

Documentbeheer

Het kloppend hart van de organisatie

Afstudeerscriptie Technische Informatica

Roland Stoker

September 2009

Afd. Software Technology
Faculteit Electrotechniek, Wiskunde en Informatica
Technische Universiteit Delft

Auteur

Roland Stoker

Titel

Documentbeheer – Het kloppend hart van de organisatie

MSc presentatie

30 september 2009

Afstudeercommissie

prof. dr. ir. J.L.G. Dietz
dr. K. van der Meer
dr. ir. P. Wiggers

Technische Universiteit Delft
Technische Universiteit Delft
Technische Universiteit Delft

Inhoudsopgave

1	Introductie.....	1
1.1	Algemeen.....	1
1.2	Probleemstelling.....	1
1.3	Outline.....	2
2	Beschrijving van de opdrachtgever.....	3
3	Onderzoek opdracht via interviews/vergaderingen.....	5
3.1	Huidige software van Van Lee & Van Everdingen.....	5
3.2	Onderzoek werking archiefafdeling/postafdeling Den Briel.....	6
3.3	Document Structuur Plan.....	6
3.4	Preservatie en XML.....	6
4	Beschrijving van de opdracht.....	7
4.1	Algemeen.....	7
4.2	Programma van Eisen.....	8
5	Extra doelstellingen.....	9
6	Theorie.....	11
6.1	Records management (ISO 15489): Procedureanalyse voor een document in een organisatie.....	11
6.2	MoReq Classification Scheme.....	12
6.3	Theoretisch Model OAIS (ISO 14721).....	14
7	Bestaande Software.....	17
7.1	Document management systemen.....	17
7.2	Archiefsystemen.....	18
7.3	Postkamerverwerkingssystemen.....	18
7.4	Bespreking verschillen bestaande software met dit project.....	19
8	Studie gebruikers.....	21
8.1	Huidige situatie.....	21
8.2	Beoogde nieuwe situatie.....	23
9	Analyse:	25
9.1	Archivering vooraf of achteraf?.....	25
9.2	Moet gedefinieerd worden wat het originele document is?.....	26
9.3	Bij welke typen organisaties is dit soort software nodig?.....	26
9.4	Waarom zou het te ontwikkelen pakket gebruikt kunnen worden in zoveel verschillende typen organisaties?.....	27
9.5	Discussie: Moet het archief bepalen welke bestandsformaten er gehanteerd worden?.....	28
9.6	Discussie: Moet het archiefsysteem in XML opgeslagen worden.....	29
10	Nieuwe ideeën.....	31
11	Toevoegbare functionaliteit.....	33
11.1	Toegang vreemde organisaties (rollen).....	33
11.2	Samenwerken aan één document.....	34
11.3	Taakgenerator.....	35
11.4	Document genereren.....	36
11.5	Toegang tot documenten.....	37
12	Formulieren.....	39

13	Te verwerken gegevens.....	41
14	Ontwikkelkeuzes.....	45
14.1	Standaarden.....	45
14.2	Softwarekeuzes.....	45
14.3	Opslag van documenten.....	46
15	Het documentensysteem.....	49
15.1	Ontwerp.....	49
15.2	Gebruik van het archief.....	52
16	Adresboek.....	57
16.1	Ontwerp.....	57
16.2	Gebruik van het adresboek.....	61
17	Veiligheid.....	65
18	Zoekmachine.....	67
18.1	Fases in de zoekmachine.....	67
18.2	Werking.....	69
18.3	Zwaktes van de zoekmachine.....	72
19	Interface Design.....	73
19.1	De acht gouden regels voor interface design.....	73
19.2	Het organiseren van de weergave.....	74
19.3	De aandacht van de gebruiker dirigeren.....	74
19.4	Faciliteren van data-invoer.....	74
19.5	Identificatie van gebruikers, taken en interactiestijl.....	75
20	Performance.....	77
20.1	Aantal documenten.....	77
20.2	Hoeveelheid data.....	78
20.3	Aantal gebruikers.....	78
20.4	Conclusie.....	79
21	Evaluatie door gebruikers.....	81
22	Conclusie.....	83
23	Further development.....	85
23.1	Conversie.....	85
23.2	Backup.....	85
23.3	Sjablonen.....	85
23.4	Conversie van documenten.....	86
23.5	Postcodes zoeken.....	86
24	Literatuurlijst.....	87

1 Introductie

1.1 Algemeen

Archivering binnen huidige organisaties gebeurt volgens methodes die eigenlijk niet meer thuis horen in de 21ste eeuw.

Veel gebeurt met de hand, alles is nog op papier. Opvragingen zijn zelden geautomatiseerd. Hierdoor is het terugzoeken van gearchiveerde documenten allemaal handwerk. Veelal uitgevoerd door de archivaris.

De documenten zijn niet rechtstreeks beschikbaar voor de gehele organisatie.

Een goed archiefsysteem moet voor een kostendaling kunnen zorgen, zonder kwalitatief minder te zijn dan huidige systemen en procedures.

Teveel mensen binnen de organisatie hebben werk aan het archief. Daarnaast verliezen ze tijd door onduidelijke zoekopdrachten (of het niet weten van bepaalde gegevens).

Verder is het beheren van papier (dat nog buiten het archief staat) omslachtig en kost handenvol tijd en ruimte. Het beheer van documenten en bedrijfsinformatie is vaak ook decentraal georganiseerd[4].

De organisatie Van Lee & Van Everdingen levert archiveringsoplossingen voor binnen gemeentes. In het verleden hebben ze hiervoor een registratiesysteem laten maken (je kan hierin de gegevens van documenten opslaan), maar verder is het eigenlijk allemaal handwerk.

De gehele procedure van archivering zou herbekeken moeten worden en hiervoor software schrijven om niet alleen het archiveren, maar met name ook het terugzoeken van documenten te automatiseren.

1.2 Probleemstelling

De huidige procedures en software moeten in een geheel nieuw project en door het invoeren van nieuwe technieken en werkmethoden verbeterd worden. Hierbij moeten de documenten zowel voor de ambtenaren als later voor de burgers gecontroleerd beschikbaar komen en niet in een ontoegankelijk archief blijven.

Dit systeem moet dusdanig ontwikkeld worden dat het bruikbaar is voor alle gemeenten.

1.3 Outline

Deze thesis is als volgt georganiseerd:

In de hoofdstukken 2 tot en met 5 worden de opdracht en doelstellingen uitgediept.

Hoofdstuk 6 behandelt de huidige theoretische inzichten in het vakgebied archivering.

De hoofdstukken 7, 8 en 9 verkennen de marktsituatie, en analyseren de wensen en mogelijkheden voor archiefsoftware.

Hoofdstuk 10 en 11 behandelen nieuwe ideeën en mogelijke uitbreidingen die de software functioneel verbeteren binnen een organisatie.

Hoofdstuk 12 tot en met 14 bespreken de keuzes en randvoorwaarden die aan de toepassing worden gesteld.

De hoofdstukken 15 tot en met 18 bespreken het resulterende product met betrekking tot ontwerp, gebruik en beveiliging.

Hoofdstuk 19 bespreekt het gemaakte product met de betrekking tot de gebruikersinterface.

Vervolgens behandelt hoofdstuk 20 de performance van de software.

In hoofdstuk 21 wordt een evaluatie gegeven op basis van ervaringen van gebruikers van het systeem.

Tenslotte worden in hoofdstuk 22 de conclusies getrokken en in hoofdstuk 23 aangevuld met onderwerpen voor toekomstige verbeteringen en uitbreidingen.

2 Beschrijving van de opdrachtgever

Van Lee & Van Everdingen is een praktijkgericht en probleemoplossend bureau voor documentaire informatievoorziening en archiefbeheer.

De organisatie omschrijft zichzelf op haar website op treffende wijze als volgt:

We staan graag dichtbij u als klant. We leggen de nadruk op directe efficiency en kwaliteitsverbetering en dragen deze visie uit temidden van de snelle ontwikkeling die ons vakgebied de laatste jaren doormaakt.

Peter van Lee en Theo van Everdingen startten het bureau in 1988. Beiden waren toen al jaren in het vakgebied werkzaam. Peter van Lee overleed in oktober 1994. Zijn naam zal aan ons bureau verbonden blijven als inspirerend ondernemer op ons vakgebied.

Ons bureau is actief in alle aandachtsgebieden van documentaire informatievoorziening en archiefbeheer. De medewerkers hebben een gedegen opleiding, zijn enthousiast over het vak en beschikken over jarenlange ervaring, opgedaan bij tal van overheidsinstellingen en particuliere organisaties.

Voor deze opdracht is ervoor gekozen om een praktijksituatie te hanteren, om zo de huidige methoden en hun beperkingen zichtbaar te maken.

Er is door Van Lee & Van Everdingen gekozen om met het archief van de gemeente Den Briel aan de slag te gaan. De gemeente Den Briel was op dat moment een klant waar Van Lee & Van Everdingen werkzaam was. Binnen die gemeente is aan mij toegang gegeven om uit te zoeken hoe archivering in de praktijk werkt, en waar die eventueel te verbeteren valt.

3 Onderzoek opdracht via interviews/vergaderingen

3.1 Huidige software van Van Lee & Van Everdingen

In het verleden heeft “Van Lee & Van Everdingen” al een softwarepakket laten ontwikkelen. Dit is een postregistratietool.

In dit systeem wordt bij binnenkomst een brief geregistreerd, voor het z'n traject doorheen de gemeente start. Hiermee kunnen de archivarissen controleren of een document terug gebracht is naar het archief.

Voornaamste velden:

Soort (een type document)

Naam (een naam voor de brief)

Voorletters (van de verzender)

Tussenvoegsels (van de verzender)

Achternaam (van de verzender)

Postcode (van de verzender)

Plaats (van de verzender)

Huisnummer (van de verzender)

Adres (van de verzender)

Onderwerp (van de brief)

Registratie (datum dat de brief binnengekomen is op de archiefafdeling)

Poststuk (datum dat de brief binnengekomen is op de postafdeling)

Signalering (berekende datum = registratie + 42 dagen)

Afhandeling (berekende datum = registratie + 56 dagen)

Afgedaan (datum brief behandeld)

Rappelerend (=boolean; of een reactie verplicht is)

De datums zijn veelal datumvelden, de andere velden veelal tekst-velden (geen foreign key)

In het pakket, met name onder andere tabbladen zijn ook veel velden die eigenlijk niet gebruikt worden (zoals routing, agenda, logboek).

Daarnaast waren er nog vragen naar uitbreidingen. Met name rond vernietiging van documenten (na 20 jaar kunnen de meeste documenten vernietigd worden, anderen pas later). Hiervoor was geen registratie opgenomen.

Ook werd het uitlenen uit het archief niet geregistreerd.

3.2 Onderzoek werking archiefafdeling/postafdeling Den Briel

De postkamer en de archiefkamer zijn naast elkaar gesitueerd.

De postkamer ontvangt de post. Daarna wordt deze opengemaakt en gesorteerd.

Daarna door de archivaris geregistreerd (met de software hierboven beschreven).

En uiteindelijk met een bode het document naar de betreffende afdeling brengen.

De archiefafdeling ontvangt van de verschillende afdelingen dossiers terug. Deze worden tijdelijk in de lokale kast opgeslagen. Na verloop van tijd (over het algemeen > 1 jaar) worden deze verplaatst naar het archief in de kelder.

Daarnaast doet de archiefafdeling opvolging van de post (ontvangstbevestigingen, afhandelingstijd van het dossier/document) alsook het opzoeken van oude documenten, mocht daar vraag naar zijn.

3.3 Document Structuur Plan

In een DSP (Document Structuur Plan)[9] wordt er een relatie gelegd tussen organisatiestructuur, werkprocessen en de uitwerking hiervan. De organisatie stelt een structuur op waarbinnen ze werken, en kopieert deze structuur in het DSP.

Dit wordt vervolgens gebruikt bij het gebruik en archivering van de documenten. Hierdoor is de structuur herkenbaar door de medewerkers.

3.4 Preservatie en XML

Bij het initiële onderzoek is er gesproken over preservatie[3] en XML[12] als standaard opslagformaat.

Aan XML kleven nogal wat praktische bezwaren om hiervan een eigen formaat van te maken om zowel documenten als hun metadata (de gegevens over het document) in hetzelfde bestand op te slaan. Wat wel kan is om de documenten direct in passende XML standaarden op te slaan zoals OASIS (Organization for the Advancement of Structured Information Standards)[33]. Dit zou inhouden dat de facto standaard tools in gemeenten zoals Microsoft Office verlaten zouden moeten worden ten voordele van met name gratis alternatieven.

Dit zag men bij Van Lee & Van Everdingen als niet realistisch, en werd dus verworpen.

Uiteraard vraagt een bedrijf dat archiveringsoplossingen levert naar preservatie. Nu is preservatie veeleer procedureel dan werkelijk iets als software.

Het kopen van de juiste hardware, en het opstellen van de juiste procedures is buiten dit project gehouden, omdat het de scope van het project gewoon te groot zou maken.

Toch is erover gesproken, en de conclusie getrokken dat als deze software de mogelijkheid zou hebben om de documenten om te zetten in XML-documenten, met de gegevens uit de database in de XML-metadata, dit een enorm voordeel zou zijn. Deze “backup” van de data en de documenten, kan dan gebruikt worden in de preservatieprocedures van de data van de organisatie.

4 Beschrijving van de opdracht

4.1 Algemeen

De gemeente Den Briel heeft een soepel lopend archiefsysteem. Er is echter een probleem om binnengekomen documenten snel beschikbaar te maken voor de organisatie.

Situatie nu:

In de postkamer komen documenten binnen. Deze worden opengemaakt en doorgegeven aan de archivaris.

Deze “registreert” de documenten. Dat wil zeggen dat ze in een softwarepakket ingevoerd worden. Daarin wordt vastgesteld hoe ze in de toekomst gearchiveerd moeten worden, en voor wie ze bedoeld zijn.

Dan worden de documenten door de bode bij de betreffende personen bezorgd. Deze kunnen ze dan verwerken.

Als het dossier is afgewerkt, komen de documenten in dossiervorm terug bij de archivaris. Deze keer om te archiveren.

Na archivering moeten documenten vaak met de hand opgezocht worden. De documenten die geregistreerd staan, kunnen sneller opgezocht worden, maar het blijft veel werk.

De vraag is om de hele procedure te herbekijken met de volgende doelstellingen:

- a) Minder handwerk
- b) Minder risico op verlies of schade aan documenten

4.2 Programma van Eisen

1) Vanuit de opdrachtgever: Van Lee & Van Everdingen (commercieel belang) en semi-opdrachtgever Gemeente Den Briel

- Zo min mogelijk veranderingen aan de werkwijze
- Integratie gemeentelijke basisadministratie
- Software uitsluitend voor gebruik in archiefkamer
- Goed zoekstelsel dat gebruikt kan worden door de archivaris

2) Vanuit de wetgever[29]

- Opslag in een XML-formaat
- Document Structuur Plan (DSP)
- Open Standaarden hanteren [31]

3) Vanuit de eindgebruiker

- Gebruik van Windows
- Het hanteren van Word
- Geen banenverlies bij het personeel
- Bode blijft bestaan
- Simpele bediening van de software

Alhoewel deze eisen begrijpelijk zijn vanuit het standpunt van de verschillende personen, zijn deze niet erg behulpzaam bij het ontwikkelen van nieuwe archiveringstechnieken en software.

Zowel Van Lee & Everdingen, de gemeente Den Briel, als de gebruikers zijn voorstanders van het handhaven van huidige methoden en technieken.

De theorie voor het ontwikkelen van nieuwe archiefsoftware (zoals OAIS: The Open Archival Information System)[13,18], was dusdanig complex dat niemand zich een voorstelling kon maken op wat voor manier dit hun werken zou gaan beïnvloeden.

5 Extra doelstellingen

In het verleden heb ik al projecten gedaan in documentenbeheer. Hierdoor zijn mij een aantal “tekortkomingen” aan archiveringsprocedures opgevallen. Het zijn veelal dingen die op de standaardmanier worden uitgevoerd en waarvan vermoed kan worden dat het beter kan.

- Een archief maken dat het hart is van de organisatie. Van oudsher ziet men een archief als iets dat in de kelder staat. Tegenwoordig zijn er een hoop projecten om het archief in de etalage te zetten (daarmee bedoelt men ter beschikking stellen aan klanten). Met beide ideeën ben ik het oneens. Een archief moet geen eindpunt zijn van documenten (uitsluitend nog te bezichtigen), maar het hart van de verwerking van documenten. Documenten worden eerst gearchiveerd, daarna pas gebruikt.
- Het archief moet de basis zijn van een kantoorautomatisering. Het uiteindelijke doel van een digitaal archief moet zijn dat mensen efficiënter werken, niet meer taken hebben.
- Veiligheid moet generiek zijn. Het archief zal een vorm van veiligheid nodig hebben. Bij besturingssystemen weten we dat we per document kunnen aangeven hoe veilig het moet zijn (wie mag lezen, danwel schrijven). Dit is voor een archief omslachtig en gaat veel werk kosten. Veiligheid moet derhalve generiek doorfilteren.
- Drie-lagenarchief. Ieder archief binnen een organisatie heeft een vorm van dossiers. Veel software is wel geschikt om met dossiers als gegeven te werken, maar vergeten de gebruikers de mogelijkheid te geven verschillende soorten dossiers aan te maken.
- Organisaties vertonen nogal wat overeenkomsten en verschillen bij de verwerking van hun documenten. Het archief moet flexibel zijn om die verschillen mogelijk te maken, maar rigide bij de overeenkomsten om het gebruik simpel te houden.
- Koppeling tussen archief en een adresboek. Dit omdat verzender, ontvanger, beheerder, etc. werkelijke personen zijn, en niet een tekstveld waarop gezocht kan worden. De documenten moeten per persoon gezocht kunnen worden.
- Nieuwe structuur adresboek om centraal mensen “functies” te geven in plaats van verschillende tabellen te hebben voor bedrijven, afdelingen, groepen, filialen, en allerlei personen.
- Koppeling archief en todo's: Het binnenkomen of soms vertrekken van een document zet aan tot benodigde acties van bepaalde personen/computers.
- Een cleane, simpele interface. Veel interfaces zijn onnodig complex en traag.

6 Theorie

6.1 Records management (ISO 15489): Procedureanalyse voor een document in een organisatie

Een organisatie die wil voldoen aan ISO 15489[11,16,37] moet een beleid en procedures opstellen, documenteren, onderhouden en in gebruik nemen om te zorgen dat aan de behoefte van bewijsvoering, verantwoording en informatie over z'n activiteiten is voldaan.

Organisaties moeten beleid opstellen voor records management. Het doel hiervan is de creatie van beheersbaarheid van authenticiteit, betrouwbare en bruikbare records. Dit beleid moet ingevoerd en gebruikt worden doorheen de gehele organisatie.

En opgesteld worden door een analyse van de activiteiten van de organisatie.

Discussie ISO 15489

1. **Beleid:** Beleid wordt bij de leiding opgesteld. In dit project zou het dan de archivaris moeten zijn die dit implementeert. Dit zou op een manier in de structuur moeten komen. Misschien is als basis hiervoor iets als het DSP bruikbaar.
2. **Doorfilteren:** Dit beleid moet doorfilteren doorheen de gehele organisatie. De archivaris stelt het op, en bij de werknemers moet dit gevolgen hebben. Zij moeten deze structuur volgen. Ergens hierin zal nog een vorm van vrijheid moeten zijn, om niet helemaal vast te komen in de structuur.
3. **Bewijsvoering:** Er moet een vorm van bewijsvoering zijn van het bestaan, en van de datum van invoering van een document.
4. **Verantwoording:** Er moet een vorm van controle zijn door de archivaris. Hiervoor lijkt me het moment van afsluiting van een dossier een geschikt moment aangezien dit moment nu al gebruikt wordt.
5. **Informatie over de activiteiten:** Zowel in de documenten als in de metadata van deze documenten zal een hoeveelheid informatie over de activiteiten van het betreffende bedrijf (moeten) staan. De indeling zal dus ook de bedrijfsactiviteiten aan moeten kunnen.

6.2 MoReq Classification Scheme

MoReq (Model Requirements for the management of electronic records)[10,21] is op vraag van de DLM (Document Lifecycle Management)-Forum[36] bestaande uit het Europese Unie onderdeel dat archivering doet en partijen geïnteresseerd in archivering en record management) ontwikkeld door Cornwell Management Systems en het stelt de eisen op voor een ERMS (electronic records management system).

MoReq is in gebruik doorheen Europa en ver daarbuiten, en kan als dusdanig gezien worden als een succes. Hierdoor, en omdat er geen onderhoud, is er besloten MoReq2 te ontwikkelen. Dit model is gepubliceerd in 2008.

MoReq is opgesteld als een set van eisen waaraan een elektronisch archief zou moeten voldoen.

Basis-schema:

Het schema bestaat uit twee delen:

a) Hiërarchie

De structuur waarin documenten ingedeeld worden heeft veel weg van de directory-structuur van een computer.

De indeling begint bij een "classification scheme". Een soort "root" voor een directory.

Hieronder worden "levels" ingedeeld in een boomstructuur. Net zoals directory's kunnen onder die levels weer nieuwe levels zitten.

In deze levels kunnen zich "files" bevinden, net zoals documenten in een directory. En deze documenten bevatten weer "Records" oftewel gegevens.

b) Digitale opslag van het fysieke document en de bijbehorende gegevens

Hier worden digitale tegenhangers gecreëerd van fysieke zaken. De bedoeling hierbij is dat dit, omdat niets anders toegepast wordt dan wat er in een fysiek archief gebeurt, ook simpel uitgevoerd kan worden in een digitale vorm.

Een document krijgt een digitale tegenhanger, net zoals z'n gegevens, kopieën en versies van dit document alsook het dossier (volume) waartoe het document behoort.

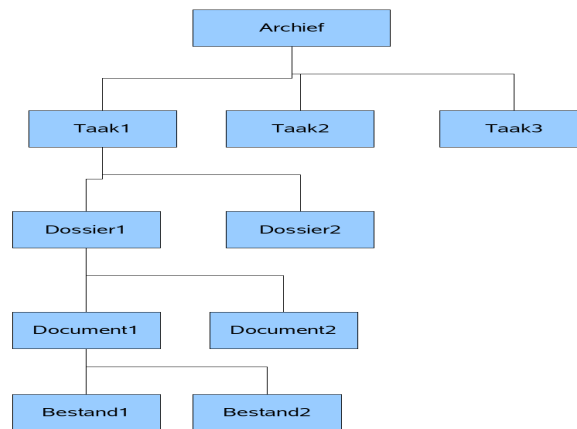
Eisen MoReq

Deze kunnen in een aantal onderdelen ingedeeld worden:

1. Classificatieschema: Structuur van het geheel.
2. Veiligheid: Toegangscontrole,
3. Verwijdering van documenten: Periodes, controles voor verwijdering en hoe verwijderen
4. Invoering van documenten: Document soorten, bulk import, e-mail, etc.
5. Verwijzingen
6. Zoeken en weergeven: Opzoeken, bekijken en printen van documenten en gegevens
7. Administratieve functies: Rapportages, veranderen en verwijderen van gegevens
8. Systeemeisen: Gebruiksgemak, performance, betrouwbaarheid

Discussie MoReq

Het archief dat, volgens mijn interpretatie van het DSP, het best gebouwd wordt, heeft een structuur van drie lagen structuur (of vier als de bestanden meegeteld worden) zoals geïllustreerd in Figuur 6.1.



Figuur 6.1: Structuur

De werking van de organisatie wordt verdeeld in taken. Een taak kan zijn de boekhouding, maar ook verkoop, of archivering. Er hoeft niet persé een afdeling gebruikt te worden, alhoewel die indeling van organisatie ook in de software gebruikt kan worden. De archivaris kan in deze laag de structuur aanleggen die de organisatie als geheel gebruikt.

Daarbinnen worden dossiers aangelegd, zoals bekend is bij de meeste organisaties. Binnen dossiers geeft een werknemer aan welke documenten gezamenlijk een taak uitvoeren, en als zodanig de gezamenlijke context van deze documenten.

Binnen een dossier zijn er documenten. Een document kan bestaan uit meerdere bestanden (zoals bladzijden in een boek).

Gebruikers:

MoReq definieert twee groepen personen:

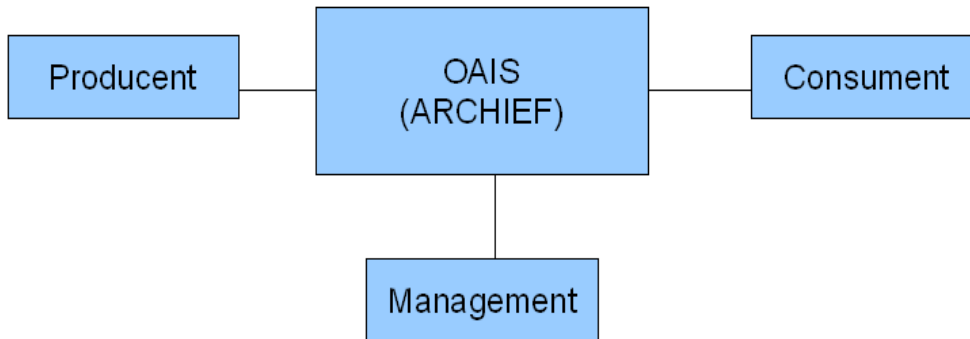
1. "User". Dat is iemand met "access" of toegang tot het systeem
2. "Administrator". Dat is iemand met beheerstoegang tot het systeem.

Voor dit systeem worden beiden in twee groepen gesplitst.

- 1.1 Gebruiker uit het bedrijf. Deze kan schrijfrechten hebben binnen de dossiers in zijn beheer.
- 1.2 Gebruikers van buiten het bedrijf. Deze hebben voor leestoegang tot dossiers en/of documenten binnen dat dossier, waarbinnen zij een bepaalde functie uitvoeren.
- 2.1 Postverwerkers. Deze voeren de initiële post in.
- 2.2 Beheerders. Deze doen de indeling van de structuur zoals nodig in bepaalde delen van het systeem, zoals veiligheid, maar ook de indeling van de taken.

6.3 Theoretisch Model OAIS (ISO 14721)

Omgevingsmodel



Figuur 6.2: Omgevingsmodel

Onderdelen van Figuur 6.2:

Producent: Diegene die de documenten oplevert.

Consument: Diegene die de documenten uit het archief haalt

Management: Beheerder van de documenten in het Archief

OAIS Archief: Het feitelijk archief

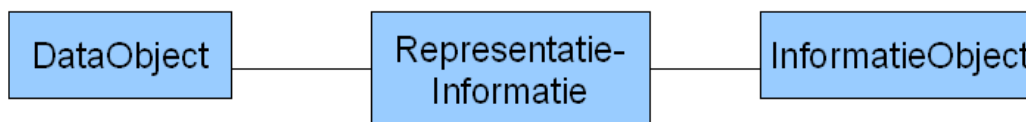
Werking: De producent levert een document aan in het OAIS-archief[13,18].

Dit bewaart dit document. Wanneer beheer aan dit document nodig is, moet dit gedaan worden door Management. Deze doet de aanpassingen die nodig zijn.

De consument kan, wanneer deze dat wil, een document opzoeken en openen uit het archief.

Informatiemodel

We moeten van een basis-informatiebestand naar een representatie gaan waar de gebruiker wat aan heeft.



Figuur 6.3: Informatie Object

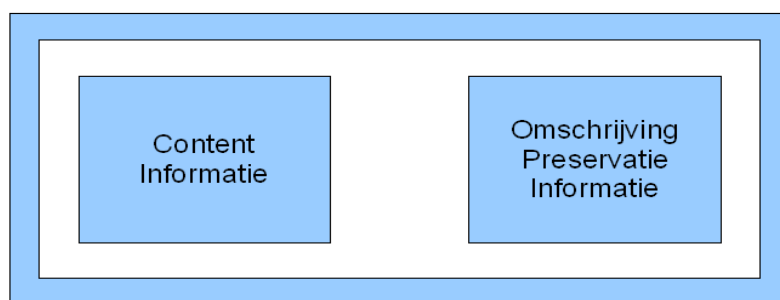
Stel we hebben een pdf-document. Dit is dus het "DataObject". De consument leest geen bits, dus moeten we deze omzetten in iets grafisch (representatie van het object, is dus het InformatieObject). Hiervoor gebruiken we dus de "Representatie Informatie". Hierin staat hoe een pdf-file gelezen moet worden, en geven we de resulterende "InformatieObject" in leesbare vorm terug aan de gebruiker.

Nu heeft een DataObject twee soorten data. De preservatie-informatie, en de gegevens die meer vertellen over de inhoud van de DataObject (beter bekend als metadata).

Deze moeten beiden opgeslagen worden in een OAIS informatie-pakket.

We hebben hier dus "Content Informatie" oftewel de data die eigenlijk moet worden bewaard en "Omschrijving Preservatie Informatie" oftewel de benodigde gegevens om de gegevens weer zichtbaar te maken.

Elk van deze twee delen hebben een eigen data_object, om de gegevens die erin staan weer zichtbaar te maken.



Figuur 6.4: Informatie Pakket Omschrijving

In de meest algemene vorm hoeft content informatie niet perse digitale informatie te zijn. Het kan ook fysieke informatie zijn, waarover informatie op de computer dient bijgehouden te worden.

De "Omschrijving Preservatie Informatie" heeft als doel de informatie te bevatten waarmee men de "content informatie" weer zichtbaar kan maken of kan begrijpen.

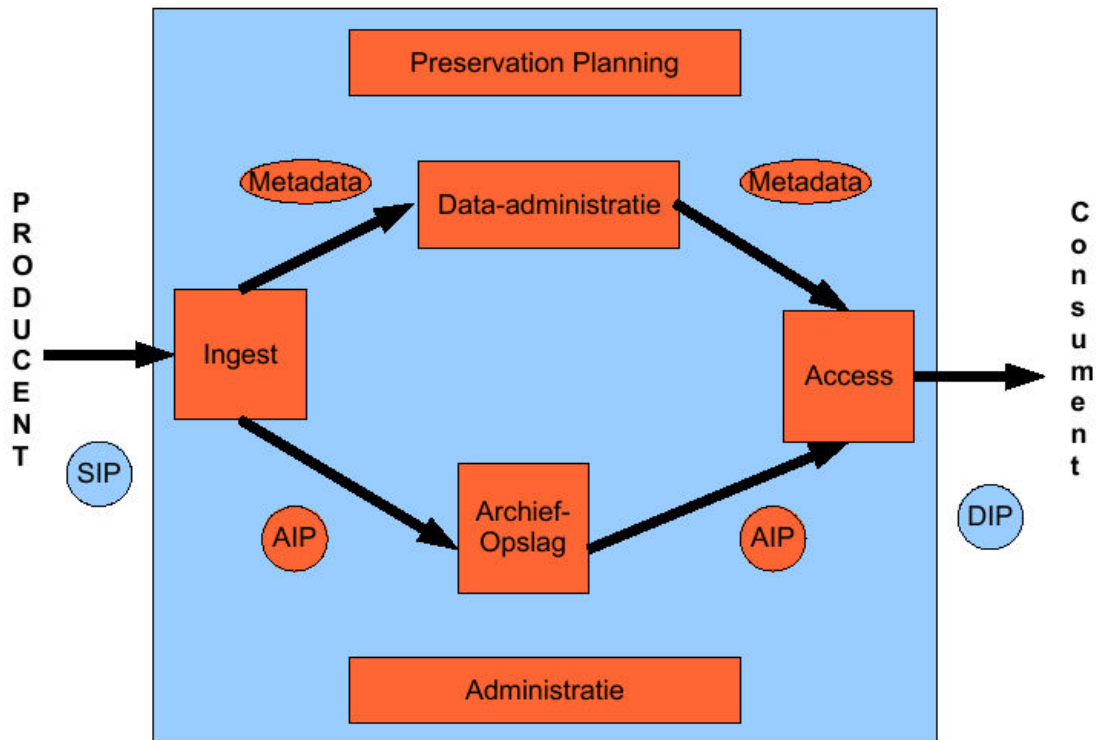
Deze informatie bevat vier sub-categorieën:

1. De "Reference Information" wordt gebruikt om "identifiers" te maken waarmee het systeem de "content Informatie" uniek kan herkennen.
2. De "Provenance Information" beschrijft de geschiedenis van de Content Informatie inclusief de bewakingsketen (welke personen verantwoordelijk zijn voor het document zodat de gebruiker zelf kan vaststellen of deze de inhoud vertrouwt).
3. De "context Informatie" maakt de associatie tussen de "Content Informatie" en andere informatie dat niet in de "Information Package" zit.
4. De "Fixity Information" draagt er zorg voor dat de "Content Informatie" niet veranderd wordt op een ongedocumenteerde methode. Denk hierbij aan checksums en digitale handtekeningen.

Het informatie-pakket bestaat uit drie varianten, afhankelijk van waar het zich bevindt in het OAIS-archief.

1. SIP (Submission Information Pakket). Deze wordt gebruikt zodat de producent van het document de benodigde informatie mee kan geven aan OAIS archief.
2. AIP (Archival Information Pakket). Deze wordt gebruikt door het OAIS archief zelf om de Content Information en de Preservation Description Information vast te houden zodat het document kan bewaren.
3. DIP (Dissemination Information Pakket). Deze is ervoor om te zorgen dat de consument de benodigde informatie ontvangt.

Functionele model



Figuur 6.5: OAIS

Er zijn zes basisfuncties te definiëren in een archief:

1. Ingest: Deze is de interface tussen de producent en het OAIS-archief. In deze fase levert de producent "Submission Information Packets" tijdens een "Data Submission Session".
2. Documenten Opslag: Deze accepteert AIP, en bewaart deze, en op aanvraag worden deze weer aangeboden.
3. Data beheer: Deze accepteert AIP van de Ingest vandaan en andere soorten meta data die nodig zijn om algemene OAIS operaties mogelijk te maken.
4. Administratie: Deze is verantwoordelijk voor de algemene gang van zaken binnen het OAIS archief.
5. Access: Deze entiteit maakt het voor consumenten mogelijk maken dingen te vinden, en hiertoe dan toegang te krijgen.
6. Preservatie planning: Voor ieder onderdeel in het archief moet continu gecontroleerd worden of het nog gebruikt kan worden in de toekomst.

Samenvatting: De producent levert een document aan in het OAIS-archief. Dit bewaart het document. Wanneer beheer aan dit document nodig is, moet dit gedaan worden door Management. Management doet de aanpassingen die nodig zijn. De consument kan indien gewenst een document opzoeken en opvragen uit het archief.

7 Bestaande Software

7.1 Document management systemen

In het kader van dit project is er gekeken naar een aantal bestaande pakketten zoals Xerox DocuShare[22,39], Alfresco[8,26], OpenDocMan[34] en Nuxeo[19,32].

Deze zouden stuk voor stuk besproken kunnen worden, maar dat zou veel tijd kosten en onnodig zijn. In de basis werken al deze systemen hetzelfde, dus worden ze gezamenlijk behandeld.

Allemaal beginnen ze bij een filesystem. Bestanden en folders.

Een bestand kan “geüpload” worden in het systeem. En daarin in een folder geplaatst worden. Van het bestand wordt bijgehouden door wie het is geplaatst, datum, commentaar, bestandsnaam, etc. Niet veel meer dan de standaardgegevens die een filesystem bijhoudt van een bestand.

Sommige kunnen ongeveer dezelfde gegevens bijhouden van een folder, maar niet allemaal.

Allemaal hebben ze rechtenbeheer. Zo kunnen er op zowel bestandsniveau als folderniveau lees- danwel schrijfrechten verleend worden (schrijfrechten geven toestemming om bestanden in te voeren, alsook de gegevens te veranderen). Dit zowel algemeen, per groep als per gebruiker. Ook wordt er gebruik gemaakt van “overerving” van rechten vanuit folders naar lagere folders of bestanden. Hierin verschillen de verschillende systemen maar weinig met bekende filesystemen zoals gebruikt in Windows of Linux.

De meer geavanceerde systemen hebben een vorm van “versiebeheer”. Veelal met een “check-out” en “check-in” methode. Hiermee kan een bestand uitgeleend worden en later een nieuwere versie ingevoerd worden.

Het beschikbaar stellen van de bestanden kan via de eigen software (veelal via een web-frontend), de meeste kunnen de bestanden ook als een “harde schijf” beschikbaar maken. Sommige nog via nieuwere methoden zoals WebDAV.

Een paar hebben een workflowmodule. Daarmee is het in een programmeertaal mogelijk om vervolg te geven aan het binnenkomen van een document.

Tekortkomingen van verkrijgbare document management systemen

Al deze systemen zijn eigenlijk filesystemen. Allemaal met een web-frontend, maar met minimale verschillen en uitbreidingen.

Geen enkele van de geteste systemen maakt gebruik van de archiveringsmethodes die gebruikt worden in bedrijven of overheden.

Nagenoeg geen enkele van de bekeken systemen kent ook maar één van de volgende in de bedrijven elementaire zaken:

1. Post (niet alle documenten worden in Word getikt en door de persoon ingevoerd)
2. Postdienst (voor het digitaliseren van binnengekomen post)
3. Archivaris (controle op alle documenten en hun gebruik)
4. Postopvolging (wordt er op tijd terug gereageerd)
5. Verzender/Ontvanger (wie heeft het document verzonden, of aan wie was het gericht?)

6. Methode verzending (brief, e-mail, fax, etc.)
7. Type document (gaat het om een aanvraag bouwvergunning?)
8. Dossier (meer dan een folder, dossiers worden geopend, afgesloten, alsook verwerkt)
9. Taken (een vorm van bundeling van dossiers, meestal verdeeld in afdelingen)
10. Rechten op basis van een functie in een dossier en type bestand
11. Een adresboek van alle verzenders/ontvangers
12. Door postopvolging, volgen er todo's voor in de agenda

In deze softwarepakketten merk je heel sterk dat men gestart is met de tools die een computer hen bood, en deze lichtjes hebben aangepast op vragen uit het bedrijfsleven.

In geen van de bekeken pakketten is er een bedrijfsanalyse gemaakt en is die analyse als basis gebruikt voor de softwareontwikkeling. Er is al helemaal niet gekeken hoe de stroming van documenten in een organisatie verbeterd kon worden.

Een aantal pakken groots uit met “collaboratie”-tools, maar in een bedrijf is dit minder nuttig dan men zou denken.

- a) Zo vaak werkt men niet met meerdere personen (die niet op dezelfde locatie zitten) samen aan één document.
- b) Uiteindelijk wil men in het archief niet alle versies van het document zien, maar juist die, die is verzonden/gebruikt.

7.2 Archiefsystemen

Een echt archiefsysteem is niet echt de bedoeling in dit project. Deze zijn voor hun doelstelling veel beter dan de verschillende document management systemen in hun doelstelling (in ieder geval veel betere theoretisch basis), maar niet bedoeld voor gebruik op de werkvloer.

Het is een “End Of Life”-behandeling van de documenten. Dit maakt dat ze zeer sterk zijn in zaken als preservatie, maar zwak in “hoe wordt een persoon op de hoogte gesteld dat er voor hem een nieuw document is”.

7.3 Postkamerverwerkingssystemen

Deze systemen zijn bedoeld om documenten te digitaliseren en metadata toe te voegen.

Typisch bevatten deze een methode om post te scannen, en dan de archivaris alle metadata toe te laten voegen (voornamelijk voor het later terugzoeken van het gescande).

Als laatste wordt het gescande “doorgestuurd” naar de geadresseerde en daarnaast meestal gearchiveerd.

Een archiefsysteem wordt hier dus niet als onderdeel van de bedrijfsvoering gebruikt, maar juist als archief zoals we dat nu kennen. Hiermee moeten de medewerkers nog steeds eigen dossiers aanmaken, die dan decentraal beheerd worden.

7.4 Bespreking verschillen bestaande software met dit project

Wat is een dossier? Voor dit project is het volgende antwoord geformuleerd: Een combinatie van documenten en gegevens waarbinnen één project uitgevoerd wordt binnen een opdracht die een reeks van gelijkaardige projecten omvat.

De doelstelling van dit project is dat niet alleen zoveel mogelijk dossiers van een bedrijf hierin gearchiveerd moeten worden, maar dat deze software gebruikt wordt als basis in de verwerking van diezelfde dossiers. Dit systeem kan dan melding maken van ontbrekende documenten, planningen en vertragingen op die planningen registeren, gegevens van verzender/ontvangers van documenten, alsook iedereen betrokken bij een dossier en deze direct gebruiken bij het werken.

Bij nagenoeg elk bedrijf dat ik geobserveerd heb, zie je die leuk gekleurde mapjes, die elk een dossier bijeenhouden. Deze methode van werken omgooien is de doelstelling en dit door een combinatie van het digitaliseren van die methode van werken als door het toevoegen van voordelen die een computer biedt. Het werken in teamverband aan dossiers zal hierdoor versterkt worden omdat alle teamleden over dezelfde documenten en gegevens beschikken. Hiernaast kan de agenda een belangrijk hulpmiddel zijn bij het tijdige verwerken van dossiers.

Dat personen/bedrijven buiten de eigen organisatie toegang kunnen krijgen tot beperkte set gegevens en documenten van hun dossier, alsook het later terug kunnen zoeken van documenten, zijn een bruikbare bijkomstigheid van dit project.

8 Studie gebruikers

8.1 Huidige situatie

Om uit te zoeken hoe het proces van postverwerking in een organisatie verbeterd kan worden, moet er gekeken worden hoe postverwerking nu gebeurt.

Daarvoor is er bij de gemeente Den Briel onderzocht hoe postverwerking gebeurt, en dit vergeleken met andere organisaties, om een uniform beeld te krijgen hoe documenten/post zich binnen organisaties beweegt.

Inkomende post

- Postbode levert een postzak aan de postkamer.
- Iemand maakt alle brieven open.
- De post wordt geregistreerd. Hierbij wordt aangegeven het type post. Of er een ontvangstbevestiging moet komen, en binnen welke tijd.
- Bepaling naar welke afdeling de brief moet voor verdere verwerking. Daarnaast wordt deze klaargelegd voor de bestemming of beheerder van het dossier.
- Ontvangstbevestiging opsturen (indien nodig) door de archivaris.
- De bestemming wordt op de hoogte gesteld van de ontvangen post.
- Ofwel de persoon zelf, of een vast iemand brengt de post naar de betreffende persoon of zijn postvakje.
- De post wordt brief voor brief doorgenomen door de bestemming, en ingedeeld in dossiers.
- Dossier wordt bekeken als geheel, documenten die ontbreken worden aangevraagd.
- Als het dossier volledig is, wordt dit afgewerkt door een antwoord op te stellen op de oorspronkelijke vraag.

Uitgaande post

- Printen uitgaande post.
- Brief brengen naar de postkamer, inclusief commentaren of deze langs B&W moet.
- Registratie uitgaande post.
- Goedkeuring B&W of gemeenteraad (in tweevoud indien nodig).
- Één kopie moet naar postvak verzender.
- Publicatie (op website of gemeentekrant, indien nodig).
- Verzending brief naar bestemming.

Archivering dossier

- Als het dossier afgehandeld is moet dit naar de archiefafdeling.
- Deze kennen een code aan het dossier toe.
- Registratie van het dossier, inclusief code, hoe dit opgeslagen wordt en voor welke tijd.
- Opslag van dossier in het archief.

Commentaar: Documenten blijven heel lang in de organisatie zweven. Archivering gebeurt pas als het nut van het document voorbij is. Dat kan na maanden of in sommige gevallen zelfs jaren zijn. Het risico is daarbij dat het origineel “verloren” of beschadigd geraakt.

Een ander probleem aan lang vasthouden van het origineel is dat dit nogal wat werk veroorzaakt. Iedere keer dat het in andere handen overgaat moet dit geregistreerd worden. Daarnaast moet het ook fysiek verplaatst worden binnen de organisatie.

Verder kan maar één persoon het tegelijk “behandelen”. Een leidinggevende kan dus geen overzicht

houden van wat de mensen onder hem doen en een collega kan ook niet inspringen. Bijvoorbeeld bij onverwachte telefoontjes als de persoon op vakantie is.
Als laatste nemen de dossiers zelf fysiek een hoeveelheid ruimte in. De hele organisatie staat dan vol met dossierkasten bij ieder bureau.

Oplossing?

Zet het archiveren van het document eerder in het proces. Als de post binnenkomt, kan deze gescand worden, en dan gearcheeerd.

Het origineel kan dan fysiek gearcheeerd worden, niet in een dossier, maar als los document (wel in het dossier plaatsen waar deze zich bevindt, bijvoorbeeld op scandatum).

Het gescande document (de bits en bytes dus) kan op een computer in een dossier geplaatst worden. De gebruiker hoeft dan alleen een e-mailtje te krijgen dat er een document is.

Doordat het document direct gearcheeerd wordt, kan het niet meer verloren of beschadigd worden.

Het document hoeft maar één keer geregistreerd te worden. De bewegingen van het document in de organisatie zijn weg omdat het document gelijk in het archief verdwijnt. Hierdoor kunnen ook meerdere personen het document bekijken en gebruiken bij de afwikkeling van dit dossier, of gebruiken in andere dossiers.

Op de werkvloer komt dan minder papier voor, waardoor er niet per werknemer een methode van bewaring ingevoerd hoeft te worden.

8.2 Beoogde nieuwe situatie

Voor de inkomende post is er een probleem. Het meest wenselijk is dat de archivaris alles invult. Dat bespaart bij de ambtenaren het meeste tijd.

Van de inhoud van de dossiers weet de archivaris echter niks af en het zou teveel tijd kosten om ze allemaal uit te pluizen; om te bepalen in welk dossier een document thuis hoort.

Voor de inkomende post is bijgevolg een oplossing in twee stappen nodig. De archivaris doet al het invulwerk van de post die binnenkomt (verzender, type document, etc.), en bepaalt voor welke “taak” dit document bedoeld is.

De ambtenaar controleert deze gegevens, en geeft alleen het dossier aan waartoe het document moet behoren.

De plaatsing van het document in het juiste dossier, alsook het beheer van de dossiers is voor de betreffende ambtenaar/afdeling, en niet voor de archivaris.

Inkomende post

- Postbode levert een postzak binnen.
- Iemand maakt alle brieven open.
- De post wordt gescand.
- Archivaris archiveert het origineel. Hiervoor moet hij deze een code geven (opeenvolgende code?).
- Archivaris geeft gegevens in op de computer. Hiermee wordt de post ingevoerd in het systeem, maar nog niet in een dossier geplaatst. Voornamelijk moet hier vastgesteld worden naar welke afdeling het document moet.
- Op de afdeling wordt het document aan een dossier toegevoegd (evt. een nieuw dossier aangemaakt).
- Dossier wordt bekeken als geheel, documenten die ontbreken worden aangevraagd.
- Als dossier volledig is, wordt dit afgewerkt door een antwoord op te stellen op de oorspronkelijke vraag.

Uitgaande post

- Printen uitgaande post bij de postafdeling over het netwerk.
- Goedkeuring B&W of gemeenteraad.
- Scannen van goedkeuring, toevoegen aan dossier.
- Publicatie.
- Verzending brief naar bestemming.

Archivering dossier

- Afsluiting dossier wanneer dit klaar is.

Commentaar: Duidelijk uit de oude en nieuwe situatie is dat er minder taken zijn, en deze veelal sneller uitgevoerd kunnen worden.

Als de archivaris de initiële gegevens heeft ingevoerd, zit dit document nog niet in een dossier. Bijgevolg zit het in een soort “limbo”.

We gaan er gemakshalve vanuit dat tijd van een archivaris goedkoper is dan tijd van iemand op een afdeling. Dit wil zeggen dat de archivaris een “service” verkoopt aan de andere afdelingen door zoveel mogelijk gegevens in te vullen, zodat deze geen tijd kwijt zijn aan gegevensverwerking, maar meer tijd hebben voor hun eigenlijke taak.

Om de tijd van de archivaris goedkoper te maken, moet er of zoveel mogelijk geautomatiseerd worden (zodat het werk minder tijd kost), of het werk makkelijker worden zodat een goedkopere archivaris gebruikt kan worden, of beide.

Automatisering is maar beperkt toepasbaar. Omdat veel werk het overzetten is van fysiek (een adres op een brief) naar digitaal is automatisering hiervan niet zo makkelijk. Het werk makkelijker maken gaat beter, omdat het nogal repetitief van aard is. Hierdoor kan veel werk door lager opgeleide mensen uitgevoerd worden, en dus goedkoper.

9 Analyse:

9.1 Archivering vooraf of achteraf?

Bij de analyse bij de gemeente Den Briel viel iets op.

De post komt binnen in de postkamer (vlak naast de kamer van de archivariissen).

Vervolgens gaat de post naar de archivaris om te registreren.

Daarna gaat het document het gebouw in, maar later moet datzelfde document weer bij de archivaris terecht komen. Is dit logisch?

Het hele doel van een digitaal archief is beschikbaarheid. Waarom hebben de mensen binnen de organisatie een document fysiek nodig?

Wat navraag leerde dat er bepaalde gebeurtenissen zijn waarbij de originele documenten gehanteerd moeten worden (zoals bij een rechtszaak). Deze redenen zijn veelal de basis waarom de documenten ook fysiek gearchiveerd moeten worden.

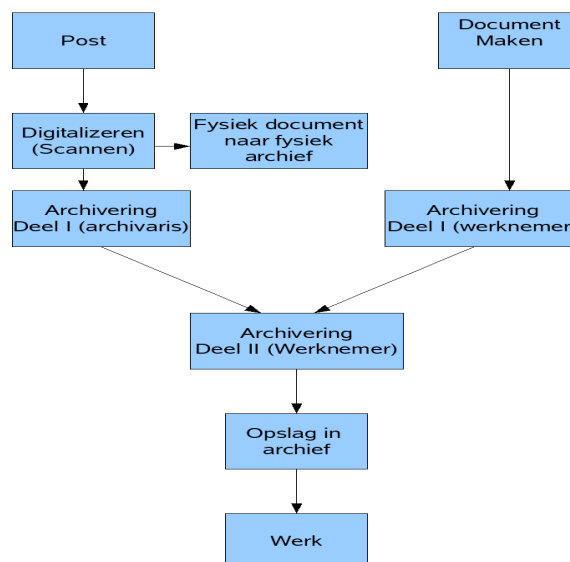
Echt nodig voor de verwerking van het dossier zijn ze veelal niet.

Archivering over het algemeen gebeurt na het actieve leven van een document. In veel projecten wordt dit als vanzelfsprekend beschouwd.

Een document invoeren na z'n actieve leven leidt tot het volgende proces:

- Registreren: Het document wordt geregistreerd bij binnenkomst
- Opvolging: Het document moet opgevolgd worden als het in de organisatie gebruikt wordt.
- Archivering: Uiteindelijke opslag van het document.

Als het document eerst gearchiveerd wordt (in plaats van de registratie dus) krijg men ongeveer de structuur zoals afgebeeld in Figuur 9.1



Figuur 9.1 Volgorde archivering

Hierbij wordt het document eerst gearchiveerd, en dan pas gebruikt. Hierdoor is er geen dubbel werk nodig van de archivaris (registreren en archiveren), en krijgt de werknemer z'n document digitaal. Het origineel kan dan niet meer zoek raken of beschadigd worden.

9.2 Moet gedefinieerd worden wat het originele document is?

Allereerst weten niet veel organisaties te definiëren wat een “origineel” document is. Voor brieven die binnenkomen is dat uiteraard de betreffende brief.

Als we het hebben over documenten die digitaal aangeleverd worden (fax, email, zelfgemaakte documenten) is het verhaal anders.

Voor velen (gevoelsmatig overigens) is de eerste “print” van het document het origineel. Als een fax binnenkomt, is de eerste kopie ervan op de printer het zogenoemde origineel. De originele bits en bytes in de computer wordt gezien als de “kopie”, alhoewel het digitaal is binnengekomen.

Dit kan natuurlijk veranderd worden. Het gevoelsmatig origineel hoeft dan niet gearchiveerd te worden, maar de eerste print-out van het document dat gearchiveerd wordt, is gewoon het origineel.

9.3 Bij welke typen organisaties is dit soort software nodig?

Bij heel verschillende bedrijven is deze software een hulpmiddel bij het dagelijks werk. Dit omdat de werkwijze in de meeste bedrijven wel op elkaar lijkt.

De meeste mensen die met documenten werken, werken ongeveer op dezelfde manier. Alle acties starten op 1 van 2 manieren:

1. Bij een document binnenkomend in het systeem.
2. Een menselijke actie (een vraag van een collega, student die iets doet, colleges die gegeven moeten worden, etc.)

Bij 2 is er eigenlijk ook een document, alleen is dit onzichtbaar voor het systeem (Je zou kunnen eisen dat voor alles een opdrachtbon wordt opgesteld. Dat is echter onwenselijk).

Een document kan van alles zijn (een klacht van een klant, een factuur die betaald moet worden, een opdrachtbon, een offerteaanvraag, etc.)

Op basis van dat document moet er een “actie” starten.

Deze actie heeft weer zijn gevolgen. Allereerst kan deze actie leiden tot documenten of vragen aan andere personen, waardoor deze weer nieuwe acties starten.

Op het einde van de actie zal een “antwoord” geschreven moeten worden op het oorspronkelijke document of een resultaatactie uitgevoerd worden (een betaling bijvoorbeeld).

9.4 Waaron zou het te ontwikkelen pakket gebruikt kunnen worden in zoveel verschillende typen organisaties?

Omdat het pakket zich niet bezig houdt met het eigenlijke werk van het bedrijf. Voor de gemeente Den Briel, houdt het pakket zich niet bezig met de beslissing of de kapvergunning moet worden afgegeven (geen automatisering in die hoek dus), maar draagt het zorg voor de documenten die de betreffende ambtenaar nodig heeft om dat besluit te nemen.

Bij een vorige klant moest er voor elk dossier bijgehouden worden welke functies personen uitvoeren in een dossier. Dat is eigenlijk bij alle dossiers zo, alleen wordt dat niet altijd vastgesteld.

Bij een aanvraag kapvergunning (bij de gemeente Den Briel), is er ook een aanvrager, een expert, een ambtenaar die de zaak behandelt, etc.

Als later mensen direct toegang krijgen tot hun eigen dossiers, is het belangrijk dat deze functies ingevuld worden. Het is de basis om te bepalen tot welke dossiers iemand toegang heeft, en wat deze van dat dossier mag zien.

Ik blijf af van de "wie waar hoe", maar archiveer de documenten en zorg ervoor dat iedereen er makkelijk toegang toe heeft.

Daarnaast wordt veel van het archiveerwerk geautomatiseerd (zoals een goed adresboek), en taken die in alle bedrijven voorkomen.

Een andere reden waarom dit pakket in veel verschillende typen organisaties gebruikt kan worden, is omdat de werkmethoden van bedrijven niet zo veel verschillen als men denkt. Men deelt het bedrijf in afdelingen op. Die afdelingen krijgen taken, maar soms vallen taken onder meerdere afdelingen. Een taak start iedere keer opnieuw, en om dit weer te geven maakt men iedere keer een dossier aan. In dit dossier zitten dan weer documenten (en voor mij hoeven documenten niet altijd tekst op papier te zijn, kan van alles zijn).

De grootste groep uitzonderingen hierop zijn mensen die fysiek werk uitvoeren (zoals het bouwen van een auto). Alhoewel die niet echt anders werken (iedere afdeling heeft z'n taak, verdeelt die onder personen, en iedereen voert het zijne uit), maar er is nauwelijks een oplevering van documenten hierin (en het is niet de bedoeling de auto te archiveren).

Ook in transport is het weer hetzelfde. Ieder transport geeft weer een dossier, alleen komen er in de dossiers maar weinig documenten (één, de CMR).

De boekhouding, is een andere taak, met z'n eigen dossiers. Zolang er geen poging wordt gedaan om in de metadata te zetten wie allemaal die CMR ondertekent heeft (verzender, ontvanger en chauffeur), zijn deze dossiers in wezen niet anders dan die van de boekhouding.

Daarmee komen we gelijk aan de vraag "wat zijn de verschillen tussen bedrijven dan, en zouden deze gegevens verwerkt kunnen worden in de software?"

De verschillen zitten gelijk in de sfeer van bedrijfsautomatisering (wat buiten dit project gehouden is), en voor dit pakket zullen die vooral invloed hebben op de tabellen "dossier" en "documenten" en eventueel in het adresboek en de agenda.

9.5 Discussie: Moet het archief bepalen welke bestandsformaten er gehanteerd worden?

Uit veel literatuur blijkt dat voor archiveringsdoeleinden men een voorkeur heeft gekregen voor bepaalde bestandsformaten[31]. Met name die die zich aan internationale standaarden houden, en specifiek voor XML als opslagformaat.

Buiten de archiefwereld is dit helemaal niet gebruikelijk. Daar hanteert men de bestandsformaten die de software gebruikt waar men mee werkt. Veelal Microsoft-formaten worden hier gehanteerd, omdat MS-Office de de facto standaard is.

Wat zijn de problemen met het opslaan in de facto standaarden voor archivering? Met name voor formaten die “gesloten” zijn (d.w.z. dat niet bekend is hoe ze precies werken), kan het voorkomen dat er in de toekomst geen methode beschikbare is om de inhoud te lezen. Dat zou een verlies van data inhouden.

Oplossingen? In hoofdzaak zijn er twee oplossingen:

- a) Opslaan in een bestand dat we in de toekomst wel kunnen openen
- b) Conversie naar zo'n bestandsformaat.

In dit project is er geen rekening gehouden met de houdbaarheid van de documenten. Wat een toekomstige uitbreiding zou kunnen/moeten zijn is een backup waarin een conversie gebeurt naar betere archiefformaten.

De reden dat de software niet vastlegt welke documentformaten mogen gebruikt worden, is om twee redenen:

- a) Iedere poging daartoe zou een limitering inhouden van het aantal softwareformaten. Juist omdat het dat niet doet kan het alles van het kantoor aan.
- b) Het is niet aan mij om een kantoor te vertellen welke formaten ze wel en niet kunnen gebruiken. Men kan wel vechten tegen de facto formaten, maar uiteindelijk zullen die op vraag van de bedrijven toch ondersteund moeten worden.

Het management van het bedrijf zal zelf de formaten bepalen die binnen het bedrijf gehanteerd worden, en kan beslissen voor archivering betere formaten te hanteren.

9.6 Discussie: Moet het archiefsysteem in XML opgeslagen worden.

Als dit project in een ideale wereld werd uitgevoerd, sloeg dit project alles op in XML-bestanden[2]. Er is in een vroeg stadium voor gekozen om dat niet te doen.

De reden hiervoor is performance en praktische uitvoerbaarheid.

Met name zoekfuncties zouden hierdoor zwaar getroffen worden. Stel er zijn één miljoen documenten en er wordt een zoekopdracht uitgevoerd doorheen al die documenten. Alleen al het openen van alle documenten kost teveel tijd.

Een database staat in het geheugen, en kan onmiddellijk resultaat geven.

Als alle data in losse XML-bestanden staat (1 per document), dan alleen al de access tijd van de harde schijf (tegenwoordig zo'n 4,2 ms gemiddeld op een 7.200 rpm schijf), zou dus 1 miljoen * 4,2ms = 4200 seconden kosten of 1 uur en 10 minuten.

Dergelijke vertragingen zijn onacceptabel.

Hoe lossen we dit op? Om efficiënt en snel te kunnen zoeken moeten de gegevens in een database zitten.

Er zou ook software gebouwd kunnen worden die alle gegevens uit de XML-bestanden leest, en de gegevens in een database plaatst. Nadeel hiervan is dat de data niet altijd up to date is.

Er zou ook software geschreven kunnen worden die de metadata uit de database samenvoegt met de bestanden uit het archief en deze combinatie in een backup in XML-formaat opslaat.

De laatste methode heeft veel voordelen voor dit project, omdat de gewone bestanden door de gebruikers geopend kunnen worden (geen werk dus), en er kan dus snel en efficiënt gewerkt worden.

10 Nieuwe ideeën

In bestaande software zitten naar mijn mening nogal wat tekortkomingen als er naar de aansluiting ervan op het dagelijks functioneren van organisaties gekeken wordt. Daardoor zijn een aantal verbeteringen ten opzichte van bestaande software mogelijk.

1. Is archivering wel een goede naam? De naam zou suggereren dat het hier gaat om het opslaan van een document na z'n actieve leven binnen de organisatie. Dat is hier niet het geval. Een document wordt eerst gearchiveerd, dan gebruikt.
Dit is dus niet een systeem waarmee mensen van buiten de organisatie achteraf de documenten/dossiers in kunnen kijken, maar een systeem dat door de eigen organisatie gebruikt wordt om met de documenten te werken.
2. Via een offerte voor een klant ben ik bij Xtandit[40] terecht gekomen. Hierbij ging het gesprek op een gegeven moment over documentmanagement systemen.
In hun systeem hebben zij het volgende idee verwerkt: Een document dat binnenkomt, kan een reactie starten binnen het systeem. Zij hebben dit uitgewerkt met een programmeertaal binnen het systeem.
Het idee is goed, maar de programmeertaal lijkt me onnodig. Die geeft weliswaar meer flexibiliteit, moeilijker om te bouwen, moeilijker voor de gebruiker. Volgens mij moet er een simpelere methode zijn.
Mogelijkheden kunnen zijn dat als een document gearchiveerd is, er een todo verschijnt bij iemand om een andere brief te schrijven. Een link naar een todo-lijst of agenda is toch iets wat zou moeten.
3. Als in de organisatie tijdwinst gecreëerd moet worden, dan is dit het best te bereiken door documenten automatisch te genereren. Dit zou dus onderdeel van het pakket moeten worden.
4. Een adresboek is nodig. Één dat zowel personen, bedrijven als afdelingen zowel naast elkaar als door elkaar kan hanteren. De links naar personen, bedrijven en afdelingen komen veelvuldig voor.
5. Een nieuwe laag na dossier? Binnen de meeste organisaties worden twee lagen gebruikt: Dossier en document. Ook bij een vorige opdracht hebben mijn collega's en ik dat zo geïmplementeerd, maar dat was nooit naar mijn tevredenheid.
Het nadeel is dan dat er maar één type dossier kan zijn. Dit heeft dus als gevolg dat er maar één type dossier bewaard kan worden. Dit was voor de betreffende organisatie prima, maar theoretisch niet ideaal, omdat bijvoorbeeld hun boekhouding dan niet gearchiveerd zou worden. Elk dossier heeft z'n eigen indeling, soorten documenten, personen die ermee te maken hebben (zoals klanten, aanvragers, schadelijders, crediteuren of debiteuren), maar niet allemaal tegelijk.
Een indeling naar afdeling is niet zo netjes, omdat soms meerdere afdelingen aan eenzelfde dossier werken.
Bij de gemeente De Briel werkte men aan een nieuwe indeling voor dossiers, een indeling die begon met een b en dan een nummer.
Nadere analyse van dat systeem maakte me duidelijk dat het eigenlijk om een "taak" ging. Aan deze taak kunnen meerdere personen of afdelingen gekoppeld worden.
6. Samenwerking: Een klant vroeg of meerdere personen, die elkaar niet zien op kantoor, samen kunnen werken aan een document (versiebeheer).
Hier is over nagedacht. Volgens mij moet het document dan niet als "af" beschouwd worden en dus ook niet in het archief geplaatst worden. Er kan wel een fase voor archiveren gebouwd worden die de versies bijhoudt, en op het einde (als het document verstuurd wordt), dan archiveert.

11 Toevoegbare functionaliteit

11.1 Toegang vreemde organisaties (rollen)

Bij een vorig project moesten er in dossiers vaak bepaalde functies toegekend worden (zoals opdrachtgever) vaak aan mensen buiten de organisatie, maar ook aan mensen binnen de eigen organisatie. Deze “rollen” waren echter vast gedefinieerd en nogal rigide (er waren gewoon vijf tabellen met personen).

Door een systeem te nemen waarin we “functies” aan personen kunnen toebedelen, en deze koppelen aan een dossier (de functie van een persoon binnen een specifiek dossier), maken we alles flexibeler. Hier kunnen we aan een dossier een functie die iemand vervult meegeven, en hieraan ook lees/schrijfrechten aan toekennen.

Er zijn dus een aantal functies maar per dossier kunnen die verschillen. En per documenttype moet vastgesteld worden of deze rol leidt tot leesrechten (mensen van buiten mogen geen metadata beheren).

Voor de volledigheid zouden we een tabel aan kunnen maken waarin een persoon aan een functie gekoppeld wordt. Dit zou dan gebruikt kunnen worden om een lijstje van personen te maken voor functie-indeling. Dit lijkt niet zoveel aan het systeem toe te voegen.

Benodigde data:

Rollen

- Omschrijving

Document rol

- Documenttype
- Rol
- Leesrechten

Dossier rol

- Dossier
- Rol
- Persoon
- Leesrechten

11.2 Samenwerken aan één document.

A) Analyse: Hoe werken mensen samen aan één document?

Op een centrale plaats wordt het document opgeslagen. Meerdere mensen kunnen daarbij, en of men werkt rechtstreeks aan het document, of men downloadt het bestand, werkt het bij, en uploadt het daarna weer naar dezelfde plaats.

B) Problemen?

Er bestaat maar één versie, de laatste.

Bij het uploaden of opslaan, kan werk verloren gaan door het overschrijven.

C) Oplossingen?

Het opslaan, zoeken en weergeven van documenten zijn dingen die het documentsysteem toch al moet kunnen.

Wat we moeten hebben zijn een aantal extra tabellen en extra ruimte op de schijf.

Er moeten meerdere mensen aan éénzelfde document samenwerken.

We hebben een document nodig, waaraan gewerkt wordt.

Dus zal er een tabel document zijn.

De samenwerking zelf, kan men het best zien als een project. Er is een projecttabel nodig.

Het downloaden/uploaden van de documenten moet ook geregistreerd worden.

D) Benodigde data

Document

- Versienummer
- Bestandsnaam
- Auteur

Project

- Titel
- Dossier (waaraan het mogelijk gelinkt is)
- Opdracht (waaraan het mogelijk gelinkt is)
- Individueel?
- Persoon (oprichter project)
- Lock (niet uploaden?)
- Locker (door wie wordt dit gedaan?)
- Locktimer (voor hoe lang?)

Upload/download

- Persoon
- Datum

11.3 Taakgenerator

Probleemstelling:

Net zoals bij Xstandit zou er een methode moeten komen om een reactie te geven op het binnenkomen van een document.

Xstandit deed dit met behulp van een programmeertaal, maar dat is volgens mij niet nodig. Dit omdat eigenlijk wel bekend is wanneer er acties nodig zijn, en er maar een gelimiteerd aantal soorten acties.

Wanneer moet er iets gebeuren: Alleen als een document gearchiveerd wordt.

Door wie? Of door de server, of door een gebruiker.

Als het de gebruiker is, zijn er twee mogelijke plaatsen om dit aan te geven:

- a) Agenda (afspraak maken)
- b) Todo-lijst

Voor de server is het een commandoregel met wat variabelen, en die moet dan real-time worden uitgevoerd.

Benodigde data:

Taak

- Omschrijving
- Commando met variabele parameters
- Opdracht
- Type document

Voor de agenda/todo

- Dossier waarin het document is opgeslagen
- Het document waarover het gaat
- Startdatum
- Datum waarop de afspraak valt, of de todo klaar moet zijn
- Prioriteit

11.4 Document genereren

Probleem? Bouw een functie die mensen veel werk uit handen neemt, en documenten standaardiseert. Met een paar klikken kan met bekende gegevens een document gecreëerd worden zodat de gebruiker deze gegevens niet meer hoeft op te zoeken en over te nemen. Denk hierbij aan het adres waar de brief heen moet, de opmaak, en bij een rapport bijvoorbeeld een basisrapport zodat er al veel gegevens en tekst in staan.

Oplossing? Dit zou te doen moeten zijn. Voor dit project wordt er alleen OpenOffice gehanteerd (maar latere uitbreidingen naar andere formaten is mogelijk). OpenOffice hanteert de standaard OASIS. De OpenOffice.org-implementatie is nogal leuk: een zip van een OASIS-document.

Hiermee zijn oude trucjes zeer bruikbaar, namelijk zoek/vervang.

Fundamenteel is een XML-bestand gewoon een tekstbestand. Met een paar simpele commando's kunnen we daar een zoek/vervang op toepassen. Hiervoor hoeft alleen eerst de zipfile uitgepakt te worden.

De gebruikers kunnen dan een standaarddocument gewoon opmaken in hun tekstverwerker, hierin “markers” (herkenningspunten) plaatsen, en dit op de goede locatie opslaan.

Daarna is het een kwestie van de data aangeven in de database en dan kan het document gelijk gebruikt worden.

Benodigde data:

- Document waarop de zoek/vervang wordt toegepast
- Brontabel en veld uit de database (gebruiker geeft verder aan welk veld)
- Marker (herkenningspunt in de tekst welke vervangen moet worden)

11.5 Toegang tot documenten

Probleem? Hoe te bepalen wie toegang heeft tot een document en hoe doen we dit zonder de veiligheid in het gedrang te brengen.

Oplossing? Een document valt onder een dossier, en een dossier onder een taak. Om niet per dossier te hoeven bepalen wie eraan werkt, doen we dit handiger een laag hoger, namelijk bij taak.

Aan een dossier kunnen meerdere afdelingen werken, of een willekeurige combinatie van afdelingen en personen uit de organisatie.

We kunnen dus een “team” samenstellen, die bestaat uit deze afdelingen/personen, die lees- en schrijfrechten hebben tot alle dossiers onder die bepaalde taak.

Anderen kunnen hierin dan meelesen, maar niet schrijven.

Er zijn taken waarbinnen andere niet mogen meekijken (zoals boekhouding), daar is er een veiligheidsbit geplaatst.

Zo hoeft niemand per dossier aan te geven wie er toegang heeft en hoe. Dit is een belangrijk punt om te vermijden, want in de praktijk schiet zo iets erbij in en brengt op die manier de veiligheid in het geding.

12 Formulieren

Om de interactie tussen gebruiker te regelen zijn er formulieren nodig. Deze zijn op te delen in zes soorten. Ingest om de invoer van documenten te verwerken. Access om diezelfde documenten weer in te kijken. Administratie om de gegevens bij te werken. Beheer/Structuur om de structuur van documenten/dossiers te maken of bij te werken. Opzoeken om te kunnen zoeken op de opgeslagen gegevens.

Daarnaast zal er een zoekstelsel moeten zijn om het stelsel te doorzoeken.

En als laatste moet er een hoeveelheid administratie uitgevoerd kunnen worden. Ook hiervoor zullen formulieren nodig zijn.

Ingest

- Invoer archivaris
- Invoer werknemer (plaatsing in een dossier)
(Praktijkervaring leert dat invoering werknemer eigenlijk moet via een todo in een agenda, met terugkoppeling naar de archivaris. Dit omdat er altijd wel een aantal werknemers het gebruik niet zien zitten of gewoon liever niet teveel werken)
- Invoer werknemer via eigen uploadmethode

Access

- Bekijken document (met metadata)
- Bekijken dossier (met metadata)
- Bekijken opdracht (met metadata)

Administratie

- Wijzigen documentgegevens
- Wijzigen dossiergegevens
- Wijzigen opdrachtgegevens
- Beheer personen/personeel

Beheer structuur

- Aanmaken/wijzigen van opdrachten
- Aanmaken en toekennen van documenttypen
- Teams toevoegen/verwijderen aan opdracht
- Lees/schrijfrechten beheren van dossiers/documenten

Opzoeken

- Overzicht dossier
- Overzicht document
- Overzicht opdracht
- Sneloverzicht opdracht
- sneloverzicht dossier
- Sneloverzicht document
- Zoekstelsel

Praktisch

- Sjablonen om op basis van bekende gegevens documenten te genereren.

13 Te verwerken gegevens

De software zal een hoeveelheid gegevens verwerken, die hier groepsgewijs opgesomt en indien nodig besproken worden.

Document

- Verzender (moet een link zijn naar het adresboek)
- Ontvanger (moet een link zijn naar het adresboek)
- Type document (en deze moet een één op veel relatie zijn om zoveel mogelijk consistente types te hanteren in de organisatie)
- Dossier
- Hoe binnengekomen/verzonden
- Link met vorig document als het een reactie ergens op is
- Naam
- Formaat
- Titel
- Registratiedatum

Dossier

- Titel
- Datum opening/sluiting
- Opdracht
- Bewaartermijn (kan een aanpassing zijn op de bewaartermijn van opdracht)
- Dossiernummer

Opdracht

- Omschrijving
- Bewaartermijn

Team

- Opdracht
- Persoon/afdeling

Documentsoort

- Omschrijving

Dossierfunctie

- Dossier
- Persoon/bedrijf/afdeling
- Functie in dossier (beheerder, opdrachtgever, aanvrager, etc.)
- Leesrechten (met name als het iemand van buiten de eigen organisatie betreft)

Document soort functie

- Documentsoort
- Opdracht
- Functie
- Leesrechten

Samenwerking

- Titel
- Dossier/opdracht
- Individueel?
- Persoon
- Lock
- Locktimer
- Door wie gelockt

Samenwerkingsdocument

- Versienummer
- Samenwerking
- Bestandsnaam
- Auteur
- Datum invoer

Sjablonencreatie

- Sjabloon/document (het document waarin zoek/vervang uitgevoerd moet worden)
- Bron_tabel (bron van het gegeven)
- Bron_veld
- Doel_veld (Welke van de gekregen gegevens is de juiste)
- Marker (plaats in het sjabloon waar ingevoerd moet worden)

Personen

- Naam
- Voornaam
- Voornamen
- Tussenvoegsels
- Voorletters
- Geslacht
- Titel (voor en na)
- Geboortedatum
- Beroep
- Bijnaam
- Login
- Wachtwoord

Bedrijf

- Naam
- BTW-nummer

Afdeling

- Naam

Object1

- Nummer van persoon, bedrijf of afdeling
- Tabel (persoon, bedrijf of afdeling)
- Boomsleutel

Adres

- Straat
- Huisnummer
- Postcode
- Postbus
- Stad

Stad

- Naam
- Briefnaam (om in sjablonen te gebruiken)
- Land
- Beginpostcode (postcode zoekstelsel)
- Eindpostcode (postcode zoekstelsel)
- Provincie (wordt in sommige landen in brieven gebruikt)
- Poststad (wordt in sommige landen in brieven gebruikt)

Provincie

- Naam
- Briefnaam
- Afkorting (sommige landen gebruiken een afkorting in adressen)

Land

- Naam
- Briefnaam
- Telefoonmasker
- Telefooncode
- Postcodemasker
- Adresopbouw (uitsluitend voor gebruik in sjablonen)

E-mail

- Soort e-mailadres (e-mail, ICQ, URL van een website, etc.)
- E-mailadres

Telefoon

- Soort telefoonnummer (fax, GSM, hoofd, privé, etc.)
- Telefoonnummer

14 Ontwikkelkeuzes

14.1 Standaarden

In principe accepteert het archief zelf alle soorten documenten, toch is er voor het creëren van documenten gekozen voor ODF. Dit heeft twee redenen:

- a) ODF is een open standaard[5,17].
- b) ODF is een gezipt XML-formaat. Dit maakt functies hiervoor bouwen relatief makkelijk en onafhankelijk van een softwareleverancier.

Voor het uitwisselen van adresboeken met andere apparatuur is gekozen voor LDAP, omdat subsets hiervan breed ondersteund worden[24].

14.2 Softwarekeuzes

Als besturingssysteem is Linux gekozen. Redenen hiervoor zijn:

- a) Aanschafkosten: Gratis
- b) Veel vrij verkrijgbare software: Helpt bij de ontwikkeling
- c) Veel broncode beschikbaar: Als het toch niet helemaal werkt zoals het in dit project zou moeten, is het aan te passen
- d) Goedkoop onderhoud: Een onbekende eigenschap van Linux: als het eenmaal werkt, blijft het werken

Als gevolg van deze keuze zal de server Linux draaien, maar omdat de interface een webinterface wordt, gaan gebruikers hiervan niet veel merken.

Voor de webserver is er voor AOLserver[1,6] gekozen. Op het internet is er een artikel dat de redenen voor het gebruik van deze webserver perfect verwoord[20].

AOLserver gebruikt als programmeertaal TCL, wat een makkelijke taal is om te gebruiken en om te lezen.

Voor de database is PostgreSQL[7] gekozen omdat deze gratis is, en veel functionaliteit[25] biedt. Andere databases zijn door de keuze van AOLserver[27] makkelijk in te voeren, mits ze volledig SQL ondersteunen.

De grootste gratis concurrent van PostgreSQL is uiteraard MySQL[28,30,38]. Deze is vergeleken met PostgreSQL, en de uitgebreidere functionaliteit van PostgreSQL (stabielere views, check functie op foreign key) hebben mij voor laatsgenoemde doen kiezen[35].

14.3 Opslag van documenten

Er zijn drie methoden om de eigenlijke documenten van het archief op te slaan.

- a) In het zelf te maken systeem opslaan
- b) In een database opslaan
- c) Op de schijf leesbaar opslaan

a) In het zelf te maken systeem opslaan

Als we de eigen bestanden in een zelfgemaakt systeem opslaan, dat wil zeggen een zwarte doos waar een bestand in- en uitgehaald kan worden, geeft dit veel ontwikkelingswerk.

Eerst moet er een invoermethode en een uitvoermethode ontwikkeld worden om de bestanden in en uit het systeem te krijgen. Daarnaast is er niks ontwikkeld om met documenten te werken. De code om een bestand te verplaatsen of te bewerken zal allemaal geschreven moeten worden.

Voordeel is wel dat de controle over de bestanden en wat ermee gebeurt totaal

is. Niemand kan iets met de bestanden doen zonder de invloed van het systeem. Maar dat werkt ook weer als nadeel. Als het binaire opslagbestand corrupt raakt zijn de bestanden niet meer terug te halen.

Voordelen

Totale controle over de bestanden

Bestanden en bijbehorende gegevens worden samen opgeslagen

Nadelen

Alles moet ontwikkeld worden

Opslag in een binair systeem bemoeilijkt eventuele preservatie

b) In een database opslaan

Als we de bestanden in een database opslaan zijn er meer hulpmiddelen

beschikbaar om de bestanden in te voeren en te exporteren. Ook zijn er meer hulpmiddelen aanwezig om deze te bewerken.

De ontwikkeling zal dus korter zijn dan bij a), maar de hulpmiddelen zullen onvolledig zijn en dus nog wel het nodige aan ontwikkeling nodig hebben om echt bruikbaar te zijn. De controle is niet meer totaal omdat een gebruiker met de correcte rechten nog wel direct bij de bestanden kan. Alhoewel een database

beter beschermd is tegen datacorruptie, blijft preservatie nog wel een probleem omdat de documenten in een binair systeem worden opgeslagen.

Voordelen

Ontwikkelingstijd korter dan alles zelf schrijven

Bestanden en bijbehorende gegevens worden samen opgeslagen

Makkelijk te bepalen waar een gebruiker bij mag en waar niet

Nadelen

Nog steeds veel ontwikkelingstijd

Opslag in een binair (proprietary) systeem is een gevaar voor preservatie

c) Op de schijf leesbaar opslaan

Als we de bestanden direct op de harde schijf opslaan hebben we een praktisch onbeperkt aantal hulpmiddelen voor eigenlijk alles wat we kunnen bedenken. Invoer en uitvoer gaat dan vanzelf. De problemen hiermee zijn eigenlijk dat rechten van gebruikers veel moeilijker wordt. Daarnaast is er dan een scheiding tussen de bestanden en de bijbehorende gegevens hetgeen “corruptie” tot gevolg kan hebben.

Voordelen

Minimale ontwikkelingstijd voor omgang met bestanden
Tools voor alles wat we met de bestanden willen doen
Prestaties optimaal

Nadelen

Bijbehorende gegevens gescheiden van documenten
Gebruikersrechten op de bestanden moeilijk goed te krijgen

Conclusie

De keuze is om de documenten in het bestandssysteem op te slaan. De doorslaggevende reden is dat de documenten anders zeer ontoegankelijk worden voor de gebruikers, omdat software de documenten in de regel alleen vanuit het bestandssysteem kan openen. Om bij de documenten te komen zouden de documenten bij enige andere keuze eerst naar het bestandssysteem gehaald moeten worden voordat zij bruikbaar zijn, wat extra ontwikkelingstijd en minder gebruiksgemak met zich mee zou brengen. Dit wordt als een te groot nadeel ervaren.

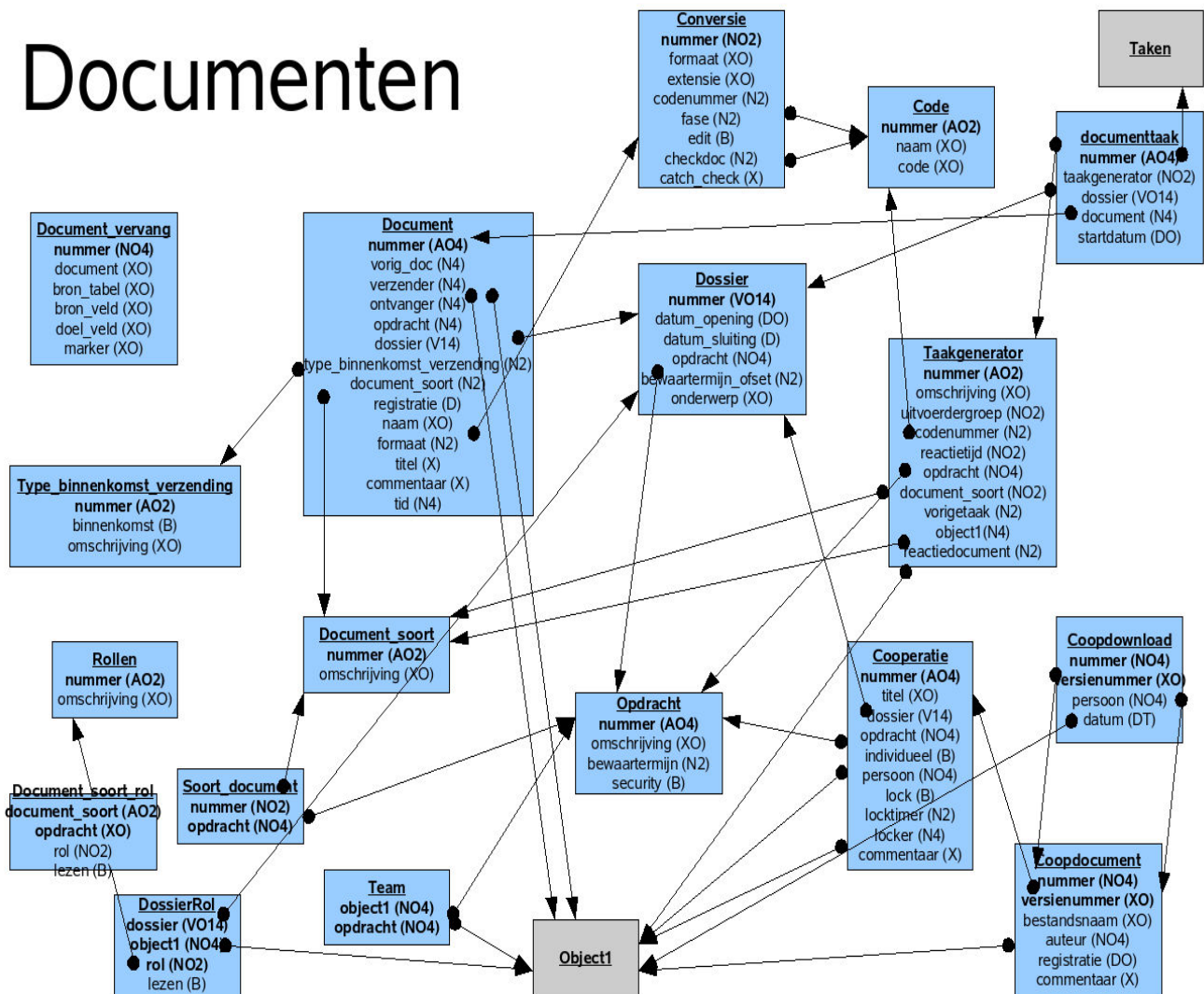
15 Het documentensysteem

15.1 Ontwerp

Document, dossier en opdracht zijn de centrale onderdelen van het geheel. Daarnaast zijn er een aantal die gegevens toevoegen aan deze. We ontwerpen een database-structuur. Een overzicht van de structuur wordt weergegeven in Figuur 15.1. Voor de structuur van het documentsysteem zijn drie tabellen belangrijk.

De meeste andere tabellen zijn om functionaliteit toe te voegen aan het geheel.

Documenten



Figuur 15.1 Databasestructuur archief

Document

Een systeem dat documenten bewaart, heeft uiteraard een tabel om de gegevens van de documenten in op te slaan.

Hierin zitten een aantal foreign keys. De verzender en de ontvanger (een document kan beiden hebben), zijn verwijzingen naar het adresboek.

Er wordt zowel de methode van verzending (brief, fax, e-mail, etc.) bijgehouden (met daarin of het ontvangen danwel verzonden document is) als een type document (factuur, aanvraag vergunning, etc.).

Dossier

Alle documenten vallen in een dossier. Voor een dossier zijn er maar weinig gegevens die echt bijgehouden moeten worden. De datum dat het dossier gestart is, alsook de datum dat een dossier gesloten is.

Daarnaast is er een offset voor de bewaartermijn die gebruikelijk is bij dit type dossier.

Opdracht

Alle dossier vallen onder een opdracht. Dit is een “taak” binnen de organisatie en als dusdanig niet persé een afdeling. Dit omdat een dossier soms door meerdere afdelingen van de organisatie behandeld wordt.

De belangrijkste gegevens van een opdracht zijn security (een boolean vlag die bepaalt of ook anderen binnen de organisatie in deze dossiers mogen kijken) en een bewaartermijn. Dit laatste is ingevoegd op vraag van Van Lee & Van Everdingen. Alle dossiers hebben een bewaartermijn, maar de lengte kon variëren afhankelijk van het “type” dossier. Dit is dus eigenlijk per opdracht.

Deze drie tabellen zijn de basis van het documenten systeem, maar daarnaast zijn er nog een hoop tabellen voor gebruik van het systeem en toevoeging van functionaliteit.

Team

Zoals al vaker vermeld wordt een opdracht niet aan één afdeling toebedeeld. Hiervoor wordt een “team” samengesteld, dat kan bestaan uit individuen en afdelingen. Dit is gewoon een veel-op-veel relatie tussen afdelingen en personen enerzijds, en opdrachten anderzijds.

DossierRol

Binnen een dossier kunnen bepaalde personen van zowel binnen de organisatie als daarbuiten een functie vervullen. Denk hierbij aan een “aanvraag bouwvergunning”. Binnen dat dossier is er iemand die het dossier beheert, de aanvrager, de verlener van de vergunning, etc. De *lezen* boolean bepaalt of deze persoon via zijn rol in het dossier rechten heeft om erin te lezen (met name bedoeld voor personen buiten de eigen organisatie).

Via de tabel DocumentSoortRol wordt bepaald welke documenten een persoon buiten de organisatie via combinaties van type_document, opdracht en rol in het dossier mag inkijken.

Coöperatie

Binnen een project of opdracht kan het zijn dat er een document moet worden samengesteld waaraan meerdere personen werken.

Er is bewust voor gekozen om de verschillende versies van een document niet te archiveren. In plaats daarvan wordt er los een “coöperatie” opgestart die de versies van het te ontwikkelen document bijhoudt. Na voltooiing kan ervoor gekozen worden om het gehele document-traject weg te gooien.

De definitieve versie kan op het einde van het traject gearchiveerd worden in het archief.

Hiervoor zijn drie tabellen aangemaakt. De eerste is om elk document te registreren. Daarnaast is er een tabel om bij te houden welke versies door elke persoon zijn gedownload (oftewel bekeken), en een tabel voor het “project”.

Taakgenerator

Het verplaatsen van een document in het archief (plaatsen in een dossier, zowel voor ontvangst als voor verzending), kan leiden tot een “taak”. Het kan zijn dat de computer deze moet uitvoeren danwel een persoon. Een computeropdracht kan zijn het uitprinten van een te verzenden document.

Een taak voor een persoon kan zijn een antwoord opstellen op een klacht van een klant.

De tabel kan beiden bevatten. Namelijk een regel code die op de command-prompt kan uitgevoerd worden (bijvoorbeeld een printcommando om het document te printen in de postkamer).

Daarnaast is het ook mogelijk om een todo op te stellen. In de todo komt dan de tijd die de ambtenaar hiervoor krijgt (reactietijd), een omschrijving, een link naar het document, etc.

Het veld vorigetaak is voor de mogelijkheid dat als een taak gestart wordt door het beëindigen van een vorige taak.

Voor elke todo die op deze manier gestart wordt, wordt een “documenttaak” opgestart. Deze maakt het mogelijk dat de archivaris controleert welke taken gestart zijn, en of deze ook uitgevoerd worden.

Documentautomatisering

Het archief bevat allerlei gegevens die gebruikt kunnen worden om nieuwe documenten al gedeeltelijk in te vullen. Met name het adresboek is bijzonder handig hiervoor.

Document_vervang (nummer, document, bron_tabel, bron_veld, doel_veld, marker).

Deze bepaalt wat er moet gebeuren bij een zoek/vervang. XML- documenten zijn gewoon ASCII-leesbaar. Hierdoor zijn de oude trucjes bruikbaar om nieuwe documenten te genereren. ODF is uiteindelijk een gezip XML-document.

Document is de bestandsnaam van het bron-document.

Bron_tabel is de naam van de tabel waarvan de informatie moet worden ingevuld, bijvoorbeeld “persoon”.

Bron_veld is de naam van het veld nodig uit de bron_tabel. Via de software zal wel de gebruiker naar het unieke nummer gevraagd moeten worden.

De software zal per tabel de primary key nodig hebben, mogelijk moet dat er in de toekomst nog in. Doch, ook als dat gebeurt blijven er een aantal problemen hangen die gegevens uit willekeurige tabellen halen niet mogelijk maken. De tabel zal in ieder geval individueel aangesproken moeten worden door de software.

15.2 Gebruik van het archief

Invoer

De invoer bestaat uit twee fases. De eerste fase is voor de archivaris. De tweede fase is voor het team van een opdracht.

Dit is om twee redenen gedaan, en beiden hebben te maken met snelheid.

a) Zoveel mogelijk is er gewerkt met uitklaplijsten. Die zijn snel in gebruik (voor de gebruikers, niet voor de computer), en correct omdat het tikfouten uitsluit.

Maar er moet eerst een opdracht gekozen worden, en pas dan een dossier (beiden een uitklaplijst) om de lijst van dossier te vullen vanwege het gebruik van HTML (het kan met JavaScript zonder herladen, maar dan wordt alles trager, en vanwege veiligheid is alle dossiernamen opsturen niet wenselijk).

b) Bij de gemeente Den Briel beheert de archivaris de gegevens van de documenten/dossiers (en vult deze volledig in). Bij een bedrijf dat ik eerder bestudeerd heb, vulde een secretaresse de dossiergegevens in, maar viel de verantwoordelijkheid over de dossiers onder de beheerders ervan. Deze moesten de documenten uiteindelijk dus goedkeuren.

Hieraan was de secretaresse bijzonder veel tijd kwijt omdat ze moest uitzoeken in welk dossier een document moest komen, dit terwijl ze zelf onbekend was met de dossiers en hun specifieke inhoud.

In deze eerste fase van invoer van documenten, geeft de archivaris in hoe het document is binnengekomen (brief per post, e-mail, fax, etc.), de opdracht, de verzender en eventueel het document waarnaar dit document refereert. Verzender en vorig document zoeken gaat via een speciale bladzijde die zwaar gebruikt maakt van de searchengine.

In de tweede fase (deze opzet is een voorlopige oplossing, uiteindelijk is het de bedoeling dat de tweede fase gekoppeld worden aan todo's in de agenda) geeft een teamlid in wat voor type document het is, en in welk dossier het thuishoort.

Opdrachten

Er zijn drie verschillende schermen om de gegevens van de opdracht te zien:

a) Alleen lezen

Als de gebruiker uitsluitend leesrechten heeft binnen deze opdracht, krijgt deze het volgende scherm te zien:



Gegevens tonen van een Opdracht

Algemene gegevens

Nummer: 10
Omschrijving: Klanten beheer
Bewaartermijn: 99
Bewaartermijn: f

[Aanmaken samenwerking](#)

Dossiers van deze Opdracht

Nummer	Onderwerp	Bewaartermijn	Datum opening	Datum Sluiting
01020090825001	Nieuw dossier		2009-08-25	
01020090825002	Nieuw dossier		2009-08-25	

Figuur 15.2 Opdracht lezen

Hierin kan de gebruiker de gegevens van deze opdracht zien, alsook een overzichtje van de dossiers die onder deze opdracht vallen.

b) Schrijven: Leden van het team die deze opdrachten beheren, hebben ook schrijfrechten in een aantal gegevens van de opdracht.



Gegevens tonen van een Opdracht

Algemene gegevens

Nummer: 10
Omschrijving:
Bewaartermijn:
Bewaartermijn:

Figuur 15.3 Opdracht schrijven

c) Beheerder van de structuur: In dit bedrijf zijn dat de leden van de afdeling "archivaris". Deze archiveren de documenten, maar beheren ook de "structuur" van het opslaan van documenten.

Bij gemeenten (alsook bijvoorbeeld bij de TU) is dit de implementatie van het Documentair Structuur Plan. De archivaris bepaalt dus niet alleen welke opdrachten er zijn, maar ook welke documenttypen er in de dossiers zitten, alsook de samenstelling van het team.

Alle documentsoorten die voorkomen in deze Opdracht	
Antwoord aan extern	Verwijder
Vraag aan extern	Verwijder
Antwoord van extern	Verwijder
Vraag van extern	Verwijder
Offerte	Verwijder
Verslag gesprek	Verwijder
Contract	Verwijder

Document

Figuur 15.4 Documentsoorten toevoegen

Dossier

Als een persoon minimaal leestoeegang heeft tot een opdracht, kan deze in deze opdracht alle dossiers zien. Net zoals bij het overzichtsblad van opdrachten, is het linkje links in het scherm naar een overzichtsscherm voor documenten, en de knop rechts voor beheer van het dossier.

In het beheer van een dossier zijn er in tegenstelling tot opdracht, maar twee schermen. Dit omdat de leden van het team de beheerders zijn van een dossier, en de enigen die schrijfrechten hebben.

a) Alleen lezen

Iemand die uitsluitend leestoeegang heeft tot het dossier kan uitsluitend de gegevens bekijken. De documenten staan aangegeven en er kan naar doorgedrukt worden.

Gegevens tonen van een Dossier

Algemene gegevens

Nummer 01020090825001
Onderwerp Nieuw dossier

Archivering gegevens

Datum Opening Dossier 25/08/2009
Datum Sluiting Dossier //
Bewaartermijn Ofset (Dit is het verschil met de bewaartermijn zoals omsch)
Totale Bewaartermijn 99

Figuur 15.5 Dossier lezen

b) Beheerders-scherm

Iemand die schrijftoegang heeft, is dus automatisch een beheerder van een dossier. Deze kan dus het dossier afsluiten, alsook documenten direct toevoegen (zonder tussenkomst van de archivaris).

Gegevens tonen van een Dossier

Algemene gegevens

Nummer 01020090825001
Onderwerp

Archivering gegevens

Datum Opening Dossier 25/08/2009
Datum Sluiting Dossier
Bewaartermijn Ofset (Dit is het verschil met de bewaartermijn zoals omschr...
Totale Bewaartermijn 99

Figuur 15.6 Dossier schrijven

Informatie over de ontbrekende documenten in een dossier. Aangezien elk type dossier vanuit opdrachten maar een vast aantal soorten documenten kan hebben, wordt in het dossiermenu weergegeven welke documentsoorten in dit dossier nog ontbreken.

Ontbrekende documenten: (Antwoord aan extern, Antwoord van extern, Contract, Offerte, Verslag gesprek, Vraag aan extern, Vraag van extern)

Figuur 15.7 Ontbrekende documenten

Dit zou in de toekomst bij afsluiting ook doorgegeven kunnen worden aan de archivaris (voor controledoeleinden).

Documenten

In documenten zijn er weer drie mogelijke schermen.

a) alleen lezen

Deze krijgt de gegevens van het document te zien zoals Documentnaam, Dossier, Documentsoort, commentaar, etc.

b) Beheerder van het dossier

Deze krijgen een extra gedeelte om een antwoord te genereren en om dit te uploaden.

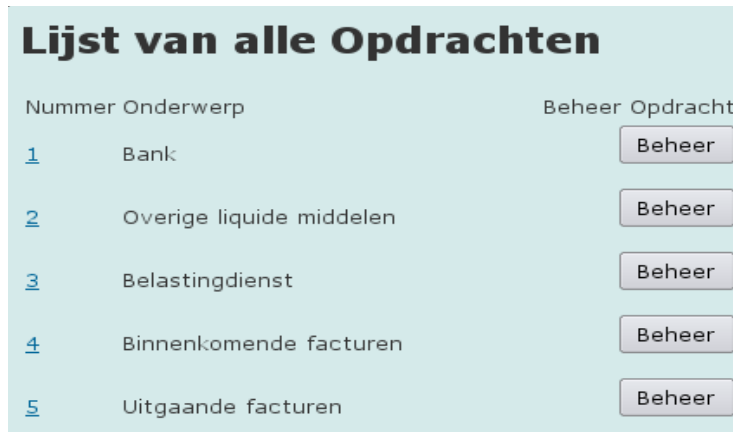
c) Archivaris

Deze krijgt naast het beheerders-scherm ook nog de mogelijkheid om de gegevens die ingevoerd zijn bij het archiveren en te maken hebben met verzending van het document, te wijzigen (in geval deze gegevens fouten bevatten).

Overzichten

a) Opdrachtoverzicht

Naar gelang de leesrechten die de gebruiker heeft, kan deze in de opdrachtenlijst de verschillende opdrachten zien. Het linkje met nummer (het nummer van de opdracht) wijst de persoon door naar eenzelfde overzichtje voor dossiers. De knop rechts gaat naar de detailpagina van een opdracht.



Nummer	Onderwerp	Beheer Opdracht
1	Bank	Beheer
2	Overige liquide middelen	Beheer
3	Belastingdienst	Beheer
4	Binnenkomende facturen	Beheer
5	Uitgaande facturen	Beheer

Figuur 15.8 Overzicht opdrachten

b) Dossieroverzicht

Naargelang de leesrechten die de gebruiker heeft, kan deze in de dossierlijst de verschillende dossiers zien. Het linkje met nummer (het nummer van de opdracht) wijst de persoon door naar eenzelfde overzichtje voor documenten. De knop rechts gaat naar de detailpagina van een dossier.

c) Documentenoverzicht

Naargelang de leesrechten die de gebruiker heeft, kan deze in de documentenlijst de verschillende documenten zien. Het linkje met nummer (het nummer van document) wijst de persoon door naar het document. Dit wordt hier inline geopend, maar alleen als het document één bladzijde bevat (dat wil zeggen één bestand, niet zozeer één bladzijde in een bestand zoals een multipage PDF of TIFF). De knop rechts gaat naar de detailpagina van een document.

varying)::"bit"))::bit varying)) AND (objectbedrijf.tabelnaam = 'bedrijf::text));

De gegevens bij persoon, bedrijf en afdeling zijn de bijbehorende gegevens uit de respectievelijke tabellen. Mocht het bij toekomstige projecten nodig zijn deze uit te breiden, dan kan dit zonder veel te hoeven aanpassen.

Een persoon kan bij meerdere afdelingen/bedrijven thuishoren. Bij elke afdeling kan deze persoon een andere functie uitvoeren voor hetzelfde of verschillende bedrijven (meerdere petten ophebben als het ware). Via instantie kan een persoon meerdere object1-entiteiten hebben, die allen terugwijzen naar dezelfde persoon.

Privé of zakelijk

Nu moet er in een goed adresboek een verschil gemaakt kunnen worden tussen een zakelijk contact (gebruikt doorheen het bedrijf), of een contact van de gebruiker zelf. Hiervoor is de tabel invul (bepaalt zakelijk of privé) en de tabel categorie (geeft een categorienaam aan het contact).

Gegevens van personen/bedrijven/afdeling

Iedere persoon heeft wel één of meerdere telefoonnummers, adressen of e-mailadressen. Deze moeten correct bijgehouden kunnen worden.

Er zijn dus via object1 koppelingen naar een adrestabel (zowel een persoon, bedrijf als een afdeling kunnen één of meerdere adressen hebben), een emailtabel of een telefoontabel. Van elk van deze kunnen meerdere soorten zijn. Iemand kan een privételefoonnummer hebben, of juist een zakelijk. Maar ook een fax. Zo kan er onder e-mail ook ICQ-adressen opgeslagen worden, etc.

Opbouw van een adres

Een adres moet geldig ingevuld kunnen worden voor de gehele wereld, en niet alleen voor Nederland. Allereerst hebben we de tabel adres met de gebruikelijke gegevens, alleen geen land. De stad wijst naar een andere tabel "stad" met de gegevens die belangrijk zijn voor een stad. Het basisidee is dat een stad altijd in één land valt. Hierop zijn mij geen uitzonderingen bekend, maar indien die er zijn zal de stad als twee steden gezien moeten worden, elk met z'n eigen land.

Door een tabel voor steden te gebruiken zorgen we ervoor dat de schrijfwijze consequent blijft, en er ook de juiste schrijfwijze gehanteerd kan worden voor op brieven (nationaal of internationaal).

Een stad kan onder een provincie vallen. Nu is er niet voor gekozen om provincies hiërarchisch op te zetten. Zou eigenlijk wel moeten omdat een aantal landen in de wereld dat hanteren. Bijvoorbeeld Italië is ingedeeld in regio's, die weer provincies bevatten.

Nu zijn er een aantal landen die het zoeken op postcode mogelijk maken en zelfs specifiek gebruiken. Door andere landen wordt dit enigszins tegengewerkt.

Nederland bijvoorbeeld maakt het mogelijk dat er uit een postcode zowel de stad als de straat af te leiden zijn.

In België is er met een postcode niet meer dan een stad bekend, maar er zijn postcodes die naar meerdere plaatsen wijzen, alsook steden die meerdere postcodes hebben. Het postcodezoeksysteem komt aan beide tegemoet. Beter nog, het zoekstelsel leert zelf bij door postcodes in volgorde te zetten en te zoeken tussen postcodes van groot naar klein of omgekeerd.

Relaties tussen personen

Er is een tabel “relatie” en “soort_relatie” die de relaties kan beschrijven tussen personen, zoals getrouwd, kind van, etc.

Afspraakbeheerder

Een persoon kan iemand anders aanwijzen (een secretaresse ofzo) die z'n afspraken mee kan beheren.

Notities

Een klein notitieblokje om dingetjes te noteren die niet vergeten mogen worden. Notities staan hier tussen de tabellen van contact omdat de gegevens nou eenmaal aan een persoon gekoppeld zijn, en niet aan een afspraak.

Werkdagen en vakantiedagen

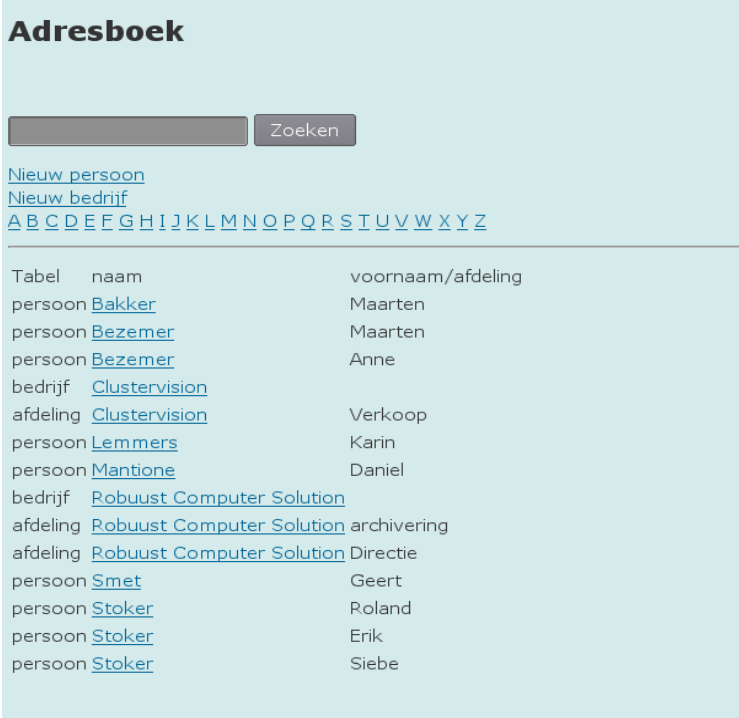
Zoals hier geobserveerd kan worden, hangt de tabel werkdagen onder een persoon, en vakantiedagen onder object1. Beiden zijn in de basis bedoeld om afspraken te kunnen maken op dagen dat de persoon werkt. Per persoon kunnen zijn werkdagen gedefinieerd worden. Dit zijn bijvoorbeeld de werkdagen dat er gewerkt wordt, met inbegrip van de begin/eindtijd van de werkdag (of gedeelte ervan).

De vakantiedagen van een persoon zijn een som van de vakantiedagen van deze persoon samen met die van z'n afdeling en bedrijf (als het bedrijf dicht is, heeft de persoon automatisch vakantie).

16.2 Gebruik van het adresboek

Overzicht

De eerste bladzijde is een overzichtspagina van de agenda. U kunt hierin zoeken naar een persoon, bedrijf of afdeling.



Tabel	naam	voornaam/afdeling
persoon	Bakker	Maarten
persoon	Bezemer	Maarten
persoon	Bezemer	Anne
bedrijf	Clustervision	
afdeling	Clustervision	Verkoop
persoon	Lemmers	Karin
persoon	Mantione	Daniel
bedrijf	Robuust Computer Solution	
afdeling	Robuust Computer Solution	archivering
afdeling	Robuust Computer Solution	Directie
persoon	Smet	Geert
persoon	Stoker	Roland
persoon	Stoker	Erik
persoon	Stoker	Siebe

Figuur 16.2 Overzicht adresboek

Bovenaan staat een zoekbalk voor de naam van de persoon, bedrijf of afdeling. Direct eronder is de mogelijkheid een nieuw persoon of bedrijf aan te maken. (een afdeling kan alleen binnen een bedrijf aangemaakt worden)

Direct daaronder ziet u de mogelijkheid om alle personen/bedrijven of afdelingen te tonen op basis van hun beginletter.

Het gedeelte eronder bevat de gevonden resultaten.

De schermen van persoon, bedrijf of afdeling bevatten drie mogelijke situaties:

- Het object is door u ingevoerd. U hebt dan volledige lees en schrijfrechten. U kunt ook bepalen of andere personen binnen uw bedrijf dit object kunnen lezen danwel wijzigingen kunnen aanbrengen.
- U hebt lees- en schrijfrechten, maar het object is niet door u ingevoerd. U kunt dan wijzigingen aanbrengen in de gegevens van het object, maar niet bepalen of andere personen lees- danwel schrijfrechten krijgen.
- U hebt uitsluitend leesrechten.

Als u geen leesrechten hebt, kunt u het object niet zien en bent u zich er niet van bewust dat dit er is.

Persoon bekijken

Bij een persoon zijn er drie gedeeltes:

a) De algemene gegevens

Deze bevatten de naam, voornaam, geboortedatum, etc.

Daarnaast ook de bepaling van toegangsrechten (indien u deze mag wijzigen) en de categorie-indeling van deze persoon.

b) De bedrijfsgegevens van deze persoon. Hierin kunt u deze persoon toevoegen aan een bedrijf en aan een afdeling binnen dit bedrijf.

c) De communicatiegegevens zoals e-mail, fax, telefoon en adressen.

Roland Stoker BSc

Naam: GeboorteDatum: 29 februari 1976
Tussenvoegsels: Geslacht: Man
Voornaam: Login:
voornamen: Paswoord:
Initialen:
Roepnaam/Bijnaam:
Titel voor/na: ✓

[Robuust Computer Solutions](#) [Directie](#) ✗
[archivering](#) ✗

2007-04-08 ✓ ✗
2007-05-17 ✓ ✗
 ✓
 08-04-2007 +32 (0) 052-462011 ✓ ✗

Figuur 16.3: Beheer persoon

Voor een bedrijf zien we het volgende scherm:

Robuust Computer Solutions

naam:
BTW-nummer: ✓

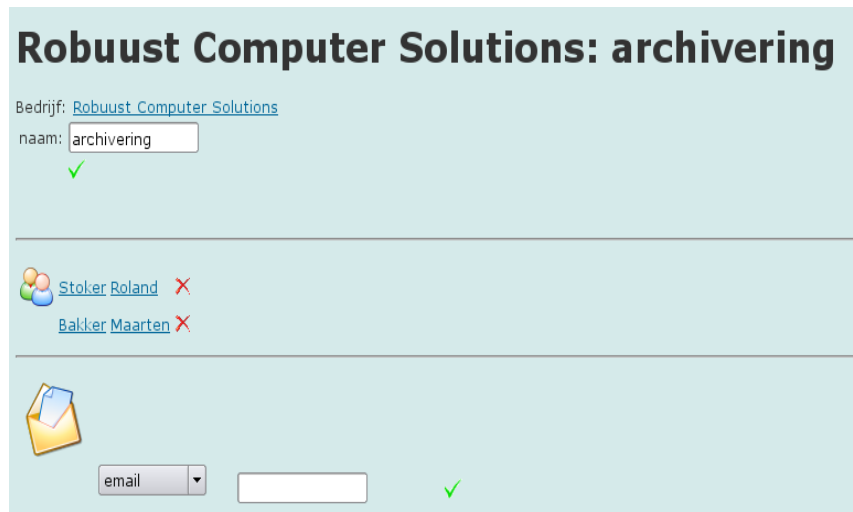
[archivering](#)
[Directie](#)

2007-04-08 ✓ ✗
2007-04-08 ✓ ✗

Figuur 16.4: Beheer bedrijf

De indeling is ongeveer dezelfde als bij persoon, alleen kunnen er afdelingen gemaakt worden.

Bij afdeling zien we de volgende indeling:



Figuur 16.5: Beheer afdeling

Hier worden ook de personeelsleden van deze afdeling getoond. Deze kunnen eventueel uit de afdeling verwijderd worden.

17 Veiligheid

Veiligheid moet generiek werken; per persoon aangeven bij welke documenten deze mag is geen goed idee. De hoeveelheid werk die daarvoor nodig is, maakt dat dat in de praktijk niet uitgevoerd zal worden.

Er is voor gekozen om veiligheid via views te realiseren. In de tabellen zijn merkpunten voor veiligheid ingebouwd waarop kan aangegeven worden hoe "sensitive" iets is. Hierop kan de view verder bouwen.

Zo is er bij de tabel opdracht een security boolean, bij document_soort_rol een lezen vinkje, en zo komt het nog op meer plaatsen voor.

Voor documenten krijgen we dan ongeveer de volgende view (alleen het security boolean is hierin verwerkt)

Deze views lijken zeer complex, maar dat valt op zich mee. Het is in feite een combinatie (UNION) van individuele mogelijkheden waarom iemand lees- danwel schrijfrechten zou hebben op bijvoorbeeld een document.

Voor de software is het daarna een stuk makkelijker. Deze voert gewoon een query uit op de view (in plaats van de verschillende tabellen te doorzoeken), en heeft dus met een korte query gelijk het antwoord.

Mocht de view toch niet helemaal doen wat men er van zou wensen, dan is de view eenvoudig aan te passen en hoeft zowel de code van de software alsook de structuur van de database niet aangepast te worden.

View veiligheid documenten:

```
SELECT 'r' AS privilege, opdracht.nummer AS opdracht, dossier.nummer AS dossier,
document.nummer AS document, persoon_opdracht2.persoon, persoon_opdracht2.login FROM
opdracht, dossier, document, persoon_opdracht2 WHERE opdracht.nummer = dossier.opdracht AND
dossier.nummer = document.dossier AND (opdracht.security = false OR opdracht.nummer =
persoon_opdracht2.opdracht) UNION
SELECT 'r' AS privilege, opdracht.nummer AS opdracht, dossier.nummer AS dossier, " AS document,
persoon_opdracht2.persoon, persoon_opdracht2.login FROM opdracht, dossier, persoon_opdracht2
WHERE opdracht.nummer = dossier.opdracht AND (opdracht.security = false OR opdracht.nummer
= persoon_opdracht2.opdracht) UNION
SELECT 'r' AS privilege, opdracht.nummer AS opdracht, " AS dossier, " AS document,
persoon_opdracht2.persoon, persoon_opdracht2.login FROM persoon_opdracht2, opdracht WHERE
(opdracht.security = false OR opdracht.nummer = persoon_opdracht2.opdracht) UNION
SELECT veiligheid_write.privilege, veiligheid_write.opdracht, veiligheid_write.dossier,
veiligheid_write.document, veiligheid_write.persoon, veiligheid_write.login FROM veiligheid_write
UNION
SELECT 'r' AS privilege, veiligheid_write.opdracht, veiligheid_write.dossier,
veiligheid_write.document, veiligheid_write.persoon, veiligheid_write.login FROM veiligheid_write;
```

De eerste 3 queries zoeken uit wie leestoegang heeft tot document, dossier en opdracht.

De voorlaatste query van deze view roept de view veiligheid_write aan (die doet ongeveer hetzelfde als deze maar bepaald wie ergens schrijftoegang heeft), en geeft deze schrijftoegang.

De laatste query roept weer veiligheid_write aan, maar geeft leestoegang (wie schrijftoegang heeft, heeft automatisch leestoegang).

Zo zijn er ook views voor adresboek, toegang voor personen buiten de organisatie en agenda.

Omdat deze views een query zijn, is veiligheid structureel aan te passen, zonder programmacode te veranderen of de databasestructuur aan te passen.

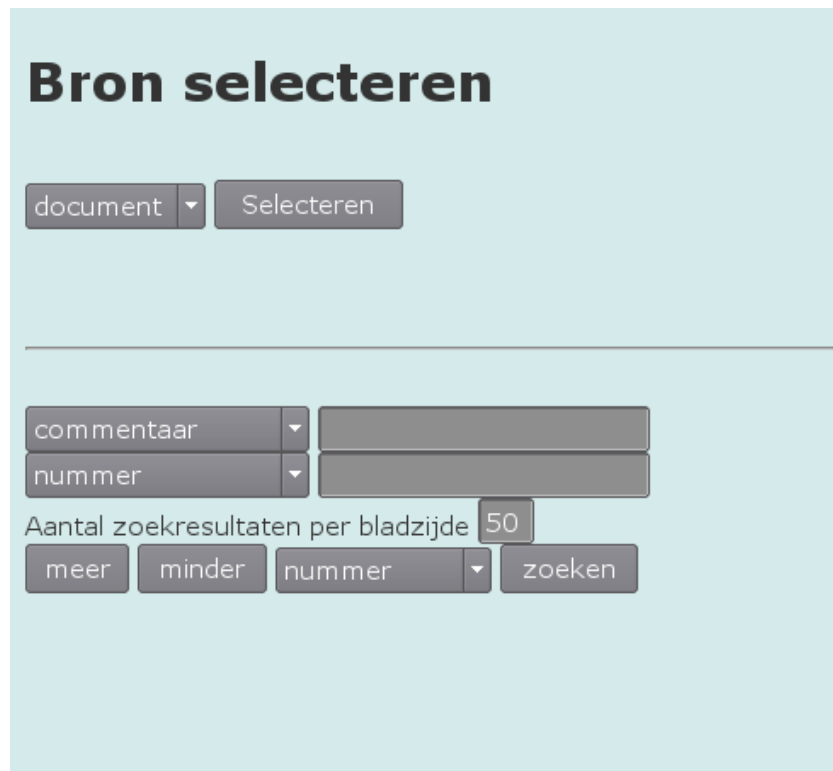
Hiermee is veiligheid redelijk makkelijk aan te passen, mocht een organisatie afwijkende veiligheidswensen hebben.

18 Zoekmachine

Ieder archief heeft een goede zoekmachine nodig. Het basisidee bij deze zoekmachine is dat overal op gezocht moet kunnen worden.

18.1 Fases in de zoekmachine

a) Interface



The screenshot shows a search interface with a light blue background. At the top, the title "Bron selecteren" is displayed in bold black text. Below the title, there is a dropdown menu with "document" selected and a "Selecteren" button. A horizontal line separates this section from the search input area. In the search area, there are two dropdown menus: "commentaar" and "nummer", each followed by a text input field. Below these, there is a label "Aantal zoekresultaten per bladzijde" with a dropdown menu showing "50". At the bottom, there are three buttons: "meer", "minder", and "nummer" (with a dropdown arrow), followed by a "zoeken" button.

Figuur 18.1 Zoekmachine

De interface bestaat uit drie onderdelen.

Allereerst moet er een "bron" gekozen worden. Deze bepaalt eigenlijk in welke tabel van de database gezocht moet worden. Dat kan document, persoon, dossier, agenda, etc. zijn.

Daarna moeten de velden ingevuld worden. Per bron zijn er een aantal velden opgegeven. Dit zijn de velden uit de tabel, maar ook uit gelinkte tabellen door middel van foreign keys.

Als laatste is er een mogelijkheid om de hoeveelheid resultaten te limiteren en een sortingsveld aan te geven.

b) Interface naar velden waarop gezocht moet worden en hun inhoud

Het scherm moet weer afgevangen worden door AOLserver. Voor de ingevulde velden moet uitgezocht worden om welke velden en in welke tabellen dit gaat. Deze worden 1 op 1 vergeleken met mogelijke velden en indien ze overeenkomen wordt het zoekveld ingevuld.

Van de ingevulde zoekvelden moet nu een query opgesteld worden.

c) Velden en hun inhoud vertalen naar een query

Per tabel is er een functie die de zoekvelden omzet in een geldige query. Eerst stellen we “select” en “from” phrases op met bekende gegevens en afhankelijk waarvoor de query is.

Dan vullen we de “where” phrase op door voor alle ingevulde gegevens in te vullen.

Een “einde” phrase houdt sortering (order by) of limitering (limit by) bij. Deze phrases worden als laatste samengevoegd tot een query.

d) Subqueries

Zoals eerder gezegd, kunnen er ook gegevens staan in andere tabellen. Dit wordt gedaan door in de where clause van een query een subquery in te vullen. Deze wordt opgesteld door de zoekfunctie van deze tabel op dezelfde manier als wanneer er voor die tabel een search gestart zou zijn.

e) Resultaat weergeven

nummer	naam	voornaam	geboortedatum
1	Stoker	Roland	1976-02-29
2	Bakker	Maarten	1970-01-01
3	Bezemer	Maarten	1970-01-01
4	Stoker	Erik	1979-05-02
5	Stoker	Siebe	1970-01-01
6	Bezemer	Anne	1970-05-01
7	Lemmers	Karin	1970-01-01
8	Smet	Geert	1970-01-01
9	Mantione	Daniel	1977-04-01

Bron selecteren

persoon

naam

Aantal zoekresultaten per bladzijde

nummer

Figuur 18.2 Resultaten zoekactie

De interface krijgt uiteindelijk dus een query terug. Deze ziet er als volgt uit:

```
SELECT DISTINCT persoon.nummer, persoon.naam, persoon.voornaam, persoon.geboortedatum  
FROM persoon WHERE UPPER(naam) LIKE '%%' ORDER BY nummer LIMIT 50 OFFSET 0;
```

De resultaten hiervan moeten door de interface verwerkt worden tot een scherm en getoond worden aan de gebruiker.

18.2 Werking

Eerst gebeurt alles in zoeken.adp. Dit is een bladzijde die onderdeel is van de website. In deze bladzijde wordt allereerst de webpagina opgemaakt. Als de gebruiker “zoeken” aanklikt wordt de functie zoeken gebruikt.

In de functie zoeken wordt de zoekrijen omgezet naar een lijst, en deze opgestuurd naar de functie zoekoplossen (geschreven in TCL):

```
proc zoeken {f aantal sorteer aantalres bladzijde db} {
  global bron
  set result "<H1>Resultaten zoekopdracht</H1>
  <FORM action='zoeken.adp' method='post'>
  <INPUT type='hidden' name='bron' value='$bron'>
  <INPUT type='hidden' name='aantal' value='$aantal'>
  <INPUT type='hidden' name='sorteer' value='$sorteer'>
  <INPUT type='hidden' name='aantalres' value='$aantalres'>"
  set rij 1
  set opzoeklijst ""
  while {$rij <= $aantal} {
    set {invoer$rij} ""
    set test ""
    set {zoekfunctie$rij} ""
    set invoer [getfromset $f invoer$rij]
    set zoekfunctie [getfromset $f zoekfunctie$rij]
    lappend test "$invoer" "$zoekfunctie"
    lappend opzoeklijst $test
    set rij [expr $rij + 1]
  }
  set result [zoekoplossen $bron $opzoeklijst $bladzijde $aantalres $sorteer]
  ns_puts $result
}
```

Zoekoplossen staat niet in de website, maar in de modules van AOLserver. Dit houdt in dat deze functie altijd in het geheugen staat.

In zoekoplossen zijn alle mogelijke vragen bekend. Hier wordt de lijst omgezet in een tabel waarin gezocht moet worden en de velden waarin dingen ingevuld staan. Ook wordt uitgezocht of er in subtabellen gezocht moet worden en welke velden daarin ingevuld zijn. Als de query is opgesteld moet deze getest en dan uitgevoerd worden, en de resultaten op het scherm weergegeven worden.

Nu als we zoeken op persoon nemen, geeft dat de volgende code:

```
persoon {
  set query [query_persoonzoek [list count "" "" ""] $naam $voornaam $geboortedatum
  $geboortedatum $voorletters $titel $gbacode $straat $huisnummer $postcode $email $telefoon]
  set count 1
  set sel [ns_db 0or1row $db "$query"]
  set count [getfromset $sel aantal]
  if {$count > 1000} then {
    append result"TEVEEL RESULTATEN, GEEF MEER EISEN OP!"
  } else {
    set berekening [expr $bladzijde - 1]
    set limietbegin [expr $berekening * $aantalres]
    set query [query_persoonzoek [list sorteer $sorteer "$limietbegin" "$aantalres"] $naam
```

```

$voornaam $geboortedatum $geboortedatum $voorletters $titel $gbacode $straat $huisnummer
$postcode $email $telefoon]
  set sel [ns_db select $db "$query"]
  if {$sel != ""} then {
    append result
"<TABLE><TR><TD>nummer</TD><TD>naam</TD><TD>voornaam</TD><TD>geboortedatum
</TD></TR>"
    while {[ns_db getrow $db $sel]} {
      set persnr ""
      set persnaam ""
      set persvoornaam ""
      set persdatum ""
      set persnr [ns_set value $sel 0]
      set persnaam [ns_set value $sel 1]
      set persvoornaam [ns_set value $sel 2]
      set persdatum [ns_set value $sel 3]
      append result
"<TR><TD>$persnr</TD><TD>$persnaam</TD><TD>$persvoornaam</TD><TD>$persdatum</T
D></TR>"
    }
    append result "</TABLE><HR noshade>"
  }
}
}
}

```

Nu is er alleen tussentijds een query opgesteld zodat deze naar de database opgestuurd kan worden. Deze code ziet er als volgt uit:

```

proc query_persoonzoek {zoekmode naam voornaam vanafgeb_datum totgeb_datum voorletters titel
gbacode straat huisnummer postc
  set mode [lindex $zoekmode 0]
  if {$mode == "volledig"} then {
    set select "SELECT DISTINCT nummer, naam, voornaam, geboortedatum, geboortedatum,
voorletters, titel, gbacode"
    set from " FROM persoon"
    set einde ";"
  } elseif {$mode == "count"} then {
    set select "SELECT COUNT (persoon.nummer) as aantal"
    set from " FROM persoon"
    set einde ";"
  } elseif {$mode == "sorteer"} then {
    set sorteer [lindex $zoekmode 1]
    set limietbegin [lindex $zoekmode 2]
    set limietaantal [lindex $zoekmode 3]
    set select "SELECT DISTINCT persoon.nummer, persoon.naam, persoon.voornaam,
persoon.geboortedatum"
    set from " FROM persoon"
    set einde " ORDER BY $sorteer LIMIT $limietaantal OFFSET $limietbegin;"
  } else {
    set select "SELECT DISTINCT nummer"
    set from " FROM persoon"
    set einde ""
  }
}

```

```

}
set where ""
if {$snaam != "" || $voornaam != "" || $vanafgeb_datum != "" || $totgeb_datum != "" || $voorletters !=
"" || $titel != "" |
set where " WHERE "
set previous 0
if {$snaam != ""} then {
set naam [searchtext $snaam]
append where "UPPER(naam) LIKE '{$naam}'"
set previous 1
}
...
if {$semail != ""} then {
set where [previouswhere $previous $where]
set previous 1
append from ", email_adres_van"
append where "persoon.nummer = email_adres_van.object1 AND email_adres_van.email_adres
IN ("
append where [query_zoekemail "onvolledig" $semail "" ""]
append where ")"
}
...
}
append select $from
append select $where
append select $einde
return $select
}

```

De resulterende query ziet er dan als volgt uit:

```

SELECT DISTINCT persoon.nummer, persoon.naam, persoon.voornaam, persoon.geboortedatum
FROM persoon, email_adres_van WHERE UPPER(voornaam) LIKE '%ROL%' AND
persoon.nummer = email_adres_van.object1 AND email_adres_van.email_adres IN (SELECT
email_adres_van.email_adres FROM email_adres_van WHERE UPPER(email_adres) LIKE
'%STO%') ORDER BY nummer LIMIT 50 OFFSET 0;

```

Zoals u kunt lezen in de code van query_persoonzoek, wordt een vraag naar het e-mailadres van de persoon gewoon doorgegeven naar query_zoekmail. Deze komt op dezelfde manier als query_persoonzoek met een query terug die het eraan voldoende emailadres als oplossing heeft. Dit resultaat wordt als geneste query toegevoegd aan de query van persoonzoek.

Voor iedere tabel is er een query zoekfunctie aangemaakt, en deze roepen elkaar aan. Hierdoor wordt de totale hoeveelheid code drastisch beperkt.

Een extra kracht van de zoekfunctie is dat deze niet beperkt is tot gebruik van alleen de zoekpagina. Ook elders in de software kan deze zoekmachine gevraagd worden om iets op te zoeken. Een voorbeeld hiervan is het adresboek. Op de eerste bladzijde is er een overzichtslijst van bekende contacten gesorteerd op de eerste letter (die bovenaan aangeklikt kan worden om de letter te specificeren). De code voor deze bladzijde is vrij klein omdat deze gebruikt maakt van de zoekmachine (zoek alle object1 beginnend met de letter 'a', object1 gebruikt geneste queries van persoon, afdeling en bedrijf).

18.3 Zwaktes van de zoekmachine

In de zoekmachine zitten op dit moment nog twee zwaktes waarvoor nog geen kant en klare oplossing voorhanden is.

a) Geneste queries zijn door de database zelf minder te optimaliseren dan “join” oplossingen. Er is toch bewust voor geneste queries gekozen. Redenering is dat “join” van twee tabellen in de zoekmachine heel veel extra code zou vergen omdat het doorgeven van vragen aan een andere query niet meer mogelijk zou zijn. Dit brengt veel extra werk met zich mee en maakt de risico's op bugs veel groter. Ook is mijn inschatting dat gemiddeld genomen in deze oplossing minder code uitgevoerd zou moeten worden dan bij de “join” oplossing. Dit zou weer een stijging geven van de reactietijd, in het voordeel van deze oplossing.

b) Het kan juist handig zijn dat alles uitgezocht word in één zoekveld (bijvoorbeeld naam, voornaam, bijnaam en e-mailadres als men zoekt naar een persoon). Dit is momenteel niet verwerkt, maar kan wel degelijk later ingebouwd worden.

19 Interface Design

Voor iedere ontwikkelaar geldt dat deze een interface moet ontwikkelen die aangepast is aan z'n gebruikers. Dit om het werken met z'n software zo aangenaam mogelijk te maken. Uiteindelijk is het de interface die bepaalt of gebruikers werkelijk het product gaan gebruiken of niet.

Voor de ontwikkeling van de interface is gesteund op de theorie van interface design[23]. Alhoewel dit niet letterlijk is overgenomen, is het wel continu een leidraad geweest bij de ontwikkeling.

19.1 De acht gouden regels voor interface design

De basis van interface design zijn de zogenaamde acht gouden regels. Deze principes zijn gebaseerd op ervaring en zijn continu bijgeschaafd.

1. Streef naar consistentie
Gebruik consequent dezelfde tekst, dezelfde invoervelden, etc.
2. Creëer universele bruikbaarheid
Inventariseer de verschillende gebruikers en hun verschillende wensen. Elk van deze beïnvloeden het ontwerp van één en dezelfde interface.
3. Geef informatieve terugkoppeling
Voor elke gebruikersinteractie zou er een informatieve feedback moeten zijn.
4. Ontwerp dialogen zonder open einde
Een actie zou een begin, midden en een einde moeten bevatten zodat het de gebruiker duidelijk is waar zijn actie begint en wanneer deze ten einde is.
5. Voorkom fouten
Voor zover mogelijk ontwerp een interface die fouten van de gebruiker voorkomt.
6. Maak een eenvoudige functie voor het ongedaan maken van acties
Zoveel mogelijk zouden acties ongedaan gemaakt moeten kunnen worden.
7. Ondersteun het interne gevoel van controle
Schep de vertrouwen dat de interface doet wat men ervan verwacht in die situatie.
8. Verminder kortetermijngeheugenbelasting van de gebruiker
De limieten van de hoeveelheid dingen die een persoon in het korte-termijngeheugen kan onthouden, maken dat interface simpel gehouden moet worden, en niet moet verwachten dat de gebruiker van het scherm teveel onthoudt.

Alhoewel ik graag zou hebben dat alle acht regels perfect geïmplementeerd zijn, is dat niet helemaal het geval. Met name 3, 6 en 7 zouden verbeterd moeten worden doorheen de interface.

19.2 Het organiseren van de weergave

Bij een project als dit slaan we heel veel data op, en die correct worden weergegeven. Er zijn vijf doelstellingen voor dataweergave:

1. Consistentie van de dataweergave
2. Efficiënte informatieopname door de gebruiker
3. Minimale geheugenbelasting voor de gebruiker
4. Overeenkomst van de dataweergave met de data
5. Flexibiliteit voor de gebruiker van de dataweergave

Met name het laatste punt is slecht uitgevoerd, omdat er momenteel niet de indruk is dat met de beperkte hoeveelheid knoppen en functies op een bladzijde (die er nu zijn) deze flexibiliteit al echt nodig is.

19.3 De aandacht van de gebruiker dirigeren

De gebruiker heeft een gelimiteerde hoeveelheid aandacht. Door de volgende technieken kan deze gedirigeerd worden naar die plaatsen waar de informatie staat die de gebruiker direct nodig heeft.

1. Intensiteit
2. Markeringen
3. Grootte
4. Keuze van fonts
5. Inverse kleuren
6. Knipperen
7. Kleur
8. Geluid

Niet al deze technieken zijn letterlijk overgenomen. Uit de theorie van datacommunicatie komt dat de keuze van communicatiemethoden belangrijk is wat betreft de rust die een interface geeft naar de gebruiker. Alhoewel technieken als knipperen, inverse kleuren of geluid ideaal zijn om de aandacht van de gebruiker te trekken, kunnen deze ook als hinderlijk ervaren worden.

Hierom is bewust gekozen voor rust door het gebruik van kleuren waarop de receptoren van onze ogen niet zo wild reageren.

19.4 Faciliteren van data-invoer

Data-invoer kan een behoorlijk deel van de gebruikers tijd in beslag innemen. Hiervoor zijn vijf basisregels voor opgesteld.

1. Consistentie van data-invoer transacties
2. Minimale invoer acties van de gebruiker
3. Minimale geheugenbelasting voor de gebruiker
4. Overeenkomst van de data met de dataweergave
5. Flexibiliteit voor de gebruiker van de data-invoer

19.5 Identificatie van gebruikers, taken en interactiestijl

Gebruikers

Expert frequente gebruikers

Deze gebruikers zijn door en door bekend met de werking van de interface, en zoeken vooral hun werk zo snel mogelijk uit te voeren.

Acties

Frequente acties

Acties die door de gebruiker zeer vaak uitgevoerd worden, en nogal repetitief van aard zijn. Hieronder valt met name het ingest systeem voor de archivaris, wat een menu is waar sommige archivariissen de hele dag mee werken.

Minder frequente acties

De meeste acties komen minder frequent voor. De gebruiker komt die in het dagelijks gebruik wel tegen, maar is er niet continu mee bezig.

Dit is het grootste gedeelte van het project.

Interactiestijl

Menuselectie

Voor het snel navigeren tussen de totaal verschillende delen van de software is links een menu opgesteld.

Formulieren invullen

Als data-invoer vereist is, is menuselectie nogal onhandig. Daarvoor is het gebruik van formulieren een simpele methode.

In de meer complexe menu's en ter voorkoming van fouten zijn meer interactieve menu's zoals pulldownmenu's behulpzaam.

Verder zijn er in de ontwikkeling nog een aantal andere principes gebruikt

a) Het aanmaken van een "beheerder" die de structuur van gebruik vastlegt, alsook een "archivaris" die de binnenkomende documenten archiveert. Bij kleinere organisaties is dit dezelfde persoon.

De beheerder is iemand met kennis van de organisatie als geheel. Hij deelt de afdelingen in, de groepen, de "opdrachten" (soorten dossiers), etc.

De archivaris heeft een simpelere taak. Hij scant en verwerkt de documenten opdat niet iedereen in het bedrijf z'n eigen post hoeft te verwerken (vermindering van taken binnen de organisatie).

Ook verwerkt hij de uitgaande post (printen, op de post doen, etc.)

b) De vloeiende lijn. Een pagina binnen het softwarepakket is nooit het eindpunt. Op alle vragen die zich men binnen die bladzijde zou kunnen stellen, moet men een antwoord kunnen vinden door op de bijbehorende link te klikken. Als men bijvoorbeeld naar een brief kijkt, en men ziet de verzender, moet men niet via het menu gaan zoeken naar deze persoon en zijn gegevens, maar moeten die gegevens beschikbaar zijn door op de link te klikken.

c) Vlotheid. Als een gebruiker moet wachten is het een slechte interface. Simpel gesteld, een interface die dingen direct kan uitvoeren is een voordeel. Gebruikers zijn niet tevreden van een interface waarop men na iedere klik een paar tellen moet wachten.

Wat dat betreft is een webinterface een risico. In de praktijk blijkt dit erg mee te vallen. Op een lokaal netwerk is de verwerking onmiddellijk. Erger nog, soms moest ikzelf zelfs nakijken of een verwerking wel gebeurd was, omdat ik deze niet zo snel verwachtte. Alles bij elkaar bestaat het “laden” van een bladzijde niet.

Waarom niet? Door de snelheid van het netwerk, maar ook zijn de pagina's klein (zoveel mogelijk tekst, weinig plaatjes). De grootste vertraging die momenteel opgemerkt is, is de hoeveelheid code die bepaalde individuele pagina's moeten uitvoeren.

20 Performance

Op dit moment reageert de software snel, maar zijn er nog niet teveel data of documenten in opgeslagen. Hoe gaat dit zich verhouden als er meer documenten en gegevens worden opgeslagen?

20.1 Aantal documenten

Hoe wordt de performance beïnvloed door het aantal opgeslagen documenten (bestanden) op de schijf?

De software doet maar op een paar plaatsen een access naar fysieke files op de harde schijf:

Archivering door de archivaris

Als de archivaris archiveert, wordt er de volgende code uitgevoerd:

```
[glob -nocomplain $locatie/$document/*]
```

Deze code leest alle bestandsnamen in op de gegeven locatie.

Daarnaast zijn er nog een aantal acties op individuele bestanden (een rename, of een controle van de access-time), maar eigenlijk is die glob de enige actie die beïnvloedbaar is door aantallen bestanden.

Indien in de locatie steeds meer documenten staan, zal dit de snelheid negatief beïnvloeden.

Als daar echter veel bestanden in staan, wil dat zeggen dat de archivaris achterloopt. Als deze z'n achterstand beperkt houdt, dan zal dat ook geen invloed hebben in de snelheid.

Archivering door de werknemer

Eigenlijk hetzelfde als bij de archivaris, met die opmerking dat elke keer als deze een bestand wil archiveren, alle bestanden gecontroleerd worden of deze wel door deze persoon gearchiveerd mogen worden. Oftewel, vertraging treedt ook op als de collega's achterlopen met hun werk.

Opvragen van een dossier

Hierin wordt de volgende code uitgevoerd:

```
[glob -nocomplain /archieff/$dosdir/*]
```

Alle documenten op de schijf worden nagelopen om te zien of er naast de door de database aangegeven bestanden ook gelinkte bestanden bestaan (deze worden alleen getoond). Theoretisch, als er heel veel bestanden in een dossier staan (lees duizenden), zou dit tot vertragingen kunnen leiden. Een goede indeling voorkomt dit.

Openen van een bestand

Uiteindelijk kan men een bestand openen, maar dit is er uitsluitend één tegelijk. Bestandssystemen zijn tegenwoordig snel in het vinden van bestanden, dus ook dat zal geen invloed hebben.

Samenvatting

De vertragingen bij de werknemers kunnen in een later stadium opgelost worden. Dit door de individuele documenten door te geven in een "todo" in de agenda. Hierdoor hoeft de software niet meer alle documenten af te lopen.

Als de archivariissen het archiveren bijhouden, en dossiers niet oneindig groot kunnen worden, wordt de software niet beïnvloed door het aantal documenten.

Algemeen kan men stellen dat de werkelijke beperking in het aantal documenten de grootte van de schijf zal zijn.

Als de bestanden 100KB gemiddeld zijn, kan men op 100TB zo'n miljard documenten kwijt.

20.2 Hoeveelheid data

Wordt de software beïnvloed door de hoeveelheid data opgeslagen in de database?

Deze vraag is tweeledig.

Hoe wordt de database beïnvloed door de hoeveelheid data

De gekozen database is PostgreSQL. Maar daarnaast ondersteunt AOLserver een aantal andere databases zoals IBM DB2 en Oracle. Die staan er om bekend voldoende data aan te kunnen.

Dat wil niet zeggen dat met PostgreSQL men limieten gaat tegenkomen[14,15].

Samengevat komt het erop neer dat in een database van 1TB ongeveer de gegevens van 5 miljard documenten opgeslagen kunnen worden. Dat is echter meer dan enkele andere limieten (zoals opslag van een dergelijke hoeveelheid documenten op een schijf)

Hoe wordt de software beïnvloed door de hoeveelheid data in de database

Er is geen enkele bladzijde die een oneindig hoeveelheid gegevens uitleest. Dus zodra bekende maxima bereikt worden, wordt de software niet verder negatief beïnvloed.

20.3 Aantal gebruikers

Wat is de invloed van het aantal gebruikers op de performance?

Ook deze vraag is tweeledig.

Hoeveel hits per seconde kan de webserver aan

In 1999, met de toen actuele software en hardware, 28.000 hits/seconde[20]. Aangenomen mag worden dat met de huidige release van AOLserver en huidige hardware de performance beter zal zijn.

Hoeveel bandbreedte verbruikt de software per gebruiker

Het komt maar zelden voor dat bladzijden boven de 6KB aan dataverkeer nodig hebben. Een gigabit-verbinding zou dan per seconde ruim 16.000 bladzijden kunnen leveren.

De uitzondering op het dataverkeer per pagina is het bekijken van opgeslagen documenten. Dit is afhankelijk van de grootte van de opgeslagen documenten.

De software is wel zo gemaakt dat deze niet continu het document opvraagt. Alleen als de gebruiker het document echt opgevraagd wordt, zal deze data verstuurd worden.

Als we ervan uitgaan dat een gemiddeld document 100KB (twee gescande bladzijden) is, kan een gigabit-verbinding 1.000 documenten per seconde versturen.

20.4 Conclusie

Theoretisch gezien, zou men uit het bovenstaande kunnen concluderen dat de software pas zal vertragen als de database, de webserver of het filesysteem vertragen.

Daaruit kunnen de volgende conclusies getrokken worden:

De theoretische limiet van de huidige implementatie zou ongeveer 1 miljard documenten zijn, en zo'n 15.000 hits/seconde.

Dat zou echter een te snelle conclusie zijn. Dat zijn de limieten van de gebruikte software (AOLserver en PostgreSQL) en hardware.

Mijn persoonlijke verwachting is echter dat mijn eigen code de limieten zal creëren. De verwachting is dat de hoeveelheid uit te voeren code op sommige plaatsen, en complexe queries, veel eerder voor vertragingen zullen gaan zorgen.

Door de software te tweaken (verbeteren), zullen de limieten steeds verder opgerekt worden, maar dan nog zullen de hierboven geschetste optimistische cijfers nooit gehaald worden. In de praktijk is er door mij al vaker vastgesteld dat als de limieten opgezocht worden van wat kan, de realiteit wat weerbarstiger blijkt dan de theoretisch opgegeven cijfers.

21 Evaluatie door gebruikers

De evaluatie is opgedeeld per soort pagina.

Scannen: Omdat de gebruiker zich weinig druk hoeft te maken over instellingen voor het archief, werkt dit zeer vlot, zeker bij gebruik van een sheetfeeder

Ingest fase 1: Invoer door de archivaris gaat vlot omdat deze relatief weinig velden hoeft in te vullen, en de pulldown menu's maken dit werk makkelijk.

Bij het aangeven van vorig_document en de verzender (ontvanger wordt aangenomen als het eigen bedrijf), wordt als iets minder vlot ervaren vanwege het gebruik van een tweede scherm. Initieel werden de gegevens van de invoervelden van de ingest niet goed doorgegeven naar het tweede scherm, maar dat is inmiddels opgelost.

Een hardnekkiger probleem is dat soms het document blijft terugkomen als het eigenlijk overgeslagen moet worden. Voor de snelheid van verwerking is de ingest zo opgebouwd dat de documenten niet in volgorde afgehandeld hoeven te worden, maar men een document kan overslaan. Dit blijkt niet altijd correct te werken, al is nog niet duidelijk waarom.

Ingest fase 2: De naam van een nieuw dossier mag niet “Nieuw Dossier” zijn, vanwege verwarring met de keuzemogelijkheid die een nieuw dossier aanmaakt.

Handig is dat meerdere bestanden als bladzijden van een document geïnterpreteerd kunnen worden. Zo kunnen meerdere TIFF-bestanden als één document gezien worden (en kan de volgorde opgegeven worden), maar kan ook een word-bestand kan gecombineerd worden met een powerpoint bestand. Het hangt ervan af wat er als een “document” gezien wordt.

Verder vond men het vlot werken.

Overzichtsschermen (opdracht, dossier en document): Deze werken zeer intuïtief. Ze zijn simpel en snel in gebruik.

Opmerkingen zijn het ontbreken van een “refine” invoerveld (een methode om het aantal oplossingen snel terug te brengen en meer in de richting van de gebruiker. Mogelijk iets met een beginletter zoals in het adresboek).

Ook het sorteren op onderwerp/titel ontbreekt nog.

Schermen beheer van opdracht, dossier en document: Voor de beheerder/archivaris begint “information overload” een probleem te worden. Er zijn dusdanig veel mogelijkheden dat de gebruiker op een gegeven moment door de bomen het bos niet meer ziet. Maar dit is alleen bij beheerders, andere gebruikers hebben geen probleem.

De werking is wel intuïtief en overzichtelijk.

Algemene opmerkingen: De software werkt sneller dan papieren dossiers.

Breed toepasbaar vanwege meertaligheid en het aankunnen van allerlei soorten dossiers.

Sjablonen zijn een zeer goede aanvulling aan dit soort software. Ze besparen de gebruiker zeer veel tijd als er een brief opgesteld moet worden, en deze brief is dan altijd in de bedrijfsstijl.

De agenda ontbreekt nog gedeeltelijk. Het ontbreken van todo's, die eigenlijk nodig zijn voor het archiveren van documenten zijn een gemis. Documenten die binnenkomen zouden direct todo's moeten genereren bij de gebruikers die ermee aan de slag moeten. Hierdoor worden ze automatisch op de hoogte gesteld, inclusief een tijdslijmet, van binnengekomen documenten.

Workflow is op zijn beurt ook zwaar afhankelijk van de agenda en todo's. Deze kan een nieuwe todo genereren op basis van een binnengekomen document. Als laatste ontbreekt ook nog een ontvangstbevestiging.

22 Conclusie

De probleemstelling van “Van Lee & van Everdingen” en van de gemeente Den Briel, was om een archiveringspakket te bouwen voor gebruik in een organisatie, dat alle soorten documenten in allerlei dossiers kan plaatsen, en dat binnen allerlei soorten opdrachten.

Als eerste is er research gedaan bij de klant, om uit te vinden hoe deze werken. Als ervaring heb ik in het verleden bij een aantal diverse klanten gewerkt, en daarvan de werkwijzen binnen de context geplaatst. De verschillende werkwijzen van al deze bedrijven zijn door mij geanalyseerd. Als basis voor de ontwikkeling van de software is gebruik gemaakt van de overeenkomsten in de omgang met documenten binnen deze organisaties.

Vervolgens ben ik in de theorie gedoken. Deze geeft een leidraad in een project als dit. Met name het model OAIS toont zeer scherp welke onderdelen geïmplementeerd moeten worden in een archief.

Met de theorie, ervaring uit vorige projecten en heel wat college-uren als achtergrondkennis, is er door mij vervolgens in dit project heel wat tijd gestopt in het analyseren hoe zo'n softwarepakket eruit zou moeten zien. Vervolgens de implementatie.

Het eindresultaat is een behoorlijk uitgebreid pakket, met tal van functies die het werken met documenten vlotter maken. Nog voor een document de organisatie ingaat is het gedigitaliseerd, en kan er met de digitale variant gewerkt worden. Niet alleen wordt het document digitaal doorgegeven, ook wordt er een omgeving gecreëerd die de gebruiker helpt bij het werken met dossiers en documenten. Het geheel is opgebouwd volgens de eisen rond het document structuur plan, dat wil zeggen dat de structuur van een document structuur plan geïmplementeerd kan worden in deze software.

Het beheer van dossiers is in handen van teams, en niet van één afdeling. Dit omdat een opdracht niet altijd echt aan een afdeling toe te kennen is, en soms meerdere afdelingen betrokken zijn bij een opdracht.

Het hele team kan in de opdracht documenten toevoegen, of gegevens gebruiken om nieuwe documenten op te stellen. Om dit te versnellen kunnen sjablonen direct met deze gegevens gevuld worden.

De gebruikers ervaren de interface als prettig, rustig maar tegelijk volledig en zeer snel. Dit ondanks het gebruik van standaard-HTML, zonder gebruik van Flash, Java of JavaScript.

Voor het zoeken van gegevens in de database is er een uitgebreide zoekmachine gebouwd. Archivarissen zijn vaak veel tijd kwijt aan het terugzoeken van documenten. Dit zou nu door de gebruiker zelf gedaan kunnen worden, waardoor de archivaris meer tijd over houdt voor andere zaken of voor de echt onvindbare documenten.

Het is mogelijk om mensen van buiten de eigen organisatie toegang te geven tot bepaalde documenten, zonder de gehele database of alle documenten open te stellen. Dit kan door aan te geven welke functie (of taak) ze in een bepaald dossier uitoefenen.

Dit product voldoet dus al aan de probleemstelling. Dit soort producten zijn ook met name geschikt om tal van taken binnen de organisatie te automatiseren. Zulke uitbreidingen zijn nog mogelijk, en volgen mij voor veel organisaties wenselijk.

Met name de verbinding tussen de agenda en het archief geeft nog tal van mogelijkheden voor toekomstige uitbreidingen.

23 Further development

23.1 Conversie

Probleem: Bij een vorig project converteerden we MS-word documenten naar PDF voor de mensen die de website gebruikten.

Alhoewel dit wel aardig werkte was het niet ideaal.

Word is daar ondertussen verouderd, maar omdat alles hardcoded staat, is het moeilijk te vervangen.

Een systeem als dit zou flexibel moeten zijn t.o.v. alle type documenten die opgeslagen zijn.

Conversie is vaak ook maar één regel code die uiteindelijk uitgevoerd moet worden.

Oplossing? In de database het regeltje code opslaan dat moet uitgevoerd worden voor de conversie. Dan kan er naar de extensie/formaat van het brondocument gekeken worden, bepaald worden naar welk formaat dit geconverteerd moet worden, en dit aan de gebruiker voorgelegd worden.

Het Probleem dat blijft is hoe controleer ik of deze code nog werkt? Na verloop van tijd (bijvoorbeeld na een upgrade) werkt code niet meer, dus moet deze aangepast worden. Een beheerder zal echter nooit alle conversies controleren. Hiervoor zal nog een oplossing gezocht moeten worden.

Benodigde data:

- Formaat
- Extensie
- Code

23.2 Backup

Met name vanuit de overheid is de vraag om documenten te archiveren samen met hun metadata in een XML-formaat.

De documenten zouden naar een XML-formaat geconverteerd moeten worden, en dan samengevoegd worden met hun metadata. De resulterende set bestanden zouden dan gecomprimeerd kunnen worden en op een backupmedium geplaatst kunnen worden als archief.

23.3 Sjablonen

Wegens tijdgebrek nog niet uitgevoerd maar een bijzonder belangrijk onderdeel van het project als geheel.

Sjablonen die gebruik maken van data uit de database vormen juist die functie die gebruikers in staat stelt om tijd te besparen.

Dit onderdeel is nogal complex. Om gebruikers maximaal te helpen moeten zoveel mogelijk gegevens gebruikt worden. Dit met een minimum aan input van diezelfde gebruikers (anders kunnen die het net zo goed zelf opzoeken). Daarnaast is er de wens om adressen semantisch correct te plaatsen (hiermee wordt bedoeld dat het adres klopt, ongeacht welk land bedoeld wordt), zonder dit per land in de code uit te pluizen.

23.4 Conversie van documenten

Documenten zullen vaak worden opgeslagen in de formaten die mensen dagdagelijks gebruiken. Dit zal niet altijd het formaat zijn dat mensen binnen en met name buiten de organisatie willen zien. Hiervoor zal enige vorm van conversie gebouwd moeten worden, zodat niet elke keer manueel geconverteerd hoeft te worden. De software moet de tools zelf beschikbaar hebben om zowel in het access gedeelte als eventueel tijdens ingest te kunnen converteren zonder ingrijpen van een beheerder.

23.5 Postcodes zoeken

De betere adresboeksystemen voor Nederland hebben de mogelijkheid om een stad of straat te zoeken door het ingeven van de postcode.

Nu is het een beetje naïef om te denken dat we alleen in Nederland adressen hebben, dus dit zal uitgebreid moeten worden.

Voor de meeste landen definieert de postcode een plaats. Voor een paar landen (zoals Nederland en het Verenigd Koninkrijk) zelfs de straat of een deel daarvan.

De databasestructuur is al aangepast om een dergelijk systeem te bevatten. Beter nog, omdat het onmogelijk is om alle postcodes van de wereld te gaan invoeren, is het systeem zelflerend gemaakt. Binnen een land, als de postcode valt binnen twee andere postcodes van dezelfde plaats, is het redelijk aan te nemen dat het om dezelfde plaats gaat. Mocht dat niet het geval zijn, kunnen we in ieder geval de twee plaatsen geven met postcodes omheen deze postcode. Er is een redelijke kans dat één van deze twee correct is. Zo zijn 2600AA tot 2629ZZ de postcodes van Delft.

Naarmate er meer adressen ingevoerd worden, wordt het systeem vanzelf correcter.

24 Literatuurlijst

Gepubliceerde artikelen

- [1] Ahmed E. Hassan en Richard C. Holt. A Reference Architecture for Web Servers. *Proceedings of the 2001 ACM Symposium on Document engineering*, volume 01 2001, P84 – 94
- [2] Ari Salminen en Frank Wm. Tompa. Requirements for XML Document Database Systems. *Proceedings of the 2001 ACM Symposium on Document engineering*, Sessie Document Databases P85 – 94
- [3] Kyong-Ho Lee, Oliver Slattery, Richang Lu, Xiao Tang en Victor McCrary. The state of the art and practice in digital preservation. *Journal of Research of the National Institute of Standards and Technology*, Januari-februari 2002, P93 – 106
- [4] Martin Waldron. Developing an information management strategy. *Business Information Review*, 2008; 25 P101 – 104
- [5] Michael Tiemann. An objective definition of open standards. *Computer standards and interfaces*, 28 P495 – 507
- [6] Reuven M. Lerner. At the forge: introducing AOLserver. *Linux Journal*, 2002 volume 101, P12
- [7] Rolf Herzog. PostgreSQL--The Linux of Databases. *Linux Journal*, 1998 Volume 46, P1

Artikelen

- [8] Alfresco Software Inc. Alfresco Document Management.
<http://www.alfresco.com/resources/datasheets/Alfresco-Document-Management.pdf> (17/9/2009)
- [9] ArchiefBC. Een Documentair Structuurplan.
<http://www.archiefbc.nl/documenten/handleiding%20dspmaart%202003.pdf>
(17/9/2009)
- [10] Cornwell. Model Requirements for the management of electronic records.
<http://www.cornwell.co.uk/edrm/moreq.asp> (17/9/2009)
- [11] Data Capture Solutions LTD. ISO 15489 – The essentials.
<http://www.datacapture.co.uk/pdf/ISO-15489-White-Paper.pdf?=&SID> (17/9/2009)
- [12] Digital Preservation Testbed. E-mail-XML Demonstrator: Technical description.
<http://www.digitaleduurzaamheid.nl/bibliotheek/docs/email-demo-en.pdf>
(17/9/2009)
- [13] Donald Sawyer, Lou Reich, David Giaretta, Patrick Mazal, Claude Huc, Michel Nonon-Latapie en Nestor Peccia. The Open Archival Information System (OAIS) Reference Model and its Usage.
<http://www.iaaa.org/Spaceops2002Archive/papers/SpaceOps02-P-T5-39.pdf>
(17/9/2009)
- [14] Greg Smith, Robert Treat, and Christopher Browne. Tuning Your PostgreSQL Server.
http://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server (17/9/2009)
- [15] Jignesh Shah. Problems with PostgreSQL on multi-core systems with multi-Terabyte data.
http://blogs.sun.com/jkshah/resource/pgcon_problems.pdf (17/9/2009)
- [16] ISO. Information and documentation - Records management.
http://www.iso.org/iso/catalogue_detail?csnumber=31908
- [17] ISO. Information technology - Open Document Format for Office Applications (OpenDocument) v1.0. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485
(17/9/2009)
- [18] National Aeronautics and Space Administration. Reference model for an open archival information system.
<http://public.ccsds.org/publications/archive/650x0b1.pdf> (17/9/2009)
- [19] Nuxeo. Brochure Nuxeo Platform.
http://www.nuxeo.com/publications/brochures/nuxeo-platform-en/downloadFile/file/brochure_platform_en.pdf (17/9/2009)
- [20] Philip Greenspun. Introduction to AOLserver.
<http://philip.greenspun.com/wtr/aolserver/introduction-1.html> (17/9/2009)
- [21] Serco Consulting. Model Requirements for the management of electronic records update and extension.
http://ec.europa.eu/transparency/archival_policy/moreq/spec_moreq2_en.htm (17/9/2009)

[22] Xerox. Xerox DocuShare Share, Search, Automate, Save.
<http://docushare.xerox.com/global/pdfs/DS65.pdf> (17/9/2009)

Boeken

[23] Ben Shneiderman & Catherine Plaisant. Designing the User Interface. Pearson Education, Inc. 2005 (ISBN 0-321-26978-0)

[24] Gerald Carter. LDAP System Administration. O'Reilly Media, Inc 2003 (ISBN 978-1-56592-491-8)

[25] Korry Douglas & Susan Douglas. PostgreSQL - A comprehensive guide to building, programming and administering PostgreSQL databases. Sams Publishing 2003 (ISBN 0-7357-1257-3)

Websites

[26] <http://www.alfresco.com/> (17/9/2009)

[27] <http://dev.aolserver.com/wiki/Modules> (17/9/2009)

[28] <http://forums.devshed.com/postgresql-help-21/postgresql-vs-mysql-view-performance-629512.html> (17/9/2009)

[29] <http://www.digitaleduurzaamheid.nl/home.cfm> (17/9/2009)

[30] <http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html> (17/9/2009)

[31] <http://www.noiv.nl/> (17/9/2009)

[32] <http://www.nuxeo.com/en/> (17/9/2009)

[33] <http://www.oasis-open.org/home/index.php> (17/9/2009)

[34] <http://www.opendocman.com/> (17/9/2009)

[35] <http://www.postgresql.org/about/> (17/9/2009)

[36] <http://dlimforum.typepad.com/> (17/9/2009)

[37] http://en.wikipedia.org/wiki/Records_management (17/9/2009)

[38] http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL (17/9/2009)

[39] <http://docushare.xerox.com/> (17/9/2009)

[40] <http://www.xtandit.nl/> (17/9/2009)

