra at formatering i kommentarene gikk tapt. Google Docs åpnet kommentarene midt i teksten, som en del av denne og ikke som kommentarer, og støtter derfor åpenbart ikke kommentarer i det hele tatt. Apple TextEdit fjernet kommentarene uten å gjøre skade på dokumentet.

I AbiWord for Mac ble kommentarene satt inn midt i teksten, mens i AbiWord for Windows dukket det opp nummererte markører for hver kommentar, mens selve kommentarene sto nederst på siden. Informasjon om hvem som satte inn kommentarene og når gikk tapt, men formateringen ble bevart. Dette er antagelig slik det er tiltenkt, og selv om det kanskje ikke er optimalt grensesnitt må det vel sies å fungere. Med litt romslighet kan nok dette aksepteres.

**Vår konklusjon er** at utveksling av kommentarer fungerer fint i verktøy basert på OpenOffice samt i Microsoft Office og AbiWord. Det feiler dog fullstendig i alle andre verktøy vi har testet.

#### 4.2.5 Formler i regneark

En enkel test med det samme regnearket som ble brukt til å teste OOXML ble gjort for ODF, med OpenOffice.org som utgangspunkt.

Gnumeric for Mac brukte veldig lang tid på å åpne regnearket, og feilet til slutt med meldingen "Sheet2! A1048576 : Content past the maximum number of rows supported in this build (65536). Please recompile with larger limits". Deretter åpnet den regnearket tilsynelatende korrekt, men det lot seg ikke redigere. I Gnumeric for Windows fungerte det helt perfekt.

Google Docs klarte ikke referansen til en annen arbeidsbok i samme regneark, og viste 0.5 avrundet til 1, men klarte ellers alt annet. Tables klarte ikke å åpne regnearket. I NeoOffice og IBM Lotus Symphony fungerte det helt prikkfritt.

I Microsoft Office forsvant alle formlene, og ble erstattet av de utregnede tallene. Det vil si: en celle som i OpenOffice inneholdt `=sum(A2:A16)` inneholdt i Office 120, som er den korrekte summen. Ellers åpnet regnearket helt perfekt. På vei tilbake til OpenOffice ble alle formler til tekststrenger av format `msoxl1:=sum(A2:A16)`, som betyr at Office har lagret formlene i sitt eget formelspråk og så brukt et navneromsprefiks, slik ODF 1.1 teknisk sett tillater.

**Vår konklusjon er** at dette fungerer i OpenOffice.org-avleggerene NeoOffice og IBM Lotus Symphony, og at det ser ut til å fungere Gnumeric for Windows. Alle andre verktøy feilet alvorlig i vår test.

#### 4.2.6 Skjemaer

Vi testet dette ved å lage et enkelt skjema i OpenOffice med noen forskjellige typer felter, og laget en enkelt seksjon som inneholdt hele skjemaet, og denne seksjonen ble låst for endringer. Dokumentet ble så åpnet i andre verktøy, utfyllt, og åpnet igjen i OpenOffice.

AbiWord mistet skjemafeltene, slik at dette fungerte ikke der. I NeoOffice og IBM Lotus Symphony fungerte det helt prikkfritt. I Microsoft Office fikk vi feilmeldingen "Dokumentet er passordbeskyttet og kan ikke åpnes", slik at dokumentet ikke lot seg åpne i det hele tatt.

**Vår konklusjon er** er dermed den "vanlige": det fungerer i de fire verktøyene basert på OpenOffice.org, men ellers ikke.

### 4.3 Oppsummering

Interoperabilitet mellom verktøy er ikke enkelt å teste, og å bevise at det fungerer er svært krevende, siden selv om alle tester fungerer er det fortsatt mulig at neste test feiler. I dette tilfellet valgte vi å legge oss på en svært overfladisk linje, og kun gjøre noen svært enkle tester for hvert funksjonsområde. Med et så enkelt og begrenset materiale burde man forvente at dersom interoperabiliteten er god burde det ikke forekomme feil i det hele tatt.

I stedet fant vi at interoperabilitet knapt fungerer på tvers av verktøy i det hele tatt. Gitt hvor enkle testene er kan dette knapt bety noe annet enn at interoperabilitet på tvers av verktøy fungerer svært dårlig i dag. Resultatene er oppsummert i tabellen under.

Funksjonalitet	OOXML	ODF	Kommentar
Formatering	Kun MS Office	2 verktøy OK	
Endringssporing	2 verktøy OK	2 verktøy OK	
Kommentarer	Kun MS Office	4 verktøy OK	
Formler regneark	4 verktøy OK	3 verktøy OK	
Skjemaer	Kun MS Office	2 verktøy OK	

Dette er svært nedslående, ettersom det betyr at for området redigerbare dokumenter har standardiseringen av formater fortsatt ikke gitt det resultatet som burde være hele formålet med standardiseringen: interoperabilitet på tvers av verktøy.

Satt litt på spissen kan man si at for OOXML er man i praksis henvist til å kun bruke Microsoft Office, mens for ODF kan man velge mellom OpenOffice.org og IBM Lotus Symphony.

## 5 Anvendelsesområdene

Oppdragsgiver har ønsket å få en vurdering av de to standardene innen fire forskjellige anvendelsesområder, beskrevet i hvert sitt kapittel under. Felles for alle anvendelsesområdene er at bruk av standardene kun er relevant der dokumenter utveksles med aktører utenfor etaten. Hvilke formater en etat bruker internt er opp til den enkelte etat.

## 5.1 Publisering på offentlige nettsider

Dette anvendelsesområdet heter i referansekatalogen "Publisering av tekstdokumenter på offentlige nettsider", og er beskrevet slik:

Anvendelsesområdet som her omtales er publisering av tekstdokumenter på statlige nettsider. Et dokument er i denne sammenheng definert som en lesbar meddelelse som inneholder informasjon. Elektronisk utveksling av dokumenter på annet vis (e-post og lignende), internt i en statlig virksomhet, mellom offentlige virksomheter og med innbyggere og næringsliv omfattes ikke av dette pålegget.

Vi har tolket dette dithen at det her er snakk om redigerbare dokumenter som legges ut på et offentlig nettsted, med tanke på at mottager siden skal redigere dette på ett eller annet vis. Antagelig (men ikke nødvendigvis) skal det redigerte dokumentet siden sendes til en offentlig etat.

Det scenariet som kanskje kan synes som det mest opplagte for dette anvendelsesområdet er publisering av vanlige dokumenter, som f.eks utlysning av anbud, oversikter over åpningstider, regelverk osv. Dette er dog ikke dokumenter som skal redigeres av mottaker, slik at i disse tilfellene er det i henhold til referansekatalogen HTML og PDF som skal brukes, og ikke redigerbare formater som ODF og OOXML. Dette er dermed ikke et relevant scenario, og det vil derfor ikke bli vurdert her.

Publisering av dokumenter på offentlige intranett anser vi at faller utenfor dette anvendelsesområdet.

### 5.1.1 Scenario #1: Skjemaer på nett

I noen tilfeller vil offentlige etater ønske å legge ut skjemaer av ulike slag som dokumenter på nett slik at brukere kan fylle dem ut elektronisk og sende dem inn igjen elektronisk via epost. Dersom skjemaene skal skrives ut og sendes inn på papir er det igjen PDF som er foretrukket format, slik at vi vurderer kun elektronisk innsending her.

Dokumentet vil her være produsert manuelt i et vanlig kontorverktøy, mest sannsynlig med bruk av vanlig formatering, og i noen tilfeller også skjemafunksjonalitet og kanskje med dokumentet låst, slik at mottager kun kan fylle ut de definerte feltene. Mottaker må kunne åpne dokumentet, fylle inn feltene, lagre og sende tilbake.

OOXML-standarden beskriver denne funksjonaliteten helt greit, men utveksling mellom verktøy av dokumenter med skjemaer og beskyttelse fungerte dårlig.

ODF-standarden har god støtte for skjemaer, men i ODF 1.1 er støtten for beskyttelse av dokumentet dårlig, slik at verdien av å bruke skjemaer er kraftig redusert. Utveksling mellom verktøy fungerte for verktøyene basert på OpenOffice, men ikke andre.

### 5.1.2 Scenario #2: Distribusjon av maler

Offentlige etater publiserer maler til dokumenter av ulike slag på offentlige nettsteder. Disse kan lastes ned av både offentlige og private brukere som utgangspunkt for å skrive dokumenter av ymse typer (se appendiks 7 side 37).

Her vil malen være produsert av avsender i en vanlig kontorpakke, som altså må ha støtte for å lagre i formatet. Tilsvarende må mottakers kontorpakke ha støtte for å åpne malen og bruke den som utgangspunkt for videre redigering av dokumentet. Avsender må kunne lagre i formatet, mens mottaker trenger altså ikke å kunne dette.

Merk at selv om OOXML og ODF har egen støtte for maler, det vil si dokumenter som inneholder en grafisk profil, men svært lite innhold, så er det altså ikke denne funksjonaliteten det er snakk om her. Når offentlige etater legger ut maler er det typisk snakk om å kommunisere strukturen på dokumentet og hva slags informasjon som skal være med, snarere enn å formidle et bestemt grafisk uttrykk.

Funksjonaliteten man trenger for denne anvendelsen er dermed kun helt grunnleggende formatering. I noen tilfeller er det snakk om skjemaer, og i noen tilfeller også regneark med formler.

OOXML-standarden er ut til å spesifisere den nødvendige funksjonaliteten helt greit, men interoperabilitet mellom verktøyene er ikke helt god nok i dag. Støtte for grunnleggende formatering, og også for skjemaer og dokumentbeskyttelse, er ikke helt tilstrekkelig for denne anvendelsen.

For ODF er selve spesifikasjonen noe svakere, men stort sett tilstrekkelig for denne anvendelsen, unntatt for regneark, der den har grove mangler. Interoperabiliteten er stort sett god mellom verktøyene basert på OpenOffice, men ellers svak.

### 5.1.3 Scenario #3: Standardkontrakter

I mange tilfeller publiserer offentlige etater forslag til kontrakter for forskjellige formål på offentlige nettsteder. Disse publiseres som redigerbare dokumenter fordi det skal være mulig for mottaker å redigere dem. I noen tilfeller vil redigering kun innebære å fylle inn markerte felter (som i scenario #1), mens i andre tilfeller vil det være mulig å redigere selve teksten for å tilpasse den til det enkelte tilfelle.

En rask gjennomgang av offentlige standardkontrakter på nett (blant annet på anskaffelser.no, regjeringen.no, oslo.kommune.no og uio.no) viser at dette stort sett er helt vanlige dokumenter med svært enkel formatering. I noen tilfeller brukes også skjemafunksjonalitet. Se appendiks 7 side 37 for eksempler på slike dokumenter på nett.

Her vil altså typisk mottager laste ned dokumentet, åpne det, og deretter redigere det videre selv. Den eneste funksjonaliteten som er påkrevd er dermed at mottaker har et verktøy som kan lese formatet, og at dette støtter grunnleggende formatering. Støtte for skjemaer og beskyttelse av dokumenter er også en fordel.

Siden funksjonaliteten som trengs er den samme som i scenario #2 blir også vurderingene de samme, bortsett fra at i dette scenariet er det ikke behov for regneark.

### 5.1.4 Anbefalinger

Observasjonene fra de tre scenariene for dette anvendelsesområdet er oppsummert i tabellen under. Verktøy-kolonnene teller hvor mange verktøy vi fikk identifisert funksjonalitet til å fungere i. Microsoft Office og OpenOffice telles med, slik at tallet kommer aldri under 1.

Scenario	OOXML spec	OOXML verktøy	ODF spec	ODF verktøy
#1 Skjemaer på nett	Godt nok	1	Ikke godt nok	2
#2 Maler på nett	Bra	1	Ikke godt nok	2
#3 Kontrakter	Bra	1	Godt nok	2

For dette anvendelsesområdet kommer altså ECMA-376:2006 godt ut som spesifisering betraktet, men interoperabilitet mellom verktøyene er så dårlig at man i praksis kun kan stole på ett verktøy: Microsoft Office.

For ODF er situasjonen den at ODF 1.1-spesifikasjonen ikke helt er god nok, og det er kun to verktøy som ser ut til å ha pålitelig utveksling: OpenOffice.org og IBM Lotus Symphony.

I dagens versjon av referansekatalogen [Referansekatalogen] er ODF 1.1 obligatorisk for dette anvendelsesområdet, mens OOXML er under observasjon. Gitt vurderingene over er det likevel vanskelig å se hvordan vi skal kunne anbefale OOXML eller ODF som standarder i offentlig forvaltning på dette anvendelsesområdet.

**Vår anbefaling:** vi anbefaler at referansekatalogen fører opp både ECMA-376:2006 og ODF 1.1 som "under observasjon". Se kapittel 6.1 side 34 for mer bakgrunn for dette valget.

## 5.2 Utveksling offentlig/privat via epost

Dette anvendelsesområdet er heter i referansekatalogen "Utteksling av dokumenter som vedlegg til e-post mellom offentlig virksomheter og innbyggere/næringsliv" og er beskrevet slik:

Anvendelsesområdet som her omtales er utveksling av dokumenter som vedlegg til e-post mellom offentlige virksomheter og innbyggere/næringsliv. Et dokument i denne sammenhengen tradisjonelle tekstbehandlingsdokumenter hvor det kan forekomme enkle tabeller, bilder og figurer, men ikke lyd, video eller regneark. Elektronisk utveksling av dokumenter på annet vis (utfylte webskjema på web, maskin til maskin og lignende), internt i en offentlig virksomhet, mellom offentlige virksomheter og med innbyggere og næringsliv omfattes ikke av dette pålegget.

Igjen gjør vi den presiseringen at det må dreie seg om redigerbare dokumenter som skal redigeres av mottaker og (kanskje) sendes tilbake.

### 5.2.1 Scenario #4: Utfylling av skjema

I noen tilfeller der privatpersoner eller firmaer skal fylle ut skjemaer for offentlige etater sender man ut skjemaene som redigerbare dokumenter via epost. Som i scenario #1 (kapittel 5.1.1 side 30) gjelder dette kun når skjemaet skal utfylles elektronisk, siden det ellers vil være PDF som er foretrukket format.

Dette scenariet er egentlig helt likt scenario #1, unntatt at transporten av dokumentet fra avsender til mottager går over epost i stedet for via web. Produksjon av skjemaet på offentlig side, utfylling av skjemaet, og innsending av det vil være helt likt, slik at vurderingene fra scenario #1 gjelder også her.

### 5.2.2 Scenario #5: Utarbeiding av kontrakt

Når en offentlig etat går til innskaffelse av en rapport, et IT-system, eller lignende gjennomføres ofte en anbudsrunde først. Etter at denne er avsluttet inngås typisk en kontrakt med den foretrukne leverandøren, og her vil det typisk være behov for at partene sammen forfatter kontrakten. I disse tilfellene vil man typisk sende stadig nye utkast til kontrakt frem og tilbake på epost, med utstrakt bruk av sporing av endringer av kommentarer.

Dette scenariet er i praksis helt likt scenario #6, som er beskrevet i kapittel 5.3.1 side 33 og leseren henvises dit for en vurdering.

### 5.2.3 Anbefalinger

Tabellen under gir en kort oppsummering av de to scenariene for dette anvendelsesområdet:

Scenario	OOXML spec	OOXML verktøy	ODF spec	ODF verktøy
#4 Skjemaer på epost	Godt nok	1	Ikke godt nok	2
#5 Kontraktsskriving	Bra	1	Ikke godt nok	2

I dagens versjon av referansekatalogen [Referansekatalogen] skal følgende innføres fra 1. januar 2011: ODF 1.1 blir obligatorisk for dette anvendelsesområdet, mens OOXML er under observasjon. Likevel må konklusjonen her bli den samme som for det forrige anvendelsesområdet: det fremstår som vanskelig å anbefale standardene for dette anvendelsesområdet.

**Vår anbefaling:** vi anbefaler at referansekatalogen fører opp både ECMA-376:2006 og ODF 1.1 som "under observasjon". Se kapittel 6.1 side 34 for mer bakgrunn for dette valget.



### 5.3 Utveksling offentlig/offentlig via epost

Dette anvendelsesområdet er beskrevet som "Utteksling av dokumenter som vedlegg til e-post mellom offentlige virksomheter" i konkurransegrunnlaget. Igjen er det altså snakk om dokumenter som skal redigeres av mottaker.

#### 5.3.1 Scenario #6: Samarbeidende skrijving på tvers av etat

Dokumenter skrives ofte av en gruppe forfattere fra flere ulike etater, slik at det blir behov for å sende nye utkast og bidrag til dokumentet i epost mellom forfatterne. Her vil alle medlemmer av gruppen vil ha behov for å kunne lese mottatte dokumenter med korrekt formatering, og de fleste vil også ha behov for å kunne redigere dem. Videre vil det være behov for at funksjoner som å spore og vurdere endringer og å legge inn kommentarer vil være helt avgjørende for at dette skal fungere optimalt.

Studier av spesifikasjonene tyder på at OOXML har alle forutsetninger for at dette skal fungere, mens ODF i mindre grad har det, siden spesifikasjonen er svak på dette punktet. Enkle tester av implementasjoner viser dog at denne funksjonaliteten fungerer dårlig på tvers av OOXML-implementasjoner, mens for ODF-implementasjoner fungerer det mellom de fire OpenOffice-baserte verktøyene.

Konklusjonen må dermed bli at verktøystøtten for begge formater i dag er svak, men at den er best for ODF.

#### 5.3.2 Scenario #7: Utveksling av budsjettdata

I arbeidet med å sette sammen årets budsjettforslag vil typisk underliggende etater få tilsendt regneark med beskjed om å fylle inn de relevante tallene for etaten, og så å sende dette tilbake til overliggende etat. Formen på slike regneark varierer veldig, fra helt enkle to-kolonne regneark (regnskapspost, budsjettanslag) med en sum nederst, til avanserte regneark med flere arbeidsbøker, formler og referanser, og i noen tilfeller også makroer som sikrer at regnearket blir mest mulig korrekt utfyllt.

Variasjonen i form på slike regneark gjør det vanskelig å være presis på hva slags funksjonalitet som er påkrevet, men layout, tall og tekst må åpenbart bevares. Korrekt tolkning og representasjon av enklere formler må også bevares. I tillegg er det altså et behov for makroprogrammering som bør ivaretas.

Her er det åpenbare svakheter i ODF-spesifikasjonen, som altså ikke definerer formel- og makro-språk i det hele tatt. OOXML-spesifikasjonen har definert formelspråket grundig, men har altså heller ikke spesifisert noe makrospråk. I testing fant vi at enkle formler i ODF-regneark kun kunne utveksles mellom de fire OpenOffice-verktøyene, mens i OOXML fungerte det mellom de verktøyene (4) som støtter formatet.

I og med at ODF 1.1 ikke spesifiserer noe format for formler i regneark, og siden testing viser at dette heller ikke fungerer i praksis, kan dette formatet kort og godt ikke anbefales for bruk i dette scenariet. For OOXML finnes det et akseptabelt antall implementasjoner med grei interoperabilitet, slik at OOXML kan faktisk anbefales. Det må dog bemerkes at det ikke er noen grunn til å tro at makroer vil kunne utveksles, slik at for avanserte regneark vil man være avhengig av at avsender og mottager bruker samme verktøy.

#### 5.3.3 Anbefalinger

Vurderingene av de to scenariene for dette anvendelsesområdet er oppsummert under:

Scenario	OOXML spec	OOXML verktøy	ODF spec	ODF verktøy
#6 Samarbeidende skrijving	Bra	1	Ikke godt nok	2
#7 Budsjettdata	Bra	4	Dårlig	2

Gitt at ODF 1.1 faktisk ikke definerer noe formelspråk for regneark i det hele tatt finner vi det umulig å anbefale ODF 1.1 for dette anvendelsesområdet, gitt at regneark fremstår som et viktig bruksområde her. Man kunne kanskje valgt å anbefale ODF 1.1 med en presisering av at dette ikke gjelder for regneark, men gitt at ODF heller ikke gjør det så bra på scenario #6 ser vi ingen grunn til å gjøre dette.

For OOXML er spesifikasjonen for så vidt god nok, og utveksling av regneark ser ut til å fungere godt, men for scenario #6 fungerer utveksling mellom verktøyene så dårlig at man ikke godt kan anbefale OOXML heller.

**Vår anbefaling:** vi anbefaler at referanse katalogen fører opp både ECMA-376:2006 og ODF 1.1 som "under observasjon". Se kapittel 6.1 side 34 for mer bakgrunn for dette valget.

## 5.4 Produksjonsformat

Dette anvendelsesområdet er beskrevet slik i konkurransegrunnlaget: "Produksjonsformat i offentlige virksomheter (Formatet som benyttes i kontorstøtteprogramvare og grensesnittet mellom kontorstøtteprogramvaren og fagsystemene)".

Vi jobber fortsatt med analysen av dette anvendelsesområdet og vil komme tilbake med mer innhold om dette senere.

## 6 Ytterligere vurderinger

Dette kapittelet samler ulike vurderinger rundt bruk av åpne standarder for redigerbare dokumenter i offentlig sektor som i noen grad går ut over en snever tolkning av oppdraget, men som likevel kan være relevante for oppdragsgiver.

### 6.1 Fremtidig utvikling

Det er selvsagt vanskelig å spå hvordan utviklingen vil arte seg videre innen dette området, men vi mener likevel at det er mulig å gjøre noen antagelser om hva som er sannsynlig.

Selv om ODF 1.1 har en lang rekke svakheter er det tydelig at ODF 1.2, når den blir ferdigstilt, vil utbedre svært mange av disse. Deriblant den kanskje mest alvorlige: mangelen på spesifisering av formler i regneark. Dermed er det all grunn til å tro at vurderingen av ODF 1.2 som spesifisering vil bli langt mer positiv enn hva tilfellet er for ODF 1.1. Forhåpentlig vil ODF 1.2 også bli mer stabil og trenge færre endringer enn ODF 1.1, men dette er selvfølgelig langt fra sikkert.

Hva angår implementasjoner av ODF har Microsoft i sin implementasjon nektet å basere seg på "reverse engineering" av oppførselen til OpenOffice, og heller valgt å implementere spesifiseringen direkte, og da kun de delene av den som ble vurdert som tilstrekkelig godt beskrevet til at de kan implementeres pålitelig. Gitt de store forbedringene som nok vil komme i ODF 1.2 er det all grunn til å tro at hvis Microsoft skulle implementere ODF 1.2 (hvilket de sannsynligvis vil gjøre) så vil denne implementasjonen ha langt bedre interoperabilitet med de andre verktøyene.

For Google Docs, som også har basert sin implementasjon på ODF 1.1, er situasjonen mer uklar, men det er i hvert fall ingen grunn til å tro at implementasjonen vil bli noe dårligere. Det samme gjelder AbiWord og KOffice.

Med andre ord er det all grunn til å tro at verktøysituasjonen for ODF vil bli bedre etter at ODF 1.2 blir ferdig, selv om dette fort kan ta et år eller to.

For OOXML er situasjonen mye den samme. I ISO jobbes det med å rydde opp og rette i standarden, slik at det er all grunn til å tro at fremtidige versjoner vil ha høyere kvalitet, med færre feil og mindre duplisert funksjonalitet. Sannsynligvis vil de også være mer stabile etterhvert som ISO får jobbet seg gjennom de kjente problemene med spesifiseringen.

Dersom OOXML Strict implementeres i neste versjon av Office (slik Microsoft har lovet) er det all grunn til å tro at dette også vil ha positiv innvirkning på andre verktøy. Disse vil da få et mindre og enklere format å implementere, samtidig som man har fått mer tid på seg til å forbedre konverteringsfunksjonene i eksisterende programvare. Kanskje vil også flere kontorpakker, som f.eks Apples iWork, begynne å støtte lagring i formatet.

Det er altså all grunn til å tro at verktøystøtten for OOXML vil bli bedre når ISO får ferdigstilt amendment 1 til ISO 29500.

Gitt at situasjonen for begge formater, hva angår både spesifisering og programvare, sannsynligvis vil bli bedre i årene fremover anbefaler vi at begge formatene settes under observasjon. Mest sannsynlig vil de versjonene det er mest interessant å vurdere i fremtiden for inkludering i referansekatalogen være ODF 1.2 og ISO 29500 amendment 1, men dette må vente til spesifiseringene av disse faktisk er ferdig. Det bør også vente til implementasjonene begynner å modne.

## 6.2 Konsekvenser av anbefalingene

Det er vanskelig å si med sikkerhet hva konsekvensene vil bli dersom disse anbefalingene tas til følge, men vi føler at det er bryet verdt å skissere opp det vi betrakter som de mest sannsynlige scenariene.

Det siste tiåret har offentlig sektor nesten utelukkende brukt kontorpakken Microsoft Office, og denne har vært tett integrert i mange av fagsystemene i offentlig sektor.

Etter at Standardiseringsrådet gjorde ODF til obligatorisk standard for redigerbare dokumenter innen to anvendelsesområder uten noe alternativ har vi sett en utvikling der:

- Noen få offentlige etater har begynt overgang fra Microsoft Office til OpenOffice.
- Stadig færre offentlige etater publiserer dokumenter på web i Word-format. I stedet har man begynt å publisere disse først og fremst som PDF (i tråd med referansekatalogen anvisninger), og i noen få tilfeller i ODF.
- Noen leverandører av fagsystemer har integrert disse også med OpenOffice.

Det er grunn til å tro at dersom referansekatalogen endres slik at det blir fritt valg av format for redigerbare dokumenter, så vil denne utviklingen bremses eller kanskje til og med stoppe helt opp. Konsekvensen vil mest sannsynlig være at norsk offentlig sektor fortsetter å bruke Microsoft Office og fortsetter den tette bindingen til dette produktet.

Samtidig er det slik at Microsoft Office nå kan lese og skrive ODF, slik at dersom man opprettholder dagens retningslinjer med ODF som eneste obligatoriske format vil ikke det nødvendigvis bety at offentlig sektor dermed opplever noe stort press i retning av å forlate Microsoft Office.

Likevel vil det nok gi uttrykk for en generell holdning fra sentralt hold om at man ønsker å se mer bruk av andre verktøy. I hvilken grad de ulike etatene velger å ta dette til følge vil bli opp til dem.

## 6.3 Om redigerbare dokumenter og standarder

Standarder for redigerbare dokumenter har vært gjenstand for en voldsom interesse og oppmerksomhet i media og andre steder, men paradoksalt nok ser det ut til at standarder på dette området er langt mindre sentrale enn det man kan få inntrykk av. Denne rapporten inneholder et ganske begrenset antall scenarier fordi i de aller fleste tilfeller der dokumenter skal utveksles, på nett, i epost, eller på andre måter, er det PDF eller papir som blir utvekslet.

Dette skyldes ganske enkelt at referansekatalogen krever at man bruker HTML og PDF til dokumenter som ikke skal endres av mottaker, og i de aller fleste tilfeller skal ikke mottaker endre dokumentet. Dermed blir det relativt få anvendelser igjen for OOXML og ODF som er relevante for referansekatalogen. Samtidig er det selvsagt slik at de fleste PDF-dokumenter produseres fra et OOXML- eller ODF-dokument laget i en kontorpakke, men referansekatalogen involverer seg ikke i hvordan PDF-dokumentene blir produsert.

Gitt hvor lite sentralt utveksling av redigerbare dokumenter med eksterne aktører ser ut til å være kan man spørre seg om referansekatalogen egentlig bør sette noen krav her i det hele tatt. Hvis man ser på situasjonen internasjonalt (kapittel 2.4 side 7), der svært få land har satt strenge krav på dette området, blir det enda større grunn til å stille spørsmålsteget ved dette.

Samtidig ser man at de sannsynlige konsekvensene av å ikke stille noen krav er tett binding i hele offentlig sektor til én enkelt leverandør (se kapittel 6.2 side 35). Hvorvidt det er riktig å ta kostnadene på kort sikt for å forsøke å unngå dette er til syvende og sist et politisk spørsmål som må avgjøres politisk.

## Appendiks A. Publiserte redigerbare dokumenter

Dette appendikset inneholder noen lenker til eksempler på redigerbare dokumenter publisert av offentlige etater på nett, som beskrevet i scenario #1 og #2.

Dokumentmaler:

- Maler fra Konkursrådet publisert som ODF. Stort sett vanlige dokumenter, men noen skjemaer og noen regneark.
- Maler fra NHO i Word-format. Helt vanlige dokumenter. (Strengt tatt ikke en offentlig etat, men likevel relevant som eksempler på behovet.)
- Maler fra Datatilsynet i ODF- og Word-format. Helt vanlige dokumenter, noen er skjemaer.
- Maler fra Universitetet i Bergen i Word-format. Helt vanlige dokumenter.
- Maler fra Plan- og bygningsetaten i Oslo Kommune i Word-format.. Helt vanlige dokumenter.

De fleste standardkontraktene vi fant var publisert i PDF-format, men følgende eksempler i redigerbare format ble også funnet:

- Kontrakter for ansatte ved Universitetet i Oslo. Helt vanlige dokumenter.
- Kontrakter for utlån eller utleie fra Kulturetaten i Oslo Kommune. Skjemaer.
- Standardkontrakter for oppdragsforskning fra Regjeringen.no. Vanlige redigerbare Word-dokumenter.

## Appendiks B. Verktøy brukt til testing

Testing er gjennomført med følgende verktøy:

- Microsoft Office 2008 for Mac 12.2.3 091001 og Microsoft Office 2007 for Windows SP2 MSO 12.0.6514.5000.
- Go-oo (OpenOffice.org-versjon) for Mac 3.1.1 build 9420.
- AbiWord 2.4.5 for Mac og 2.8.4 for Windows.
- NeoOffice 3.0.2 patch 0.
- IBM Lotus Symphony for Mac 1.3 20090908.0930.
- Apple iWork '09 2.0.3 (332).
- Apple TextEdit 1.6 (264).
- Tables 1.5.5.
- Gnumeric 1.8.1 for Mac og 1.10.2 for Windows.

KOffice burde strengt tatt også vært testet, men tross gjentatte forsøk klarte vi ikke å installere det på hverken Windows eller Mac. På grunn av tidsnød ble det aldri testet på Linux.

Microsoft Office 2010 for Windows er ikke testet, siden denne først ble tilgjengelig i det rapporten skulle ferdigstilles. Det er dog ikke grunn til å tro at resultatene ville blitt vesentlig annerledes med denne versjonen, i og med at Office helt bevisst har valgt å ikke implementere endringssporing og regnearkformler. Resultatene på grunnleggende formatering og skjemaer kunne dog blitt bedre.

## Bibliografi

**[Brown2008]**

*OOXML and Microsoft Office 2007 Conformance: a Smoke Test*, Alex Brown , 2008-04, available at <<http://www.adjb.net/post/OOXML-and-Office-2007-Conformance-a-Smoke-Test.aspx>>

**[Brown2010]**

*Microsoft Fails the Standards Test*, Alex Brown , 2010-03-31, available at <<http://www.adjb.net/post/Microsoft-Fails-the-Standards-Test.aspx>>

**[Fraunhofer]**

*Document Interoperability*, Open Document Format and Office Open XML, Dr. Klaus-Peter Eckert , Jan Henrik Ziesing , Ucheoma Ishionwu , Fraunhofer Verlag, 2009, ISBN 978-3-8396-0047-4, available at <[http://regmedia.co.uk/2009/08/20/fokus\\_odf\\_ooxml\\_report.pdf](http://regmedia.co.uk/2009/08/20/fokus_odf_ooxml_report.pdf)>

**[Hamilton2010]**

*OOXML Implementation: Can Expectations Ever Trump Reality?*, Dennis Hamilton , 2010-04-04, available at <<http://orcmid.com/blog/2010/04/ooxml-implementation-can-expectations.asp>>

**[Horton2009]**

*Why do we need serial dates in the Transitional form of IS 29500?*, Gareth Horton , 2009-10-22, available at <<http://aristippus303.wordpress.com/2009/10/22/why-do-we-need-serial-dates-in-the-transitional-form-of-is-29500/>>

**[IBM2008]**

*IBM Commits to Future of ODF With Symphony Roadmap*, Michael Azzi , IBM Press Room, 2008-11-05, available at <<http://www-03.ibm.com/press/us/en/pressrelease/25912.wss>>

**[Jelliffe2009]**

*Where everyone knows your name: ODF 1.1 formula support in Office SP2*, Rick Jelliffe , 2009-05-08, available at <<http://broadcast.oreilly.com/2009/05/odf-1-1-formula-support-in-office-sp2.html>>

**[Macnaghten2007]**

*Technical Distinctions of ODF and OOXML*, Edward Macnaghten ODF Alliance UK Action Group , Free Software Magazine, 2007-05-02, available at <<http://www.freesoftwaremagazine.com/node/2138/pdf>>

**[Mahugh2009]**

*1 + 2 = 1?*, Doug Mahugh , 2009-05-09, available at <<http://blogs.msdn.com/dmahugh/archive/2009/05/09/1-2-1.aspx>>

**[Mahugh2009b]**

*Tracked changes*, Doug Mahugh , 2009-05-13, available at <<http://blogs.msdn.com/dmahugh/archive/2009/05/13/tracked-changes.aspx>>

**[Mahugh2010]**

*Office's Support for ISO/IEC 29500 Strict*, Doug Mahugh , 2010-04-06, available at <<http://blogs.msdn.com/dmahugh/archive/2010/04/06/office-s-support-for-iso-iec-29500-strict.aspx>>

**[Microsoft2008]**

*Microsoft Expands List of Formats Supported in Microsoft Office*, Microsoft, 2008-05-21, available at <<http://www.microsoft.com/presspass/press/2008/may08/05-21ExpandedFormatsPR.msp>>

**[NeoOffice]**

*NeoOffice*, NeoOffice.org, 2010-04-07, available at <<http://www.neooffice.org/neojava/en/index.php>>

**[NeoOffice2009]**

*NeoOffice 3.0 released*, NeoOffice.org, 2009-03-30, available at <<https://trinity.neooffice.org/modules.php?name=News&file=article&sid=134>>

**[OODevGuide]**

*OpenOffice.org Developer's Guide, Scripting Framework*, OpenOffice.org, 2009-05-15, available at <[http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Scripting/Scripting\\_Framework](http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Scripting/Scripting_Framework)>

**[Office1277]**

*ODF still fails to specify scripting properly (ODF 1.2 CD01)*, 2009-12-13, OASIS, available at <<http://tools.oasis-open.org/issues/browse/OFFICE-1277>>

**[Office1836]**

*Public Comment: office:version (ODF all versions)*, 2009-07-30, OASIS, available at <<http://tools.oasis-open.org/issues/browse/OFFICE-1836>>

**[OpenFormula]**

*Open Document Format for Office Applications (OpenDocument) Version 1.2, Part 2: Recalculated Formula (OpenFormula) Format – Annotated Version*, David A. Wheeler , Patrick Durusau , Eike Rathke , Robert Weir , 2010-02-08, available at <<http://www.oasis-open.org/committees/download.php/36320/OpenDocument-part2-draft20-editor-revision.odt>>

**[OpenMalaysia2007a]**

*Malaysia's History is ill-formed*, yoonkit , 2007-06-14, available at <[http://www.openmalaysiablog.com/2007/06/malaysias\\_histo.html](http://www.openmalaysiablog.com/2007/06/malaysias_histo.html)>

**[OpenMalaysia2007b]**

*UOF and ODF comparison. Which should we choose?*, yoonkit , 2007-01-18, available at <[http://www.openmalaysiablog.com/2007/01/uof\\_and\\_odf\\_com.html](http://www.openmalaysiablog.com/2007/01/uof_and_odf_com.html)>

**[Oram2007]**

*A most ingenious paradox: make 1900 a leap year?*, Andy Oram , O'Reilly ONLamp.com, 2007-01-19, available at <[http://www.oreillynet.com/onlamp/blog/2007/01/a\\_most\\_ingenious\\_paradox\\_make.html](http://www.oreillynet.com/onlamp/blog/2007/01/a_most_ingenious_paradox_make.html)>

**[RGI 1.0]**

*Référentiel Général d'Interopérabilité RGI, Version 1.0*, Direction générale de la modernisation de l'État, 2009-05-12, available at <[http://www.references.modernisation.gouv.fr/sites/default/files/RGI\\_Version1%200\\_0.pdf](http://www.references.modernisation.gouv.fr/sites/default/files/RGI_Version1%200_0.pdf)>

**[RGI 1.0]**

*Référentiel Général d'Interopérabilité RGI, Version 1.0*, Direction générale de la modernisation de l'État, 2009-05-12, available at <[http://www.references.modernisation.gouv.fr/sites/default/files/RGI\\_Version1%200\\_0.pdf](http://www.references.modernisation.gouv.fr/sites/default/files/RGI_Version1%200_0.pdf)>

**[Referansekatalogen]**

*Referansekatalog for IT-standarder i offentlig sektor*, Difi, Versjon 2.0, 2009-06-25, available at <<http://www.regjeringen.no/nb/dep/fad/dok/rundskriv/2009/referansekatalogen.html?id=570673>>

**[SAGA 4.0]**

*Standards und Architekturen für E-Government-Anwendungen*, Version 4.0, Bundesministerium des Innern, 2008-03, available at <[http://www.cio.bund.de/cae/servlet/contentblob/77116/publicationFile/3995/saga\\_4\\_0\\_download.pdf](http://www.cio.bund.de/cae/servlet/contentblob/77116/publicationFile/3995/saga_4_0_download.pdf)>

**[SC34n1404]**

*Recommendations of ISO/IEC JTC 1/SC 34/WG 6 Meeting held in Stockholm, Sweden, 2010-03-25/26 and Project Subdivision Proposal for Amendment to ISO/IEC 26300: 2006, Open Document Format for Office Applications (OpenDocument) v1.0 [WG 6]*, ISO SC34, 2010-03-26, available at <<http://www.itscj.ipsj.or.jp/sc34/open/1404.htm>>



**[Smith2008]**

, James Smith , FutureGov, 2008-08-22, available at <<http://www.futuregov.net/articles/2008/aug/22/chinese-ministries-push-uof-document-format/>>

**[Stocholm2009]**

*Versioning of OOXML (thank you for all the fish)*, Jesper Lund Stocholm , 2009-02-06, available at <[http://idippedut.dk/post/2009/02/06/Versioning-of-OOXML-\(thank-you-for-all-the-fish\).aspx](http://idippedut.dk/post/2009/02/06/Versioning-of-OOXML-(thank-you-for-all-the-fish).aspx)>

**[TAC2004]**

*TAC approval on conclusions and recommendations on open document formats*, IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens), 2004-05-25, available at <<http://ec.europa.eu/idabc/en/document/2592/5588>>

**[VTU2010]**

*Konklusionspapir om anvendelsen af åbne standarder for software i det offentlige*, Ministeriet for Videnskab, Teknologi og Udvikling, 2010-01-29, available at <<http://vtu.dk/filer/aabne-standarder/bilag58.html>>

**[Valoris2003]**

*Comparative assessment of Open Documents Formats — Market Overview*, Valoris, 2003-12-30, 1.6, available at <<http://ec.europa.eu/idabc/servlets/Doc?id=17982>>

**[Weir2006]**

*A Leap Back*, Rob Weir , 2006-10-12, available at <<http://www.robweir.com/blog/2006/10/leap-back.html>>

**[Weir2007]**

*The Formula for Failure*, Rob Weir , 2007-07-09, available at <<http://www.robweir.com/blog/2007/07/formula-for-failure.html>>

**[Weir2008]**

*How many defects remain in OOXML?*, Rob Weir , 2008-03-18, available at <<http://www.robweir.com/blog/2008/03/how-many-defects-remain-in-ooxml.html>>

**[Weir2009a]**

*Update on ODF Spreadsheet Interoperability*, Rob Weir , 2009-05-03, available at <<http://www.robweir.com/blog/2009/05/update-on-odf-spreadsheet-interoperability.html>>

**[Weir2009b]**

*A follow-up on Excel 2007 SP2's ODF support*, Rob Weir , 2009-05-07, available at <<http://www.robweir.com/blog/2009/05/follow-up-on-excel-2007-sp2s-odf.html>>

**[WikiGo-oo]**

*Go-oo*, Wikipedia, 2010-03-22, available at <<http://en.wikipedia.org/wiki/Go-oo>>

**[WikiNeoOffice]**

*NeoOffice*, Wikipedia, 2010-02-15, available at <<http://en.wikipedia.org/wiki/NeoOffice>>

**[WikiODFAdoption]**

*OpenDocument adoption*, Wikipedia, 2010-04-05, available at <[http://en.wikipedia.org/wiki/OpenDocument\\_adoption](http://en.wikipedia.org/wiki/OpenDocument_adoption)>

**[WikiSymphony]**

*IBM Lotus Symphony*, Wikipedia, 2010-03-12, available at <[http://en.wikipedia.org/wiki/IBM\\_Lotus\\_Symphony](http://en.wikipedia.org/wiki/IBM_Lotus_Symphony)>

**[WikiUOF]**

*Uniform Office Format*, Wikipedia, 2009-07-11, available at <[http://ja.wikipedia.org/wiki/Uniform\\_Office\\_Format](http://ja.wikipedia.org/wiki/Uniform_Office_Format)>

**[Xinhua2009]**

*"China's version of" document format standard is expected to break Microsoft's monopoly,*  
Xinhua Li Hong, 2009-11-01, available at <[http://news.xinhuanet.com/politics/2009-11/01/content\\_12368558.htm](http://news.xinhuanet.com/politics/2009-11/01/content_12368558.htm)>