

Metropolia Ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Mikko Laine

Dokumentinkäsittelyprosessin hallintatyökalu

Insinööritö 20.4.2009

Ohjaaja: jaosjohtaja Jani Peltopuro

Ohjaava opettaja: yliopettaja Harri Airaksinen

Metropolia Ammattikorkeakoulu Insinööri­työn tiivistelmä

Tekijä Otsikko	Mikko Laine Dokumentinkäsittelyprosessin hallintatyökalu
Sivumäärä Aika	66 sivua 20.4.2009
Koulutusohjelma	tietotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	jaosjohtaja Jani Peltopuro yliopettaja Harri Airaksinen
<p>Tässä insinöörityössä kehitettiin dokumentinkäsittelyprosessin hallintatyökalu Pöyry Industry Oy:n laitosuunnitteluosastolle. Työkalun nimeksi muodostui Flow, joka kuvastaa työkalun käyttötarkoitusta moniportaisen, virtauksen lailla etenevän työprosessin työvälineenä. Tavoitteena oli tehdä täysin itsenäinen sovellus, joka ei vaikuta muiden järjestelmien nykyiseen toimintaan. Työkalu mahdollistaa dokumentin siirtämisen työvaiheesta toiseen ja kommentoinnin täysin digitaalisesti, mikä on välttämätön edistysaskel ajan säästämiseksi ja tehokkuuden lisäämiseksi globalisoituvassa maailmassa sekä kiristyvässä kilpailussa.</p> <p>Flow-sovellusta ajetaan paikallisena sovelluksena. Se on luotu Microsoft Visual Studio 2008 -kehitysympäristössä C#-ohjelmointikielellä. Flow hakee tarvitsemansa tiedot olemassa olevista eri tietojärjestelmistä. Käyttäjätiedot saadaan Microsoftin Active Directorystä LDAP-verkkoprotokollan avulla ja suunnitteluprojektien sekä -dokumenttien tiedot IBM Lotus Notes -tietokannan päälle rakennetusta Pöyryn omasta dokumentinhallintajärjestelmä JPDocista sen URL-pohjaisista kyselyistä palautuvia XML- ja HTML-dokumentteja jäsentämällä.</p> <p>Työkalu tallentaa kaiken tarvitsemansa tiedon, jota ei käytetä nykyisissä järjestelmissä, kehityspalvelimella sijaitsevaan SQL-relaatiotietokantaan. Poikkeuksena ovat kuitenkin tarkastusprosessin päätteeksi hylättyjen dokumenttien kommentoidut versiot, jotka palaavat vesiputousmallissa takaisin edelliselle portaalle. Ne tallennetaan Microsoft SharePoint -sivun dokumenttikirjastoon. PDF-muotoisten katselutiedostojen kommentointiin käytetään Adobe Acrobat -ohjelman piirtotyökaluja. Alkuperäiset työ- ja katselutiedostot siirretään JPDociin käyttämällä Document Explorerin COM-objektia myöhäisellä sidonnalla.</p> <p>Insinöörityön tuloksena oli Flow-ohjelman koeversio, jolla voidaan käytännössä tutkia työmenetelmien muutosten mahdollistamia tehokkuusparannuksia, eri osa-alueiden heikkouksia ja vahvuuksia riippumatta työorganisaation jäsenten sijainnista. Mikäli kokeilujakso osoittautuu onnistuneeksi, tullaan vastaavia toimintoja liittämään tulevaisuudessa kehitettäviin sisäisiin dokumentinhallintajärjestelmiin.</p>	
Hakusanat	dokumentinhallinta, C#.NET, SQL, XML, HTML, SharePoint, Lotus Notes, Active Directory, LDAP, COM, ROT

Author Title	Mikko Laine Control tool of the document handling process
Number of Pages Date	66 pages 20 April 2009
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Jani Peltopuro, Section Manager Harri Airaksinen, Principal Lecturer
<p>In this thesis a control tool of the document handling process was developed for the design technology department of Pöyry Industry Oy. The name of the tool is Flow, which describes the purpose of its usage as a tool for a multi-step creation process of workflow documents. The target was to create a fully independent program, which does not affect the existing systems and programs at all. The tool makes it possible to move documents from one stage to another and comment them totally digitally, which is a necessary step to be able to compete on top in the continuously globalizing world.</p> <p>Flow is run as a desktop program. It has been developed in the integrated development environment of Microsoft Visual Studio 2008 using C# programming language. Flow collects the necessary information from existing systems. User information is imported from Microsoft Active Directory by using LDAP protocol. The project and document information is imported by parsing XML and HTML documents that are responses from URL based queries from the document management system JPDoc that runs on an IBM Lotus Domino server.</p> <p>The tool stores all the required data that is not used in existing systems to an SQL relational database. Exceptions are the commented view files that have failed the checking process of documents. They are stored into a document library of a Microsoft SharePoint site. The view files that are in PDF format are commented by using the commenting tools of Adobe Acrobat. The original work files and view files are uploaded to JPDoc by using late binding of a COM object of running Document Explorer, which is the user interface of JPDoc.</p> <p>The aim of this thesis was to develop a pilot version of Flow, which makes it possible to study the gained advantages and weaknesses in practice, regardless of how large and decentralized the project organization is. If the pilot period proves to be successful, the features of Flow will be added to future versions of in-house developed document management systems.</p>	
Keywords	document management, C#.NET, SQL, XML, HTML, SharePoint, Lotus Notes, Active Directory, LDAP, COM, ROT

Sisällys

Tiivistelmä

Abstract

Tekniset käsitteet ja lyhenteet

1	Johdanto.....	7
2	Dokumentinkäsittely	9
2.1	Dokumentinkäsittelyprosessin periaate.....	9
2.2	Nykyinen dokumentinkäsittelyprosessi	10
2.3	Olemassa olevat tietojärjestelmät ja sovellukset.....	10
2.3.1	Dokumentinhallintajärjestelmä JPDoc.....	10
2.3.2	Document Explorer	12
2.3.3	Henkilötietojärjestelmät	13
3	Flow-sovelluksen kehittäminen ja käytetyt menetelmät	16
3.1	Tarvemäärittely ja työn rajaaminen.....	16
3.2	Kehitysympäristön ja tekniikoiden valinta.....	20
3.2.1	Microsoft Visual Studio ja .NET Framework	20
3.2.2	C#-ohjelmointikieli (C-Sharp)	21
3.2.3	Adoben PDF-tiedostomuoto	22
3.2.4	Relaatiotietokanta ja SQL-kieli.....	23
3.2.5	Microsoft SharePoint ja dokumenttikirjasto	24
3.3	Olemassa olevien tietojärjestelmien hyödyntäminen.....	25
3.3.1	Käyttäjätietojen hakeminen LDAP-verkkoprotokollan avulla.....	25
3.3.2	Tietojen hakeminen JPDocista rajapinnan avulla	26
3.3.3	Tietojen vienti JPDociin COM-objektia hyödyntämällä.....	28
3.4	Tietokannan suunnittelu ja toteutus	29
3.5	Tiedon tallentaminen.....	33
3.6	Käyttöliittymän ja ohjelmistorakenteen suunnittelu	35
4	Flow-sovelluksen ominaisuudet ja toiminta	36
4.1	Ohjelman ominaisuudet	36
4.2	Tilojen yhteiset toiminnot	37

4.3	Hallintatila.....	39
4.4	Käyttäjätila.....	42
5	Koevaihe.....	49
5.1	Koevaiheen kuvaus.....	49
5.2	Saavutetut hyödyt.....	50
5.3	Tulevaisuuden kehityskohteet.....	51
6	Yhteenveto.....	57
	Lähteet.....	58

Liitteet

Liite 1: Perustarkastusprosessin vuokaavio	61
Liite 2: Flow-sovelluksen aiheuttamat muutokset työprosessin kulkuun tutkitussa projektissa	62
Liite 3: Dokumentin metatiedon XML-listaus.....	64
Liite 4: Keskeisten dokumentti- ja projektiluokkien toteutetut metodit	65
Liite 5: Vuokaavio sovelluksen toiminnasta dokumenttitilavaihdossa.....	66

Tekniset käsitteet ja lyhenteet

C#	Microsoftin kehittämä olio-ohjelmointikieli.
CAD	<i>Computer-Aided Design</i> ; suunnittelijan digitaalinen työympäristö.
COM	<i>Component Object Model</i> ; ohjelmien välisen kommunikaation mahdollistava rajapintastandardi.
HTTP	<i>Hyper Text Transfer Protocol</i> ; Internet-selainten ja palvelimien käyttämä hypertekstin siirtoprotokolla.
JPDoc	Pöyryn dokumentinhallintajärjestelmä.
LDAP	<i>Lightweight Access Protocol</i> ; hakemistopalveluiden verkkoprotokolla.
PDF	<i>Portable Document Format</i> ; järjestelmäriippumaton, siirrettävä tiedostomuoto.
ROT	<i>Running Object Table</i> ; tietojen jakamiseen kykenevien, käynnissä olevien ohjelmaprosessien listaus.
SQL	<i>Structured Query Language</i> ; relaatiotietokannan hallintakieli.
XML	<i>eXtensible Markup Language</i> ; järjestelmäriippumaton tiedon merkintäkieli.

1 Johdanto

Tässä insinööriyössä suunnitellaan ja toteutetaan sovellus, jolla voidaan nykyaikaisesti hallinnoida dokumenttien elinkaarta suunnitteluprosessissa. Työ tehdään Pöyry Industry Oy:n laitossuunnitteluosastolle, joka on osa Pöyry Oyj:tä. Pöyry Oyj, josta käytetään yleisesti lyhyemmin nimeä Pöyry, on diplomi-insinööri Jaakko Pöyryn perustama konsultointiyritys, jonka toiminta alkoi Suomessa vuonna 1958. Sitten Pöyry on laajentanut toimintaansa 49:ään eri maahan, ja se on usean eri alan markkinajohtaja, joka työllistää noin 8 000 henkilöä. Yrityksen pääkonttori sijaitsee Vantaalla, jossa työskentelee noin 1 500 henkilöä.

Pöyryn konsultointipalvelut kattavat koko asiakasprojektien elinkaaren. Jokaisessa elinkaaren vaiheessa, etenkin esisuunnitteluvaiheessa, suunnittelijat luovat tuhansittain dokumentteja, jotka kulkevat ennen julkaisua tarkastusmenettelyn läpi. Muutokset projektin yhdellä osa-alueella saattavat vaatia useiden muiden dokumenttien päivytyksen, joten dokumenttien ja niiden versioiden lukumäärä kasvaa jatkuvasti projektin edetessä. Maailmanlaajuisessa yrityksessä projektit usein toteutetaan hajautetusti, jolloin eri työvaiheet suoritetaan eri haarakonttoreissa tai peräti eri maissa.

Dokumenttien tarkastamiselle ei ole ollut yhtenäistä työtapaa. Jokainen tarkastustyötä suorittanut henkilö on kehittänyt oman työmenettelyn ja kirjanpitotavan järjestyksen säilymiseksi vaihtelevalla menestyksellä. Etenkin laajoissa projekteissa dokumenttiversioiden hallinta on erittäin vaikeaa ilman sitä varten suunniteltua työkalua. Tämän insinööriyön tarkoituksena on paikata edellä kuvattu tyhjiö ja yhtenäistää dokumenttientarkastusprosessi sijainnista, muista työtavoista ja kellonajoista riippumattomaksi. Yhtenä merkittävimpana edistysaskeleena on tarkastusmenettelyn siirtäminen täysin digitaaliseen muotoon, jossa kommentointi suoritetaan dokumentin PDF-muotoiseen katselutiedostoon.

Insinööriyön luvussa 2 käydään yleisesti läpi dokumenttien käsittelyprosessia, nykyistä käytäntöä ja olemassa olevia tietojärjestelmiä, joita Flow-sovelluksen kehityksessä voidaan hyödyntää. Kolmannessa luvussa keskitytään itse sovelluksen ominaisuuksien rajaamiseen, tekniikoiden valintaan ja sovelluksen käyttöliittymän sekä tietojen

tallentamisen toteutukseen. Neljännessä luvussa kuvataan sovellukseen toteutettuja ominaisuuksia ja käsitellään sen toimintaa käyttäjän näkökulmasta. Viimeisessä luvussa syvennytään sovelluksen koekäyttövaiheeseen, havaittuihin ongelmiin, niiden korjauksiin ja tulevaisuuden kehityskohteisiin. Liiteosiosta löytyy tarkastusprosessiin ja Flow-sovellukseen liittyviä vuokaavioita ja esimerkki JPDoc-rajapinnan XML-listauksesta. Insinööriyön kuvissa olevat projektit ja dokumentit sekä tilanvaihdokset ovat täysin fiktiivisiä, eikä niille siten ole totuus pohjaa reaali maailmassa.

2 Dokumentinkäsittely

2.1 Dokumentinkäsittelyprosessin periaate

Dokumentti on tietokokonaisuus, joka on suunniteltu ihmisen aistittavaksi ja ymmärrettäväksi. Dokumentin sisältö muodostuu osista, jotka voivat olla tekstiä, kuvia tai ääntä. Sisältö voi olla hierarkkisessa järjestyksessä, eli osat voivat sisältää toisia osia. Digitaalinen dokumentti on dokumentti, joka tuotetaan ja esitetään tietokoneen avulla. Tavallisesta dokumentista poiketen digitaalisessa dokumentissa voi olla useita ulkoisia esitysmuotoja eli keinoja, jolla dokumentin sisältö esitetään sen tarkastelijalle. Käytännössä dokumentti-sanalla voidaan viitata dokumenttiin tai digitaaliseen dokumenttiin. [1.]

Prosessi tarkoittaa vaiheittaista tilanmuutosta [2]. Dokumentinkäsittelyprosessilla tarkoitetaan tässä insinööriyössä vaiheittaista dokumentin tilanmuutosta, joka aiheutuu useiden henkilöiden työosuuden aiheuttamasta tiedon lisäämisestä, muuttamisesta tai poistamisesta yksittäisessä dokumentissa. Suunnitteluprosessilla tarkoitetaan samaa asiaa, mutta korostetaan, että dokumentti suunnitellaan työprosessin alkuvaiheessa.

Suunnitteluprosessissa on yksinkertaisimmillaan dokumentin suunnittelija ja tarkastaja, joka hyväksyy tai hylkää dokumentin. Hyväksytty dokumentti julkaistaan ja toimitetaan projektin eri osapuolille ja asiakkaalle. Hylätty dokumentti palaa korjausmerkittynä suunnittelijalle, joka korjaa havaitut virheet ja toimittaa dokumentin takaisin tarkastajalle. Prosessi on yksinkertainen, kun dokumenttien määrä pysyy rajallisena ja siihen kuuluvat henkilöt sijaitsevat lähellä toisiaan, jotta he kykenevät kasvokkain tapahtuvaan keskusteluun. Käytännössä prosessissa on yleensä vielä mukana jokin välivaihe, jonka kolmas henkilö tekee suunnittelijan tuotoksen perusteella. [3.]

Yleistasolla voidaan ajatella yksittäisen dokumentin olevan dokumentinkäsittelyprosessissa eteenpäin virtaava aine, joka väistämättä määrättyjen putousten jälkeen päättyy valmiiksi ja julkaistavaksi. Kussakin projektissa, tarpeiden mukaisesti, dokumenteilla voi olla eri määrä väliputouksia. Jokaisella väliputouksella työskenteleviä ihmisiä kutsutaan työryhmiksi, ja he voivat tarpeen vaatiessa palauttaa

dokumentin edelliselle tasolle tai päästää sen virtausprosessissa seuraavalle askeleelle. Liitteessä 1 on kuvattu perustarkastusprosessin vuokaavio.

2.2 Nykyinen dokumentinkäsittelyprosessi

Gloaalissa yrityksessä työntekijät sijaitsevat usein eri konttoreissa ja peräti eri maissa, joten heillä ei ole luontevaa keskusteluyhteyttä jo pelkästään työaikojen erilaisuuden vuoksi. Nykyisessä dokumentinkäsittelyprosessissa käytetään dokumenttien paperiversioita. Tarkastaja tulostaa suunnittelijan tekemät dokumentit ja merkitsee käsin mahdolliset korjaukset paperille, lukee kuvanlukijalla hylätyt dokumentit takaisin digitaaliseen muotoon ja lähettää ne lopulta sähköpostilla suunnittelijalle. Pelkästään edellä mainittu työvaihe usein riittää sekoittamaan loogisesti jaettujen dokumentti-ryhmien onnistuneen seuraamisen jakamalla yksittäiset dokumenttiryhmät hyväksytyihin ja hylättyihin dokumentteihin. Kun tähän vielä lisätään dokumenttien *revisiot* eli uudistetut versiot, on kokeneempikin tarkastaja kirjanpitoineen ja paperipinoineen jo pulassa. [4.]

Tarkastajien tulisi lisäksi tietää kunkin dokumentin suunnittelusta julkaisuun kulunut aika, jotta projektin tilaa ja tehokkuutta voidaan valvoa ja pidemmällä tavoitteella myös kehittää parempaan suuntaan. Jokaisella tarkastajalla on ollut oma tapansa hallinnoida kirjanpitoa, mikä on vaihdellut Microsoft Excel -taulukoista käyttöjärjestelmän kansioihin ja käsimuistiinpanoihin. Työntekijöitä haastatteleamalla kävi ilmi edistysaskeleen suuri tarve, sillä nykyään tarkastustyötä tekevä henkilö ei pysty olemassa olevin työkaluin riittävän tehokkaaseen ja virheettömään työskentelyyn. [3; 4; 5; 6.]

2.3 Olemassa olevat tietojärjestelmät ja sovellukset

2.3.1 Dokumentinhallintajärjestelmä JPDoc

JPDoc on Pöyryn sisäisesti kehittämä dokumentinhallintajärjestelmä, jota suoritetaan Lotus Domino -palvelimella. JPDoc-dokumentinhallintajärjestelmiä on Pöyryllä

maailmanlaajuisesti käytössä kaikkiaan 26. Niissä saattaa olla versioeroavaisuuksia. Dokumentihallintajärjestelmien määrä vaihtelee, mikäli uusia konttoreita otetaan käyttöön tai niitä poistuu käytöstä. Käytännössä kaikilla isoimmilla konttoreilla on oma järjestelmä ja jokaisen työntekijän selatessa projektitietoja ottaa Notus Client yhteyden lähimpään tietokantaan. Kaikkien järjestelmien hallinta suoritetaan Pöyryn pääkonttorissa Vantaalla.

Paikallisesti toteutettavien projektien tietoja ei jaeta muihin JPDoc-järjestelmiin lainkaan, mutta hajautetusti toteutettavissa projekteissa on tarve jakaa tietoa eri toimistojen välillä. Dokumenttien metatiedot ovat tekstimuotoista tietoa, joten ne eivät vaadi merkittävää tallennus- ja siirtokapasiteettia, mutta jo yksittäisen dokumentin CAD-suunnittelujärjestelmästä tuotettu työtiedosto saattaa olla kooltaan 50 megatavua. [7.]

Tietojen jakaminen JPDoc-tietojärjestelmien kesken

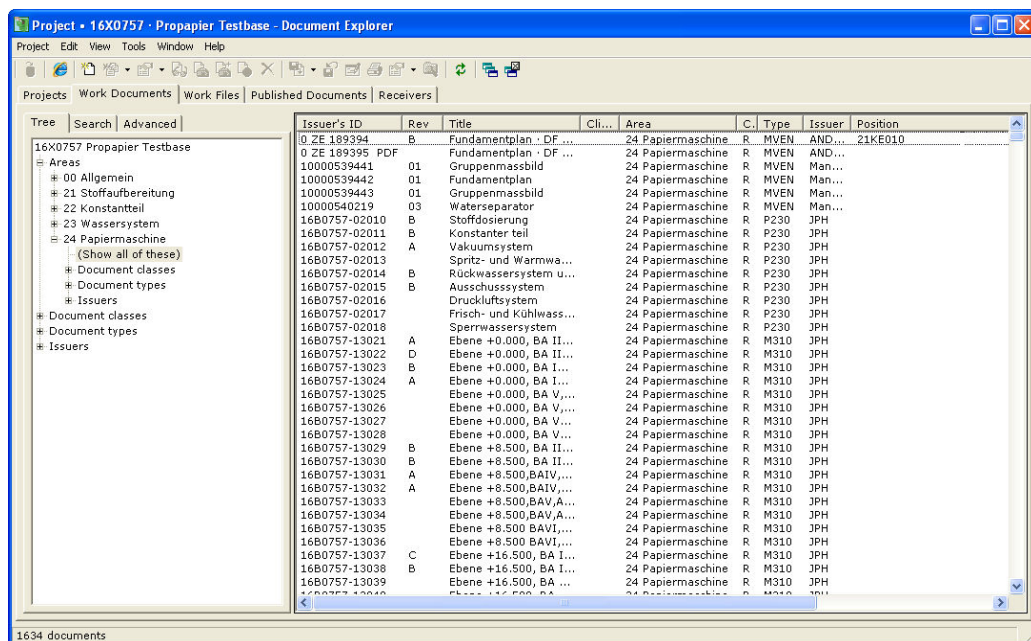
Ensimmäinen ja suositeltavampi tietojen jakamisen vaihtoehto on toimistojen määrittäminen isäntä-orjasuhteella, jolloin kaikki toimistot käyttävät todellisuudessa isännäksi määritellyn toimiston JPDoc-tietojärjestelmää. Tällöin kaikki tieto on järjestelmässä jatkuvasti ajan tasalla, mutta huonona puolena on käytettävä HTTP-protokollan avulla tapahtuvaa siirtoa, mikä saattaa olla erittäin hidasta vähemmän kehittyneillä maapallon osa-alueilla. Tämä vaihtoehto kuitenkin toimii esimerkiksi Euroopan alueella lähes moitteettomasti.

Toinen vaihtoehto on automaattisesti tapahtuva tietojen replikointi. Suuret tiedostokoot tuhansissa dokumenteissa yhdistettynä hitaisiin siirtonopeuksiin saattaa aiheuttaa päivienkin viiveitä JPDoc-järjestelmien tietojen päivitykseen toisista toimistoista. Lisäksi replikointi tapahtuu täysin automaattisesti, eikä siitä jää tutkittavaa lokitiedostoa, joten käyttäjä ei voi käytännössä ikinä olla varma, onko tietojärjestelmässä oleva tieto ajan tasalla vai ei. Tietojen replikointi on mahdollista optimoida käyttämällä monimutkaista *hämähäkkiperiaatetta*, jolloin hidasta yhteysväliä käytetään ainoastaan kertaalleen.

Tulevaisuudessa tavoitteena on vähentää replikoinnin tarvetta siten, että tiedostoja replikoitaisiin ainoastaan tarvittaessa. Kooltaan pieniä metatietoja voitaisiin säilyttää useassa tietojärjestelmässä, mutta fyysisesti tiedostot sijaitsivat aina siinä tietojärjestelmässä, jota lähimpänä dokumenttiin liittyvää tiedostoa on viimeksi muokattu. Vasta käyttäjän pyytäessä tiedoston avausta tai tallennusta, tiedosto ladattaisiin HTTP-yhteydellä vieraasta tietojärjestelmästä. [7.]

2.3.2 Document Explorer

Document Explorer on JPDocin itsenäinen Visual Basic -ohjelmointikielellä tehty käyttöliittymäsovellus, joka hyödyntää Notes-tietokannan COM-rajapintaa. Pöyryn työntekijät käyttävät sitä dokumenttien selaamiseen ja niiden tietojen muuttamiseen. Document Explorerin käynnissäolo on elinehto myös monelle siihen rakennetulle rajapinnalle, kuten Autodeskin valmistamalle AutoCAD-suunnitteluohjelmalle. Document Explorerin käynnissä oloa tullaan hyödyntämään myös Flow-sovelluksessa, sillä sen funktioita tullaan käyttämään sovelluksien välisessä kommunikaatiossa ja dokumenttiin liittyvien tiedostojen päivittämisessä JPDocin.



Kuva 1. JPDoc-dokumentinhallintajärjestelmän käyttöliittymä Document Explorer.

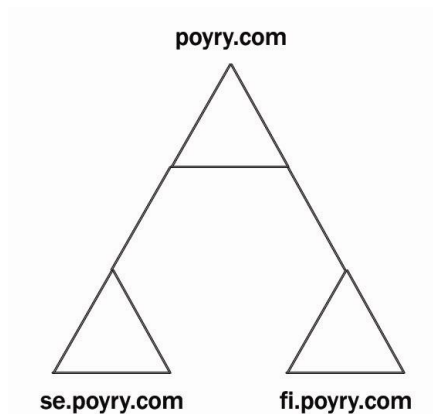
Document Explorerin käyttöliittymässä valitaan aluksi projekti, minkä jälkeen siihen kuuluvia dokumentteja voidaan listata suodattimia käyttäen (kuva 1). Esimerkkejä suodattimista ovat dokumentin tyyppi ja alue, joihin jokainen projekti on jaettu. Tätä logiikkaa pyrittiin myös mahdollisuuksien mukaan pitämään Flow-sovelluksen käyttöliittymäsuunnittelun pohjana, jotta käyttäjien oppimiskynnys madaltuisi ja siten siirtyminen uuden työkalun käyttöön olisi mahdollisimman sujuvaa. Document Explorerissa voi rekisteröidä uusia dokumentteja, muokata olemassa olevien dokumenttien metatietoja, julkaista dokumentteja ja korvata dokumenttiin kuuluvia tiedostoja, kuten työ- ja katselutiedostoja. Document Explorerissa on myös näkyvissä dokumentteihin kuuluvien tiedostojen edelliset versiot, joita käyttäjä voi halutessaan tarkastella tai ladata. [8.]

2.3.3 Henkilötietojärjestelmät

Microsoft Active Directory

Active Directory, josta käytetään yleisesti lyhennettä AD, on Microsoftin kehittämä teknologia, joka tarjoaa useita verkkopalveluita, kuten LDAP-hakemistopalvelut, Kerberos-pohjaisen käyttäjätunnistuksen ja DNS-pohjaisen nimeämisen. Active Directoryn ensimmäinen versio julkaistiin osana Windows 2000 Server Editionia.

Active Directoryn rakenne on hierarkkinen, joten siitä muodostuu puumainen. Ensimmäistä juuritasoa kutsutaankin nimellä *forest* eli metsä. Metsässä voi olla puita, jotka ovat yhden tai useamman verkko-osoitteen kokoelmia. Metsässä olevat puut luottavat toisiinsa, joten niiden sisällä olevat objektitkin luottavat ja siten kykenevät vaihtamaan tietoja keskenään. Objektit voivat olla resursseja, palveluita tai käyttäjiä, ja ne on sijoitettu aina johonkin puuhun. [9.]



Kuva 2. Esimerkki Active Directoryn sijaintiin perustuvasta puurakenteesta.

Pöyryn Active Directory -metsä on jaettu maantieteellisen sijainnin mukaan verkko-osoitteisiin (kuva 2). Kukaan verkko-osoite on jaettu pienempiin *organisaatioyksiköihin* (OU, Organization Unit) toimiston, sisäisen yrityksen ja käyttäjäryhmän mukaan. Näin voidaan tarvittaessa tehdä rajattu haku vain tietyn toimiston työntekijöistä paikallista projektia varten.

Pöyryn toiminta on vuonna 2009 jakaantunut 49 eri maahan ja toimistoja on noin 200. Monet toimistoista ovat entisiä itsenäisiä yrityksiä, joilla on itsellään jo toimiva käyttäjienhallintajärjestelmä, joka on useimmissa tapauksissa Active Directory. Eri Active Directory -järjestelmien yhtenäistäminen ja yhdistäminen olisi niin suuri työ, ettei siihen Pöyryllä, ainakaan toistaiseksi, ole ryhdytty. Pöyryllä Active Directoryyn on tallennettu etunimi, sukunimi ja Windows-järjestelmän käyttäjätunnus. [10.]

IBM Lotus Notes

Tietokonealan yritys IBM kehitti Lotus Notesin ensimmäisen version vuonna 1989, mutta sen juuret ovat 1970-luvun alulta. Viimeisin versio on 8.5, mikä julkaistiin 6.1.2009. Lotus Notes tarjoaa käyttäjälle pääsyn IBM Lotus Domino -palvelimella oleviin sähköpostitietoihin, yhteisiin kalentereihin ja ohjelmiin. [11.]

Pöyryllä käytetään Lotus Notes -ohjelmaa monien järjestelmien pohjana. Oleellisia esimerkkejä ovat dokumentinhallintajärjestelmä ja yrityksen intranet. Jokaisella

työntekijällä on käytössä Notes Client sähköposti- ja kalenterinhallintaohjelmana, joihin kirjautuminen takaa globaalien pääsyn kaikkien Domino-palvelimien resursseihin. Notes-järjestelmään on tallennettu jokaisen työntekijän tärkeistä tiedoista etunimi, sukunimi ja sähköpostiosoite. Pöyryllä Notes otettiin käyttöön 90-luvun loppupuolella ja aina yritystojen kohdalla tehdään integrointisuunnitelma Notes-järjestelmän käyttöön otosta uusissa yrityksissä. [7.]

Käytettävän henkilötietojärjestelmän valinta

Ohjelman nopeuttamiseksi jouduttiin tekemään valinta edellä mainittujen kahden vaihtoehdon välillä. Optimaalinen vaihtoehto olisi ollut tietojen hakeminen molemmista järjestelmistä, niiden yhdistäminen ja tallentaminen omaan tietojärjestelmään. Tietojen ajantasaisuus pitäisi kuitenkin tarkistaa vähintään joka käynnistyskerralla, joten ohjelman käynnistyminen hidastuisi merkittävästi, minkä vuoksi päätettiin käyttää vain toista järjestelmää. Tulevaisuudessa järjestelmiä tullaan joka tapauksessa parantamaan eli tietoja yhtenäistämään, joten ”täydellisen” implementaation tekeminen olisi ollut resurssien hukkaamista ja keskittymistä epäolennaisiin asioihin.

Active Directory oli lopullinen valinta, sillä sähköpostiosoitteet voidaan lähes poikkeuksetta selvittää sieltä saatavien tietojen avulla. Pöyryn työntekijöiden sähköpostiosoitteet ovat muodossa *etunimi.sukunimi@poyry.com*. Lisäksi Active Directoryssa on valmiiksi skandinaaviset kirjaimet muutettu sähköpostikelpoiseen muotoon. Harvinaisempien kaksiossaisten nimien mukaan muodostuvat poikkeukset on määritelty erikseen User-luokkaan, josta Email-tiedon arvoa voidaan kysyä. Pääkonttorin Active Directory ei tarjoa maailmanlaajuista käyttäjätietokantaa, mutta se tarjoaa riittävän suuren käyttäjämäärän koeversion käyttämiselle. Lotus Notes-järjestelmästä olisi puuttunut lyhyt Windows-käyttäjätunnus, jota hyödynnetään Flow-sovelluksessa esimerkiksi projektien dokumenttilistauksissa ja yksittäisten dokumenttien historialistauksissa, jotta tilantarve olisi mahdollisimman pieni.

3 Flow-sovelluksen kehittäminen ja käytetyt menetelmät

3.1 Tarvemäärittely ja työn rajaus

Tarvemäärittelyn tekeminen aloitettiin erään projektin henkilöstöä haastatteleamalla. Kyseisessä projektissa tutkittiin isometripiirrosten luontia, joka käytetystä CAD-suunnittelujärjestelmästä johtuen oli kaikkiaan neliportaista. Työprosessissa oli kolme työntekijää, joita kuvataan nimillä suunnittelija, isometraaja ja isometrien tarkastaja, josta myöhempänä käytetään lyhyemmin nimitystä tarkastaja.

Ensimmäisenä työvaiheena suunnittelija ajaa valitsemastaan putkilinjasta isometriaajon. Hän pilkkoo putkilinjan selkeyden ja luettavuuden sekä osaluettelon pituuden kannalta riittävän pieniin palasiin muodostaen isometriryhmän. Tämän jälkeen hän tulostaa isometriryhmän paperille, merkitsee käsin linja- tai ryhmätunnuksen, puumerkkinsä ja päivämäärän sekä toimittaa paperipinon tarkastajalle. [5.]

Tarkastaja merkitsee mahdolliset korjausta vaativat kohdat ja toimittaa korjattavat paperit takaisin suunnittelijalle. Hyväksytyt tai vain pieniä kommentteja vaatineet paperit luetaan kuvanlukijalla ja lähetetään sähköpostin liitetiedostoina isometraajalle. Isometraaja tekee suunnittelijan työn ja kommenttien pohjalta jokaisesta isometriryhmän osasta isometrikuvan, jotka hän toimittaa takaisin tarkastajalle. Isometraaja myös rekisteröi isometrikuvasta dokumentin dokumentinhallintajärjestelmä JPDociin. [4; 6.]

Tarkastaja tulostaa isometrikuvat tarkastamista varten. Mahdolliset virheet hän merkitsee jälleen käsin ja kuvanlukemisen suorittamisen jälkeen toimittaa ne takaisin isometraajalle. Virheettömät dokumentit menevät julkaistaviksi, minkä jälkeen ne toimitetaan projektin asiakkaalle ja tarpeellisille tahoille. [4.]

Kehityksen lähtökohta

Flow-sovelluksen kehityksen lähtökohtana oli välttää täysin paperiversioiden käsittely sekä tarve lähettää dokumentteja sähköpostilla. Dokumenttien siirtäminen henkilöltä toiselle määriteltiin tapahtuvaksi sovelluksen käyttöliittymän painiketta käyttämällä. Itse dokumentit myös päätettiin varastoida yhteen paikkaan, jotta niistä ei voisi samanaikaisesti olla useita versioita, mikä poistaa versiosekaannusten mahdollisuuden. Lisäksi jokaisesta dokumentin tilanmuutoksesta jäisi tallenne tietokantaan, joten dokumentin julkaisuprosessin pituus tiedettäisiin sekunnilleen ja samalla mahdollistettaisiin projektitehokkuuden monipuolisempi tutkiminen.

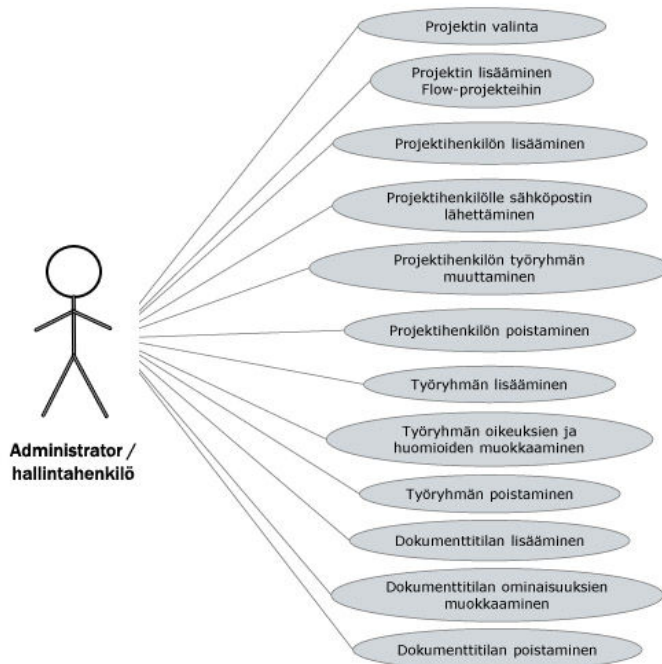
Sovelluksesta pyrittiin tekemään mihin tahansa suunnitteluprosessiin sopiva työkalu, joten tutkittua esimerkkiprojektia käytettiin vain tarkastusprosessiin periaatteen tutkimiseen, unohtamatta erikoispiirteiden olemassa oloa kussakin ohjelmistoympäristössä. Käytettävien suunnitteluohjelmien toimintoihin tulee lisätä kommunikointikeinot Flow-sovelluksen kanssa. Liitteessä 2 on kuvattu Flow-sovelluksen mahdollistamat muutokset työprosessin kulkuun.

Tutkitussa esimerkkiprojektissa Flow'n käyttöönotto tarkoittaisi suunnitteluun käytettävän Cadmatic-ohjelman isometrinluonnissa ajettavan makron muokkaamista siten, että se tekisi tarpeelliset merkinnät tietokantaan [3]. Dokumentin luonti tai rekisteröinti JPDoc-tietojärjestelmään siirrettäisiin aikaisemmaksi suunnittelijan työvaiheeseen, mutta pyrittäisiin tekemään siitä suunnittelijalle mahdollisimman automaattista. Isometraajan työosuus pysyisi lähes identtisenä, ainoastaan uutta dokumenttinumeroa ei tarvitsisi pyytää järjestelmästä, vaan se olisi esitäytettynä metatiedon täyttölomakkeessa. Metatiedot dokumentinhallintajärjestelmään olisi isometraaja täyttänyt vasta suunnitteluvaiheen toisella portaalla [6]. Tämä yksittäisen ohjelman toiminnan muuttaminen kuitenkin rajattiin tämän insinööriyön tavoitteiden ulkopuolelle ja tarkoituksena on saada aikaan yleispätevän tarkastustyökalun koeversio.

Dokumenttien kommentointi

Dokumenttien kommentointiin on olemassa useita vaihtoehtoja, joista päädyttiin käyttämään Adobe Acrobat -ohjelman tarjoamia piirtotyökaluja. Paras ja hienoin ratkaisu olisi ollut piirtotyökalujen sisällyttäminen Flow-sovellukseen ja PDF-tiedoston luominen dynaamisesti, mutta se olisi kuitenkin jo kelvannut omaksi insinööriyön aiheeksi. Perusoletukseksi asetettiin, että kullakin kommentointityötä tekevällä tulee olla Adobe Acrobat -ohjelma asennettuna. Pöyryllä ohjelma on jo käytössä, joten se ei aiheuta käytännössä suurta ongelmaa.

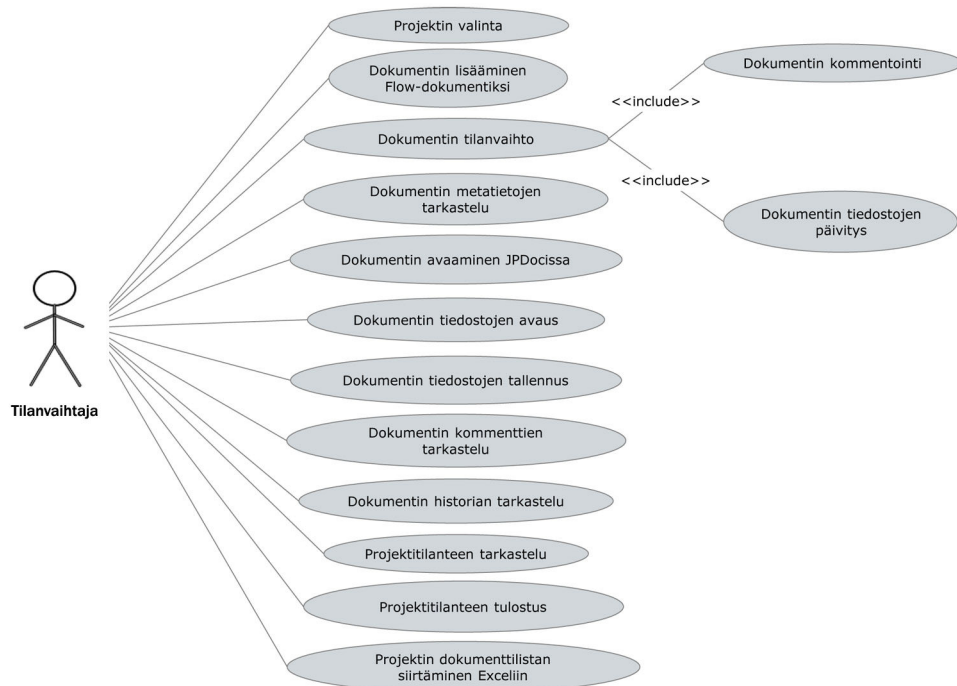
Toimijoiden kuvaukset



Kuva 3. Käyttäjän hallintatilassa olevat toimintamahdollisuudet.

Sovelluksessa tarvitaan karkeasti ottaen ainoastaan kahdenlaisia käyttäjiä, joista ensimmäiset eli hallintahenkilöt hallinnoivat projekteja, dokumenttien tiloja ja työryhmiä sekä niiden henkilöitä projektitarpeiden mukaisesti (kuva 3). Heidän projekteittain säätämien asetusten mukaisesti määräytyvät dokumenttien kulkujärjestys ja muu projektihierarkia. Pääosin projektihallinta on suunniteltu suoritettavaksi ennen projektin alkua, mutta sen on oltava mahdollista myös projektin ollessa käynnissä,

mutta joitakin toimintoja on silloin rajoitettu. Esimerkiksi työryhmiä, joissa on henkilöitä, ei ole mahdollista poistaa. Vastaavasti dokumenttitiloja, jotka ovat olemassa jollain dokumentilla, ei ole mahdollista poistaa eikä muokata.



Kuva 4. Use-case -kuva käyttäjätilassa olevista toiminnoista.

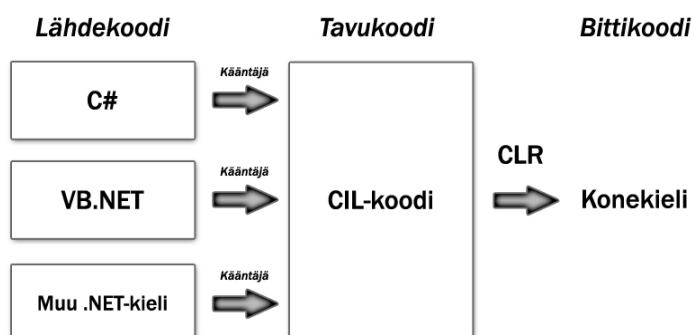
Toiset käyttäjät ovat peruskäyttäjiä, jotka ovat hallintapuolella määriteltyjä työryhmien henkilöitä, jotka kykenevät muuttamaan dokumenttien tiloja hallinnoijan antamien oikeuksien rajoissa (kuva 4). Työryhmien henkilöitä voidaan kuvata yleisesti dokumenttitilojen vaihtajiksi. Dokumenttien tilanvaihdot ja dokumenttien lisääminen Flow-dokumentiksi on sallittu ainoastaan, mikäli kyseinen henkilö on projektin työryhmissä. Kaikki muut toiminnot ovat käytettävissä, vaikka käyttäjä ei olisikaan projektissa mukana.

3.2 Kehitysympäristön ja tekniikoiden valinta

3.2.1 Microsoft Visual Studio ja .NET Framework

Microsoft Visual Studio on täysin integroitu visuaalinen kehitysympäristö, jolla voidaan luoda Windows-ohjelmia, Internet-sivustoja, -sovelluksia ja -palveluja. Integroitu-termillä tarkoitetaan sitä, että Visual Studiossa on sisään rakennettuina toimintoina käytännössä lähes kaikki mitä ohjelmointiin tarvitaan: tekstieditori, käyttöliittymänäkymä, ohjelmaikkunoiden tuki, kääntäjä ja ohjelmointivirheiden etsin. Microsoftin MSDN-dokumentaation hakupalvelu on osana kehitysympäristöä ja .NET Framework tarjoaa myös tietojenvaihtomahdollisuuden muiden ohjelmien ja tietokantojen kanssa. [12, s. 301–303.]

Visual Studio tukee C++-, VB.NET-, C#- ja J#-ohjelmointikieliä. Lisäasennuksina voidaan myös lisätä tuki muun muassa Chrome-, F#-, Python- and Ruby-kielille. Visual Studiolla käyttöliittymäsuunnittelu on helppoa ja luontevaa, sillä tuetut komponentit voidaan siirtää hiirellä käyttöliittymäpohjalle ja ohjelmoida niiden toiminnot erikseen. Visual Studion ensimmäinen versio julkaistiin vuonna 1997. Tuorein versio on Visual Studio 2008, jota kehitettiin nimellä *orcas* ja julkaistiin marraskuussa 2007. Visual Studio on Pöyryllä käytössä, joten se oli luonnollinen valinta Flow-sovelluksen kehitysympäristöksi.



Kuva 5. .NET-arkkitehtuurin kielimuunnokset, jossa ensin kielen kääntäjällä lähdekoodi käännetään CIL-välikielille ja vasta ajonaikainen käyttöympäristössä (Common Language Runtime) tavukoodi muutetaan biteiksi.

Microsoft aloitti .NET Framework -ohjelmistokomponenttikirjaston kehittämisen 1990-luvun lopulla kehitysnimellä *Next Generation Windows Services* (NGWS).

Ensimmäinen virallinen versio julkistettiin vuonna 2002. Uusin versio 3.5 julkaistiin vuoden 2007 lopussa ja siihen ensimmäinen päivityspaketti elokuussa 2008.

Ohjelmistokomponenttikirjasto kehitettiin ratkaisemaan yleisimmät ohjelmointiongelmat, ja se sisältää virtuaalikoneen, joka hallinnoi ohjelmien suorittamista. Kirjasto sisältää runsaan määrän valmiita luokkia, joten toistuvien perustoimintojen koodaaminen itse ei ole enää välttämätöntä tai edes järkevää. Lisäksi on mahdollista ohjelmoida eri kielillä ohjelmakomponentteja, jotka käännetään ensin *abstraktiin välikielimuotoon* (Common Intermediate Language) tavukoodiksi ja jotka kykenevät keskinäiseen vuorovaikutukseen lähdekoodin ohjelmointikielystä riippumatta, mikä antaa ohjelmoijalle mahdollisuuden valita ohjelmointikieli, joka hänelle parhaiten sopii (kuva 5). [13; 14.]

3.2.2 C#-ohjelmointikieli (C-Sharp)

C# on Microsoftin kehittämä suhteellisen uusi oliopohjainen ohjelmointikieli.

Syntaksiltaan se muistuttaa C++- ja Java-ohjelmointikieliä, mutta ohjelmointivirheiden välttämiseksi esimerkiksi *osoittimien* (pointer) käyttö suoraan on poistettu käytöstä.

Kielen kehittämisen tavoitteina oli kehittää yksinkertainen, moderni ja yleiskäyttöinen olio-ohjelmointikieli.

C#-ohjelmointikielellä on .NET Frameworkin ominaisuutena oma *roskienkeruu* (garbage collection), joka tarkoittaa varattujen muistiresurssien vapauttamista automaattisesti tietyin väliajoin ilman, että ohjelmoijan tarvitsee niitä vapauttaa.

Saavutetun helppouden ja yksinkertaisuuden hintana on tehokkuus. C#-ohjelma ei ole yhtä tehokas muistinkäytön ja resurssien suhteen kuin vastaava C- tai C++-ohjelma, mutta sen kehittäminen on usein huomattavasti nopeampaa. [12 s. 13–14.]

C#:in pääsuunnittelijana toimi Anders Hejlsberg, joka on aiemmin ollut kehittämässä muita ohjelmointikieliä, kuten Turbo Pascalia ja Borlandin Delphiä. C#-kieltä on kritisoitu Javan kloonikieleksi, mutta Hejlsberg on painottanut, että kielen kehityksessä

on pyritty yhdistämään C++-, Java-, Modula 2- ja Smalltalk-ohjelmointikielien parhaat puolet, eikä ainoastaan tehdä omaa versiota toisesta kielestä. Julkaisuhetkellä innovatiivisimpia toimintoja olivat muun muassa komponenttiohjelmoinnin kehittäminen *boxing*- ja *unboxing*-toiminnoilla, joilla mikä tahansa tietotyyppi voidaan muuttaa objektityypiksi ja päinvastoin. Muita merkittäviä uudistuksia olivat muistihallinnan roskienkeruu, XML-tekniikan integrointi ja ohjelmakoodin generointimahdollisuus XML-pohjaisesta XSD-skeematiedostosta. [15.]

3.2.3 Adoben PDF-tiedostomuoto

Nykyisessä Pöyryn dokumentinhallintajärjestelmässä on tallennettu jokaisen dokumentin alkuperäinen työtiedosto, joka voi olla tiedostotyypiltään esimerkiksi CAD-suunnittelujärjestelmän tai Microsoft Word -ohjelman tuotos, eli tiedostotyyppi ei ole etukäteen tiedossa. Tämän lisäksi dokumenteista on myös tallessa *katsottava tiedosto* (Viewable file), jonka tulee aueta laitteistosta tai käyttöjärjestelmästä riippumatta aina samanlaisena. Tähän tarkoitukseen parhaiten sopii Adoben kehittämä *dokumenttiformaatti PDF* (Portable Document Format).

Adobe Systems kehitti nykyään ANSI- ja ISO-standardoidun PDF-tiedostoformaatin vuonna 1990. Sen suosio kasvoi merkittävästi, kun Adobe ryhtyi jakamaan Adobe Acrobat -lukuohjelmaa ilmaiseksi vuonna 1994. PDF-dokumenteissa voi olla vektorigrafiikkaa, rasterikuvia, tekstiä ja interaktiivisia elementtejä, kuten lomakkeen kenttiä tai hyperlinkkejä. PDF tukee lisäksi läpinäkyvyyttä. Dokumentit on mahdollista suojata salasanalla tai ne voidaan allekirjoittaa digitaalisesti. [16.]

PDF-dokumentit myös osaavat tarvittaessa lukea itsensä ääneen, mikä on kasvattanut formaatin suosiota myös näkövammaisten keskuudessa. Adobe Systems yhä omistaa oikeudet tiedostoformaattiin, mutta sen käyttö on tekijänoikeusmaksuista vapaata eli kuka tahansa voi luoda ohjelman joka luo, editoi tai lukee PDF-tiedostoja. [17.]

3.2.4 Relaatietietokanta ja SQL-kieli

SQL-lyhenne tulee sanoista *Structured Query Language*, joka tarkoittaa rakenteellista kyselykieltä. Nykyään SQL on ANSI- ja ISO-standardisoitu, mutta sen kehitys on peräisin varhaiselta 1970-luvulta, jolloin IBM:n palveluksessa Donald C. Messerly ja Raymond F. Boyce kehittivät sen edeltäjää SEQUEL:ia San Josen kehityslaboratoriossa Yhdysvalloissa. Kieli on kehitetty relaatiotietokantojen kyselykieleksi. [18.]

Tietokannan taulu		
	Sarake 1	Sarake 2
Tietue 1	Kenttä	Kenttä
Tietue 2	Kenttä	Kenttä
Tietue 3	Kenttä	Kenttä
Tietue 4	Kenttä	Kenttä

Kuva 6. Esimerkki tietokannan taulusta.

SQL-palvelimella voi olla useita tietokantoja, joten kyselyitä suoritettaessa on aina yhteyden luomisen jälkeen valittava tietokanta, jota halutaan käsitellä. Tietokanta muodustuu tauluista ja taulut tietueista ja kentistä, joista vähintään yhden on oltava avainkenttä (kuva 6). Sana relaatio tulee taulujen välisistä suhteista, sillä toisessa taulussa voidaan viitata ainoastaan toisen taulun avainkenttään, jolloin päästää käsiksi koko toiseen tietueeseen kopioimatta kaikkia tietoja. Kentät voivat olla muun muassa automaattisesti inkrementoituvia, niitä voidaan määritellä vieraan taulun avaimiksi ja niiden voidaan pakottaa sisältävän jokin arvo (ei tyhjä, *not null*). Näitä ominaisuuksia hyödynnettiin Flow-sovelluksen käyttämässä tietokannassa runsaasti, jotta tietoa ei tarvitse pitää kahdessa eri paikassa. Yleisimpiä kenttien tietotyypppejä ovat kokonaisluku *int*, vaihtelevan pituinen merkkijono *varchar* ja aika *timestamp*. Kyselyitä on mahdollista tehdä useista tauluista kerralla, ja ne ovat nopeita suuristakin tietokannoista. [19.]

Pöyryllä on käytössä useita SQL-tietokantoja, joten niiden ylläpito ei ole ainoastaan yhdestä tai edes muutamasta henkilöstä riippuvaista. Tietokannat yleisesti ovat tietojen hakemiseen optimoituja ja tietokannan suora käyttäminen varmistaa hauista nopeasti saatavat vastaukset. Näiden syiden perusteella myös Flow-sovelluksessa päädyttiin käyttämään SQL-tietokantaa.

3.2.5 Microsoft SharePoint ja dokumenttikirjasto

SharePoint-palvelut (SharePoint Services) on kokelma .NET-ohjelmistokomponentteja, jotka on rakennettu Windows Server 2003:n päälle. Palvelut koostuvat kahdesta läheisestä sovelluksesta: *Windows SharePoint Servicestä* (WSS) ja *SharePoint Portal Servicestä* (SPS). Molemmat sovellukset hyödyntävät yhteistä SQL-tietokantaa sisällön varastointisijaintina. SharePointissa on sisäänrakennettuja web-palveluita (Web services), joita voidaan kutsua Internet-selaimen URL-osoitteilla. Lisäksi sovelluskehittäjät voivat luoda omia palveluitaan hyödyntämällä XML-pohjaista *SOAP-tietoliikenneprotokollaa* (Simple Object Access Protocol). [20.]

Microsoftin mukaan WSS on monimuotoinen teknologia, jota kaiken kokoiset yritykset voivat käyttää tehokkuuden ja tuottavuuden parantamiseksi. WSS antaa työntekijöille mahdollisuuden päästä käsiksi yhteisiin tietoresursseihin riippumatta organisaatioeroista tai maantieteellisistä rajoista. WSS on osa Microsoft Server 2003:a ja se toimii perustana Web-pohjaisten mukautettaville ja laajennettaville business-sovelluksille. Web-pohjainen ja jokapäiväisestä tuttu käyttöliittymä mahdollistaa sivustojen helpon ja nopean julkaisun. [21.]

Dokumenttikirjastot ovat palvelimella sijaitsevia tiedostokokoelmia, joita voidaan jakaa yhteisinä resursseina SharePoint-sivun käyttäjille. Tiedostojen etsiminen, käsittely ja kommentointi voidaan hoitaa Internet-selainta käyttämällä. Käyttäjien on mahdollista lisätä omalta tietokoneelta dokumentteja kirjastoon, muokata ja poistaa niitä. Dokumenteilla voi olla tiedoston lisäksi tietokenttiä, joilla voidaan asettaa vaadittu tietotyyppi tai haluttuja reunaehtoja. [22.]

Pöyryllä ensimmäinen SharePoint-palvelin otettiin käyttöön vuonna 2006. Noin vuoden käyttämisen jälkeen suosion kasvaessa otettiin käyttöön erillinen kehityspalvelin, joten lopullinen tuotantopalvelin eristettiin vakaaksi ympäristöksi. *Sivukokoelmia* (Site collection) Pöyryllä on käytössä 110 ja sivuja kaiken kaikkiaan noin tuhat. Sivustojen koot vaihtelevat kymmeneen gigatavuun asti. Tuotantopalvelimen sisältö on julkaistu Internetiin, mutta kehityspalvelimelle pääsy vaatii VPN-yhteyden tai yhteyden Pöyryn verkkoon. Pöyryllä SharePointin suosio kasvaa jatkuvasti arviolta kahden sivun viikkovauhdilla sen helppokäyttöisyyden ja kustannustehokkuuden ansiosta. [23.]

3.3 Olemassa olevien tietojärjestelmien hyödyntäminen

3.3.1 Käyttäjätietojen hakeminen LDAP-verkkoprotokollan avulla

LDAP (Lightweight Directory Access Protocol) on hakemistopalvelujen käyttöön tarkoitettu verkkoprotokolla, joka käyttää TCP/IP-verkkopalvelua. Sen edeltäjä oli X.500-standardi, joka oli kuitenkin liian sidottu OSI-protokollaan. X.500 käytti monimutkaista koodausmenetelmää, joka oli usein turhan raskas palveluihin, joita todellisuudessa tarvittiin. LDAP tarjoaa vain tärkeimmät ja välttämättömimmät X.500-standardin palveluista. Hakemistopalvelimiin sijoitetut tiedot on järjestetty avain-arvopareina ja puut on järjestetty oliohierarkiaan maantieteellisen tai organisaation rakenteen mukaisesti, eikä verkon rakenteella ole merkitystä siihen.

LDAP-verkkoprotokollan rajapinta tarjoaa seuraavat toiminnot: *kytkeydy* (bind), *pura yhteys* (unbind), *etsi* (search), *muokkaa* (modify), *lisää* (add), *poista* (delete) ja *luovuta* (abandon). Näistä toiminnoista kytkeytyminen aloittaa istunnon asiakasohjelman ja palvelimen välille ja tällöin käyttäjän identiteetti tunnistetaan Flow-sovellukseen kirjautuneiden tietojen perusteella. Etsi-toiminto on optimoitu palvelimilla ja suurikin määrä hakuja pystytään hoitamaan suorituskykyisesti. [24.]

Muokkaa-, lisää- ja poista-toiminnot mahdollistavat hakemistopalvelun tietojen muokkaamisen, mutta tässä insinööriyössä keskitytään ainoastaan tietojen hakemiseen,

joten toiminnot kytkeydy, etsi ja pura yhteys riittävät mainiosti. LDAP-verkko-protokollalla voidaan suorittaa nopeita kyselyitä sekä Pöyryn Active Directory -järjestelmästä että Lotus Domino -palvelimelta. [7; 10.]

3.3.2 Tietojen hakeminen JPDocista rajapinnan avulla

JPDoc on kirjoitettu Visual Basic -ohjelmointikielellä. Visual Basicistä on olemassa myös palvelimen skriptikieli, jota kutsutaan nimellä VBScript. Lotus Domino Designer -ohjelmalla on mahdollista kirjoittaa omia palveluita palvelimelle LotusScript-kielellä, joka on erittäin lähellä VBScriptiä ja sisältää ainoastaan muutaman pienen syntaksieron. LotusScriptiä kutsutaankin Basic-kielen murrekieleksi. Pöyryn sisäisenä projektina on ohjelmoitu rajapinta LotusScriptillä JPDociin, joka sisältää välttämättömimmät dokumentinhallintajärjestelmän toiminnallisuudet. Osa toiminnoista on jäänyt hieman vajavaisiksi, mikä aiheutti Flow-sovellusta kehitettäessä joitakin improvisointeja, kuten projektin luodun muodostamisen alavetovalikoihin. Kaikki muu tieto saadaan palvelimelta XML-dokumenttina, paitsi projektin luodun, joka on mahdollista saada ainoastaan HTML-dokumenttina. HTML-dokumentille generoitava projektin luodun rakennettiin alun perin vain järjestelmän kehittäjän omaa koekäyttöä varten. [7.]

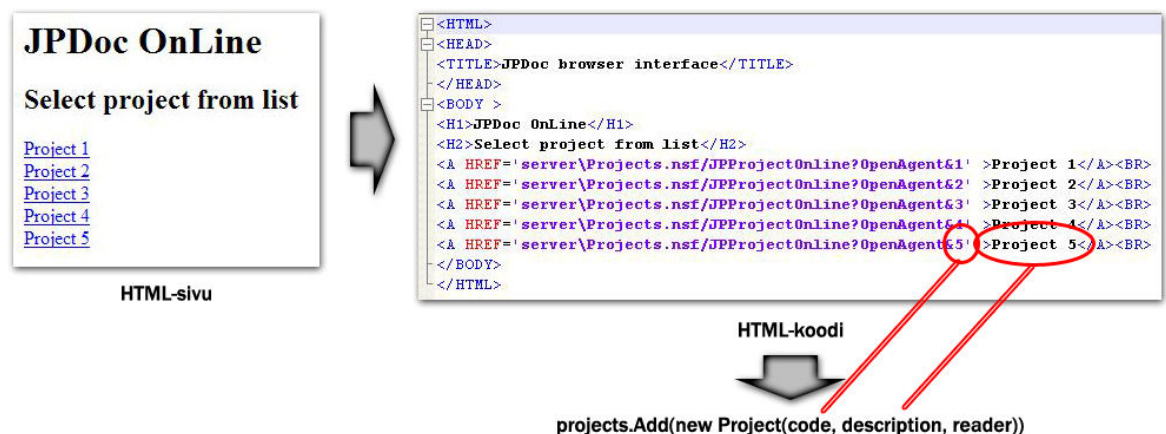
Flow-sovellukseen tehtiin JPDocin lukemista varten oma luokka JPDocReader, jolla on staattisina muuttujina palvelimen HTTP-osoite sekä dokumenttien ja niiden ominaisuuksien listauksiin tarvittavat rajapinnan funktiokutsut. JPDocReader-olio luodaan jokaiselle käyttäjälle ohjelman suorituksen aikana ainoastaan kerran, sillä se sisältää käyttäjätiedot, joita tarvitaan kaikkiin järjestelmiin kirjautuessa. Olio välitetään parametrina kaikkiin sitä tarvitseviin luokkiin. Projekti- ja dokumenttitietojen hakeminen, kuten tiedostojen latauskin, toteutetaan omassa säikeessä, jotta käyttöliittymäkomponenttien piirtotoiminnot eivät jumiudu tai häiriinny. Tämä mahdollistaa myös käyttöliittymän alaosassa sijaitsevan *tilapalkin* (Status bar) päivittämisen kunkin tapahtuman mukaisesti.

XML-tiedon jäsentäminen

Yksinkertaisimmillaan JPDoc-tietojärjestelmästä voidaan kysyä URL-polkua käyttämällä tietyn dokumentin tai tiettyjen dokumenttien metatiedot ja siihen tai niihin liittyvät tiedostot. Vastaus tällaisesta kyselystä on kuvattu liitteessä 3. Käytännössä JPDocin-tiedonhakurajapinta sisältää kaksi hieman poikkeavaa kutsua, jotka palauttavat osin päällekkäisiä ja osin eriäviä tietoja. Näin ollen jokaista dokumenttilistan muodostamista varten on tehtävä kaksi HTTP-kutsua. [7.]

Dokumentin metatiedon jäsenitys on jaettu projekti- ja dokumenttiluokan metodeihin (liite 4). Projektiluokassa jäsenetään JPDocReader-luokan ilmentymältä palautunut XML-dokumentti. Jokaisen XML-elementin sisältöineen välitetään dokumentille, joka tunnistaa lähetettävän tiedon elementin nimestä ja tallentaa elementin sisällön arvokseen. Näin jäsentämisen päätteeksi projektilla on tieto siihen kuuluvista projekteista tietyillä hakuehdoilla.

HTML-dokumentin jäsentäminen



Kuva 7. HTML-dokumentin jäsentäminen (parsiminen) projektilistaksi on suoraviivaista, sillä jokaisen linkin kohdalta etsitään href-parametrin dokumentin tunnus ja selaimella näytettävästä tekstistä saadaan projektin kuvaus.

HTML-dokumentin jäsentämiseen tehtiin oma luokka nimeltä HtmlParser, jolle välitetään parametrina JPDocReader-luokan olio. Luokan ainoana metodina on *parse*, joka vastaanottaa parametreina täytettävän projektilistan ja HTML-koodin String-

muuttujana. HTML-dokumentti muodostuu kutsumalla JPDocin rajapinnan *jpprojectlist?openagent*-palvelua. Koodista parsitaan projektikoodi ja projektin kuvaus, jotka välitetään JPDocReader-olion lisäksi uuden projektiolion konstruktorille (kuva 7). Uusi projektiolio lisätään projektilistaan ja tätä toistetaan kunnes dokumentin *runko-osio* (body) loppuu. Vaatimuksena on myös, että mikään parametrina vietävistä tiedoista ei ole tyhjä tai hypätään suoraan seuraavaan projektiin, jolloin kyseessä olevaa projektia ei myöskään näy kerättävistä tiedoista muodostettavassa alavetovalikossa. Merkkijonojen käsittelyssä hyödynnetään monipuolisia .NET Frameworkin tarjoamia valmiita String-luokan metodeita, kuten Contains, IndexOf ja Substring [25].

3.3.3 Tietojen vienti JPDociin COM-objektia hyödyntämällä

Termi sidonta viittaa vieraan ohjelman metodin kutsumiseen. Aikaisen ja myöhäisen sidonnan ero on se, että aikaisessa sidonnassa projektiin lisätään referenssi toisesta ohjelmasta. Myöhäisessä sidonnassa käytettävä metodi *herätetään* (invoke) käyttämällä .NET Frameworkin heijastusta. Heijastus on geneerinen termi, jolla kuvataan ohjelmaelementtien ajonaikaista tunnistamista ja manipulointia. COM-objektit voivat kommunikoida .NET-sovellusten kanssa ja päinvastoin. Riippuen sidontatavasta, tarpeellisten valmisteluiden jälkeen on mahdollista käyttää COM-objektin metodeita tavallisen .NET-sovelluksen luokan metodien tapaisesti. COM-objektien merkittävä ero .NET-luokkiin on se, että ne eivät vapauta varattuja muistiresursseja automaattisesti. [12 s. 257, 965–998.]

ROT-taulu

ROT-nimi tulee sanoista *Running Object Table*, mikä tarkoittaa suomeksi käynnissä olevien objektien taulua. ROT-taulu pitää kirjaa tunnistettavista, työasemassa käynnissä olevista objekteista, joita voidaan hyödyntää toisissa sovelluksissa. [26.]

Sidonnan hyödyntäminen Flow-sovelluksessa

Sidonnan ensimmäisessä vaiheessa tutkitaan, löytyykö ROT-taulusta käynnissä olevaa Document Explorer -sovellusta käyttämällä .NET Frameworkin *System.Runtime*-kirjastossa sijaitsevaa *InteropServices.Marshal.BindToMoniker*-metodia. Mikäli sovelluksen ilmentymä löytyy, palautetaan löydetty COM-objekti käytettäväksi. Mikäli objektia ei löydy, tutkitaan Windowsin rekisteristä Document Explorer -sovelluksen asennussijainti hyödyntämällä uniikkia CLSID-tunnusta, minkä jälkeen ohjelma käynnistetään hyödyntämällä *System.Diagnostics.Process*-metodia. Tämän jälkeen odotetaan, kunnes etsitty COM-objekti löytyy ja se palautetaan käytettäväksi tai virhetilanteissa näytetään käyttäjälle virheilmoitus.

Document Explorerin kehittäjän suosituksesta Flow-sovelluksen toteutuksessa päädyttiin käyttämään myöhäistä sidontaa. Tällä estetään versiomuutosten aiheuttamia sovellusten konflikteja ja siten vähennetään virheellisiä toimintoja. Näin palautettavaa *Object*-tietotyyppiä ei muunneta ennalta tiedetyksi *DocExplorer*-objektiksi, vaan vasta ohjelman suorituksen aikana tutkitaan, herätetään ja suoritetaan Document Explorerin *HandleDocument*-metodi halutuun parametrein. [8.]

3.4 Tietokannan suunnittelu ja toteutus

Tietokannan suunnittelun perustaksi otettiin luonnollisesti mahdollisimman vähäinen tiedon tallentaminen ja muualla jo tallessa olevan tiedon hyödyntäminen. Jo valmiiksi tallessa olevaa tietoa ovat työntekijöiden, projektien ja dokumenttien tiedot, joten näistä vain valittiin sopivat avaintiedot, jotka tallennetaan omaan Flow-sovelluksen käyttämään SQL-tietokantaan, joka sijaitsee Microsoft Server 2003 -palvelimella. Heti alkuvaiheessa tehtiin rajausta, että varsinaiset tiedostot tallennetaan eri paikkaan, jotta tietokannan koko pysyy pienenä ja suorituskyky mahdollisimman hyvänä.

Kaikilla tietokannan tauluilla on *pääavaimena* (primary key) *kokonaislukutyypinen* (integer) muuttuja, joka inkrementoituu automaattisesti uutta tietuetta luotaessa. Pääavainten arvot alkavat yhdestä, ja niiden maksimiarvot ovat 2 147 483 647, joka on

samalla ryhmien, dokumenttitilojen, oikeuksien ja tilanvaihtojen maksimimäärä. Mikäli tulevaisuudessa on vaara, että raja ylittyy, tulee pääavaimien ja sovelluksen muuttujien tyyppi vaihtaa *suureksi kokonaislukumuuttujaksi* (bigint), jolloin muuttujan tilantarve kasvaa neljätavuisesta kahdeksantavuiseksi ja maksimiarvo tähtitieteelliseksi reiluksi 9,2 miljardiksi miljardiksi. [27.]

Projektiasetusten muokkaamisen ehtona on, että valittu projekti on lisätty työnkulku-projekteihin. Kyseistä asiaa varten on tietokantaan luotu *työnkulkuprojektit*-taulu (workflowprojects). Taulu sisältää pääavaintiedon lisäksi avaintiedon JPDocin tietojärjestelmään eli projektitunnukseen. Toisessa toteutusvaihtoehdossa työryhmän kuuluminen kyseiseen projektiin olisi määritelty suoraan vieraan tietojärjestelmän avaintiedolla, mutta tällöin olisi aina jouduttu käyttämään projektilistauksia tehtäessä suhteellisen hidasta rajapintaa. Paremman nopeuden ansiosta päädyttiin listaamaan työnkulkuprojektit omaan tauluun ja ainoastaan niistä haetaan tietoa ulkoisesta järjestelmästä.

Kolmiportainen projektiasetuksen määrittäminen

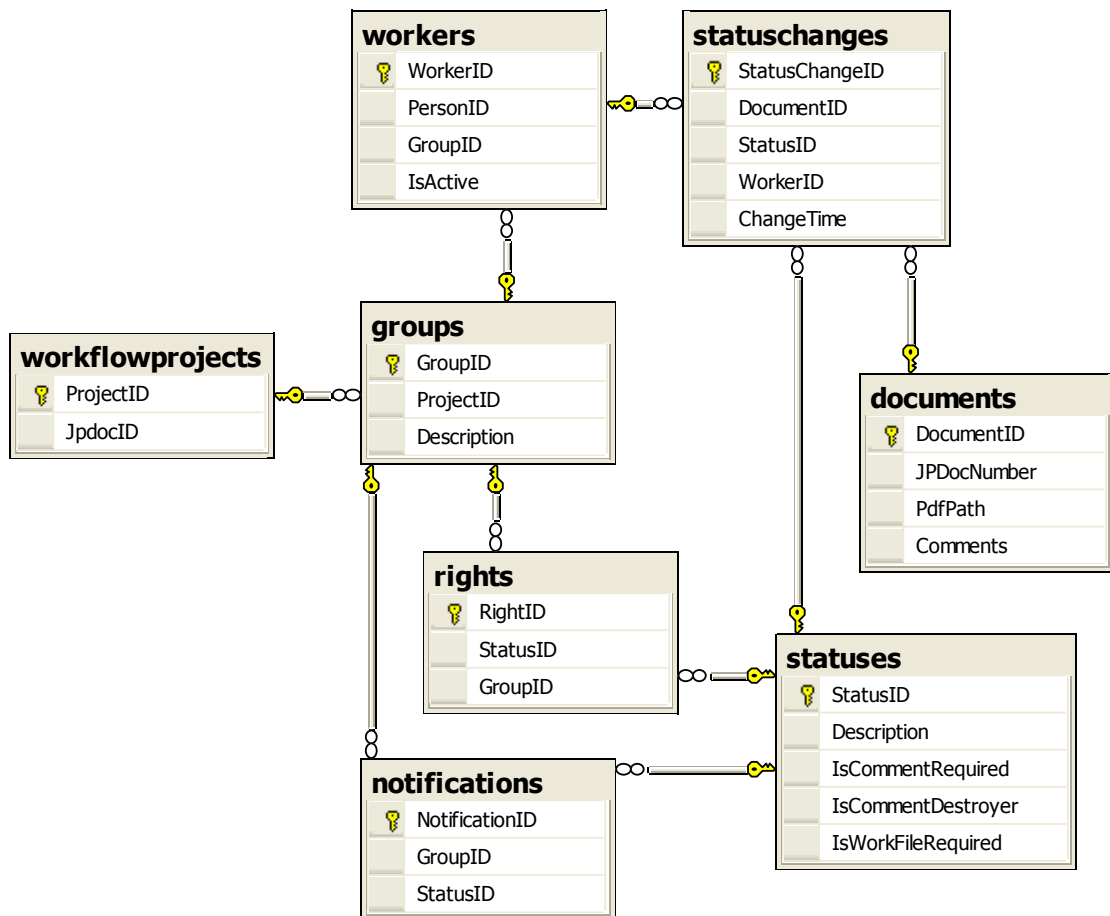
Projektin hallinta aloitetaan projektin asetusten määrittämisellä, josta tehtiin kolmiportainen. Ensimmäisessä vaiheessa luodaan tarvittavat dokumenttitilat, mikäli niitä ei vielä ole olemassa. *Kuvaus* (description) on dokumenttitilan kuvaus, mikä näkyy Flow-sovellusta käytettäessä. Se voi olla esimerkiksi suunniteltu, hyväksytty tai hylätty. Kolme muuta tietoa ovat tiedot, vaaditaanko kommentoitu katselutiedosto, tuhotaanko se, mikäli se on olemassa, ja vaaditaanko käyttäjää osoittamaan alkuperäiset työ- ja katselutiedostot. Nämä viimeiset muuttujat ovat *bittimuuttujia* eli niiden arvo voi olla *epätosi* (0) tai *tosi* (1).

Toisena portaana luodaan jokaiselle projektille erikseen *käyttäjärühmät* (groups). Käyttäjärühmän nimi ei näy kuin Flow-sovelluksen hallintapuolella, joten käytettävillä nimillä ei ole suoranaista merkitystä sovelluksen toimintaan. Selkeyden vuoksi toki yhtenäisen ja selkeän merkintätavan käyttäminen on suotavaa. Kullekin käyttäjärühmälle on mahdollista määritellä dokumenttitilat, jotka vaikuttavat ryhmän

dokumenttinäkymiin. Dokumenttitilakohtaisesti käyttäjäryhmällä voi olla oikeus vaihtaa dokumentin tila kyseessä olevaksi tilaksi, mikä määritellään *oikeudet*-taulussa (rights). Toisena vaihtoehtona on, että sitä informoidaan dokumenttien dokumenttitilan muuttumisesta kyseiseksi tilaksi *ilmoitukset*-taulussa (notifications) olevien määritysten mukaan tai sitten dokumenttitila ei vaikuta kyseiseen ryhmään lainkaan. Molemmissa oikeudet- ja ilmoitukset-tauluissa on pääavaimen lisäksi ainoastaan kaksi *viiteavainta* (foreign key), joilla linkitetään tietty käyttäjäryhmä aina haluttuun dokumenttitilaan. Samaan dokumenttitilaan voi luonnollisesti olla useita linkkejä.

Projektiasetusten viimeisenä kohtana on työntekijöitä lisäys projektiin. *Työntekijät*-taulussa (workers) ovat viiteavaimina *PersonID*, joka on Active Directoryn käyttäjätunnus ja *GroupID*, jolla määritellään, mihin käyttäjäryhmään valittu henkilö kyseisessä projektissa kuuluu. Sama henkilö voi kuulua useaan eri projektiin ja kaikkiin eri käyttäjäryhmän edustajana. Viimeiseksi sarakkeeksi lisättiin vielä bittityyppinen aktiivisuudesta kertova *IsActive*-kenttä, jolla osoitetaan työntekijän oikeuksien voimassaolo. Toisaalta vältetään myös taulun koon kasvaminen, mikäli samaa henkilöä siirrettäisiin useasti projektiin ja siitä pois.

Dokumenttien tilanvaihto on normaalin käyttäjätilan käyttämisen kannalta kaikkein oleellisin. Sinne tallennetaan viiteavaimina tiedot muutettavasta dokumentista, uudesta dokumenttitilasta ja muutoksen tekijästä. Lisäksi tallennetaan muutoksen ajankohta *UTC-ajassa* (Coordinated Universal Time), joka tunnettiin aiemmin GMT-aikana (Greenwich Mean Time). UTC-aika on sijainnista eli tarkemmin aikavyöhykkeistä ja kesä- sekä talviajoista riippumaton, joten sen käyttäminen on perusteltua maailmanlaajuisessa ympäristössä. Tulevaisuudessa Flow-sovelluksen käyttäjäasetuksiin voitaisiin lisätä tieto käyttäjän aikavyöhykkeestä, jolloin kaikki muokkaustiedot näkyisivät nopeammin hahmotettavassa muodossa. Toiminnallisuuden kannalta tällä ei ole kuitenkaan merkitystä. [28.]



Kuva 8. Flow-sovelluksen käyttämä SQL-relaatiotietokanta.

Dokumenttikohtaisesti *dokumentit*-tauluun (documents) tallennetaan JPDocin dokumentin avaintieto eli dokumentti-id. Lisäksi tallennetaan merkkijonomuodossa mahdollisen kommentoidun katselutiedoston sijainti. Tämä arvo on myös sallittu *tyhjäksi* (null) tai se voi olla merkitön merkkijono. Tätä ominaisuutta käytetään tarkistamaan, onko kyseisellä dokumentilla kommentoituja katselutiedostoja vai ei. Viimeisessä sarakkeessa dokumentilla voi olla myöskin merkkijonotyypistä tietoa, joka tällä kertaa kuvastaa tekstimuotoisia kommentteja. Niitä ei alun perin suunniteltu Flow-sovellukseen lainkaan, mutta käytännön kokemukset osoittivat koevaiheessa sen, että kaikkia kommentteja ei kannata merkitä PDF-katselutiedostoihin. Dokumentilla voi siis olla sekä kommentoitu katselutiedosto että tekstikommentteja, jompikumpi tai ei kumpaakaan. Flow-sovelluksta varten luotu tietokanta on esitetty kuvassa 8.

Tietokannan elintärkeän viite-ehyden säilyminen on varmistettu kahdella tapaa. SQL-tietokannassa on mahdollista määritellä tauluihin viiteavaimia, jolloin varmistetaan, että

tietuetta, johon viitataan toisessa taulussa, ei voida poistaa. Toisena varmistuksena on ohjelmakoodiin tehty varmistuksia, joissa sama asia tarkistetaan ennen SQL-kyselyn suorittamista SQL-virheilmoitusten välttämiseksi. Lisäksi ohjelmakoodiin on voitu asettaa myös muita reunaehtoja, joiden toteuttaminen SQL-tietokannan asetuksin ei olisi ollut mahdollista.

3.5 Tiedon tallentaminen

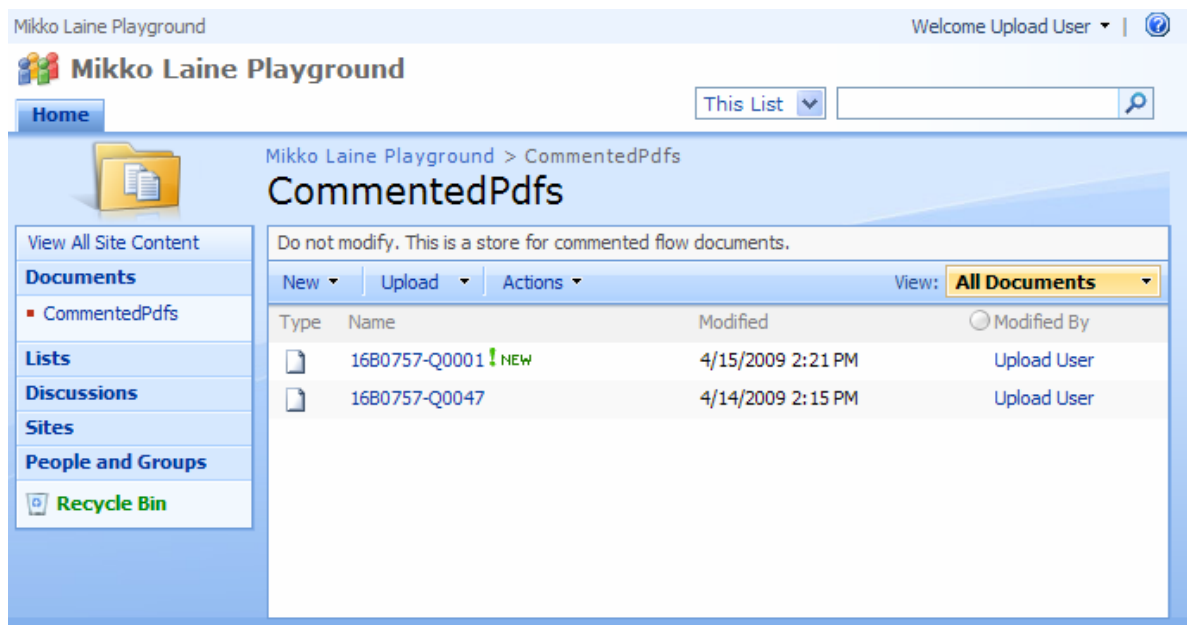
Relaatiotietokannan hallinta

Relaatiotietokannan käsittelyä varten luotiin SqlDataReader-luokka. Microsoftin .NET Framework tarjoaa SQL-tietokantayhteyksiä varten valmiin SqlConnection-luokan [29]. SqlDataReader-luokkaan on sisällytetty tarvittava *yhteysmerkkijono* (Connection String), joka sisältää tietokannan sijaintitiedon palvelimella ja käytettävän tietokannan nimen, sekä virheenkäsittelyt kokonaisuudessaan, jotta muualla sovelluksessa tietokannan käsittely pysyisi mahdollisimman yksinkertaisena. Virheenkäsittely on hoidettu monista ohjelmointikielistä tutulla *try-catch* -toiminnolla, joissa virhetapauksissa tulostetaan käyttäjälle virheilmoitus MessageBox-luokan *Show*-metodia kutsumalla. Parametrina välitetään poikkeuksen virheilmoitus, jotta käyttäjälle muodostuisi jonkinlainen käsitys virheen laadusta. Todellisuudessa virheilmoitus helpottaa sovelluksen kehittäjää paikallistamaan ohjelmakoodista ongelman aiheuttaneen virheen ja käyttäjälle annettava informaatio on toissijaista. Pääasia on, että käyttäjä havaitsee ohjelmassa tapahtuneen virheen ja käsittää, mikä työvaihe jäi kesken tai puutteelliseksi.

Aseta-, päivitä- ja poista-komennoista palautetaan metodin kutsukohtaan Boolean-muuttuja, jotta tiedetään, ovatko halutut muutokset onnistuneet ja voidaanko siten toimintoketjussa, jossa tulevat toiminnot ovat edellisistä riippuvaisia, siirtyä eteenpäin. Valitse-komennoista palautetaan SqlDataAdapter-olion avulla monikäyttöinen DataSet-olio, joka sisältää kaiken tietokannasta kysytyn tiedon sisältäen taulut, sarakkeet, rivit ja niiden väliset suhteet. DataSet-olio voi myös sisältää monen kyselyn tuloksen, sitä voidaan käyttää ja muokata vaikka yhteyttä tietokantaa ei olisi, se voidaan luoda XML-dokumentista ja päinvastoin. [30.]

SharePointin dokumenttikirjaston hallinta

SharePoint-ympäristössä on välttämätöntä tunnistaa käyttäjä, sillä ilman riittäviä oikeuksia sivun sisällön listaaminen ja muokkaaminen on mahdotonta [31]. Jotta jokaista käyttäjää ei tarvitsisi erikseen lisätä vain tiedostojen väliaikaista tallennusta varten, päätettiin käyttää yhtä, ainoastaan Flow-sovelluksen käytössä olevaa käyttäjätunnusta. Käytännössä käyttäjällä ei pitäisi olla lainkaan tarvetta päästä käsiksi kommentoituihin dokumenttien katselutiedostoihin Flow-sovelluksen ulkopuolelta.



Kuva 9. Näkymä SharePoint-sivun dokumenttikirjastosta.

Kommentoitudut katselutiedostot eli PDF-tiedostot sijaitsevat SharePointin dokumenttikirjastossa, joka on nimeltään *CommentedPdfs* (kuva 9). Kommentoitujen dokumenttien tallennusta ja lataamista varten luotiin SharePointReader-luokka, jonka oleellisiin toimintoihin kuuluvat *uploadFile*- ja *downloadFile*-menetelmät. Microsoftin .NET Framework tarjoaa *WebRequest*-luokan, jonka ilmentymälle voidaan antaa käyttäjätunnistukseen tarvittavat tiedot ja ladata tiedosto palvelimelle käyttämällä *put*-menetelmää [12, s. 1070–1073]. Vastaavasti dokumentti voidaan ladata ilman käyttäjätunnistusta hyödyntämällä *WebClient*-luokan *DownloadFile*-menetelmää [12, s. 1068]. SharePointReader-luokalla on muuttujina tiedossa SharePoint-palvelimen sijainti, dokumenttikirjaston nimi ja sen uniikki lista-id, jolla kukin lista on yksilöity

palvelimella. Vastaavasti myös listoissa ja dokumenttikirjastoissa, jotka ovat todellisuudessa vain listojen ilmentymiä, dokumenteilla on automaattisesti juokseva numerointi eli dokumentti-id [31].

3.6 Käyttöliittymän ja ohjelmistorakenteen suunnittelu

Visual Studiossa käyttöliittymäsuunnittelu on nimensä mukaisesti visuaalista ja helposti lähestyttävää toimintaa, sillä komponenttien sijoittaminen sovelluksen työalueelle onnistuu pitkälti käyttämällä hiiren *vedä ja pudota* -toimintoa (drag and drop). Kehitysympäristö luo komponenttien alustamiseen tarvittavat ohjelmakoodin lauseet, joten sovelluksen suunnittelija pystyy keskittymään toimintojen rakentamiseen. Näin myös sovelluksen jakaantuu loogisesti kahteen osa-alueeseen, mikä mahdollistaa sovelluksen rungon hyödyntämisen myös muissa projekteissa tai toisessa ympäristössä hyvin pienin muutoksin.

Ohjelman tai sen osien siirrettävyys on erittäin tärkeää, sillä nykypäivän on varauduttava esimerkiksi käyttöjärjestelmien vaihtumiseen, jolloin graafisen käyttöliittymän luonti voidaan joutua muuttamaan kokonaisuudessaan. Siirrettävyyden etuja ovat myytävän sovelluksen markkinoiden laajentuminen, vankkarakenteisuus, vapaus käytettävän järjestelmälustan valinnassa, mukautuminen tekniikan muutoksiin, kuten siirtyminen 32-bittisestä ympäristöstä 64-bittiseen, ja ylipäätänsä uusien mahdollisuuksien aukeaminen. [32, s. 2.]

Käyttöliittymäsuunnittelun pohjaksi otettiin käyttäjien opettelukynnyksen pysyminen matalana, joten mukauduttiin käyttämään nykyisen dokumentinhallintajärjestelmän rakennetta hieman yksinkertaistetuin toiminnoin. Näin jäi loogisesti tilaa myös uudistetuille toiminnoille ilman, että käyttäjä hukkuisi vaihtoehtojen määrään.

4 Flow-sovelluksen ominaisuudet ja toiminta

4.1 Ohjelman ominaisuudet

Flow on paikallinen Windows-sovellus, joka hakee tietoa palvelimilla sijaitsevista resursseista ja tallentaa tietoa palvelimille, mikä tarkoittaa tietovarastojen olevan ohjelman kaikille käyttäjille yhteisiä. Nykyisten monipuolisten ajoympäristöjen mahdollistaman Flow-sovelluksen kaikki raskas toiminta on jaettu eri säikeisiin pääohjelman kanssa, jotta käyttöliittymäkomponenttien toiminta ei jumiutuisi ja näin käyttäjä kyetään pitämällä informaatiopalkilla tietoisena kullakin hetkellä tapahtumista toiminnoista. Sovelluksen toiminta on jaettu kahteen eri osa-alueeseen: hallinta- ja käyttäjäpuoleen.

Hallintapuolen yleispiirteet

Vain rajatulla joukolla työntekijöitä on oikeus kirjautua sisään hallintapuolelle. Kirjautuminen on hieman harhaanjohtava ilmaisu, sillä oikeudettomilla työntekijöillä hallintapuolen valintavaihtoehto ei näy lainkaan sovelluksen valikossa. Huonona puolena sallitut käyttäjätunnukset on koekäyttöä varten kirjoitettu suoraan osaksi ohjelmakoodia, joten hallintaoikeuksien muuttaminen vaatii aina uuden sovelluksen kääntämisen. Tämän osa-alueen kehitys jätettiin odottamaan päätöstä siitä, mitä kautta kyseistä toimintoa halutaan hallinnoida.

Hallintapuolella perustoimintona on projektien lisääminen *työnkulkuprojekteihin* (Flow projects), mikä on edellytyksenä, jotta muita asetuksia voidaan muokata. Kun projekti on työnkulun kontrollissa, lisätään kaikki mahdolliset dokumentin tilat, mikäli niitä ei ole muissa projekteissa käytetty aikaisemmin eli dokumenttitilat ovat yhteisiä kaikille työnkulkuprojekteille. Tämän jälkeen luodaan käyttäjäryhmät, kuten suunnittelija ja tarkastaja, sekä määritellään ryhmille oikeudet tiettyihin dokumenttitiloihin. Viimeisenä vaiheena on henkilöiden lisääminen projektiin ja näiden projektihenkilöiden lisääminen työryhmiin (kuva 3, s. 18).

Käyttäjäpuolen yleispiirteet

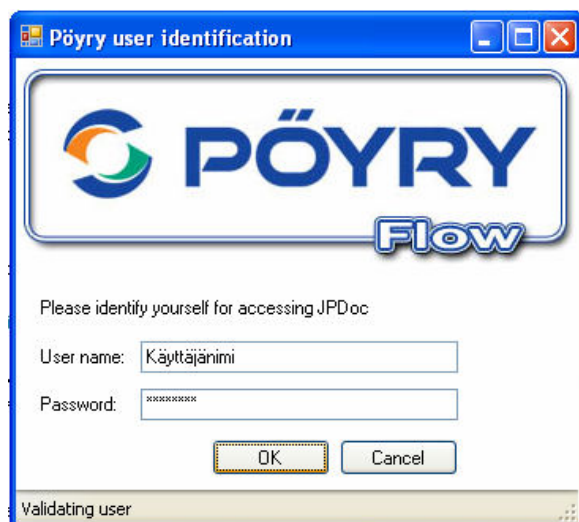
Päätoimintona käyttäjäpuolella on mahdollista selata projekteja ja vaihtaa dokumenttien tiloja hallintapuolella määriteltyjen oikeuksien puitteissa (kuva 4, s. 19). Aluksi käyttäjäpuolella valitaan projekti ja halutut suodatusvaihtoehdot dokumenttilistauksen rajaamiseksi. Hakuvalintojen perusteella tehty haku aktivoituu käyttäjän painettua *Search / Refresh* -näppäintä.

Käyttäjäpuolen projektityötila on jaettu kolmeen välilehteen. Ensimmäinen välilehti on nimeltään *My Group's Documents*, jossa näkyvät kaikki kirjautuneen käyttäjän työryhmää koskevat dokumentit. Tällä välilehdellä tehdään dokumenttien kommentointi ja tilanvaihdokset. Tapahtuma on kuvattu kokonaisuudessaan liitteessä 5.

Toinen välilehti on *All Flow Documents*, jossa näkyvät kaikki valitun projektin työnkulkudokumentit. Tässä tilassa näkyvät myös edellisellä välilehdellä näkyneet dokumentit. Viimeinen välilehti *All Documents* on tarkoitettu vain metatietojen katseluun ja listaksi mahdollisista dokumenteista lisättäväksi työnkulkudokumentteihin.

4.2 Tilojen yhteiset toiminnot

Kirjautuminen järjestelmään

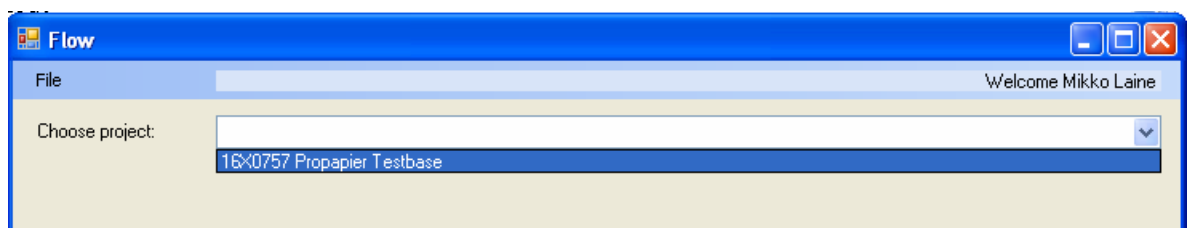


Kuva 10. Käyttäjän kirjautumisruutu ohjelman käynnistyessä.

Flow-sovelluksen ulkopuolisten tietojen hakeminen vaatii käyttäjän tunnistuksen, joka hoidetaan erillisillä kirjautumisikkunalla ennen varsinaisen sovelluksen käynnistymistä (kuva 10). Pöyryn kaikilla työntekijöillä on lukuoikeus Active Directoryyn, jossa on tiedot kaikista käyttäjistä. Vastaava oikeus on kaikilla työntekijöillä myös Lotus Domino -palvelimelle. Käyttäjätunnistuksessa olisi mahdollista hyödyntää automaattisesti käyttäjän kirjautumistunnusta Windowsiin, mutta koeversiovaiheessa haluttiin mahdollisuus kirjautua eri käyttäjänä Flow-sovellukseen ja näin säästyttiin myös joissakin tapauksissa erittäin monimutkaisten luottamusasioiden käsittelyltä. Toisaalta kirjautuminen Lotus Domino -palvelimelle on elinehto, jotta kyetään hyödyntämään Pöyryn dokumentinhallintajärjestelmä JPDociin tehtyä rajapintaa. [7.]

Kirjautuminen siis tapahtuu todellisuudessa Lotus Domino -palvelimelle, jolloin käyttäjätunnus on muotoa *etunimi sukunimi*. Domino-palvelimelle on kuitenkin käyttäjätietoihin määritelty myös AD:n ja Windowsin käyttäjätunnus useimmille käyttäjille, joten kirjautuminen onnistuu millä tahansa käyttäjälle kirjatuille kirjautumistiedoilla, joten tapahtuma on mahdollisimman joustava.

Projektin valinta



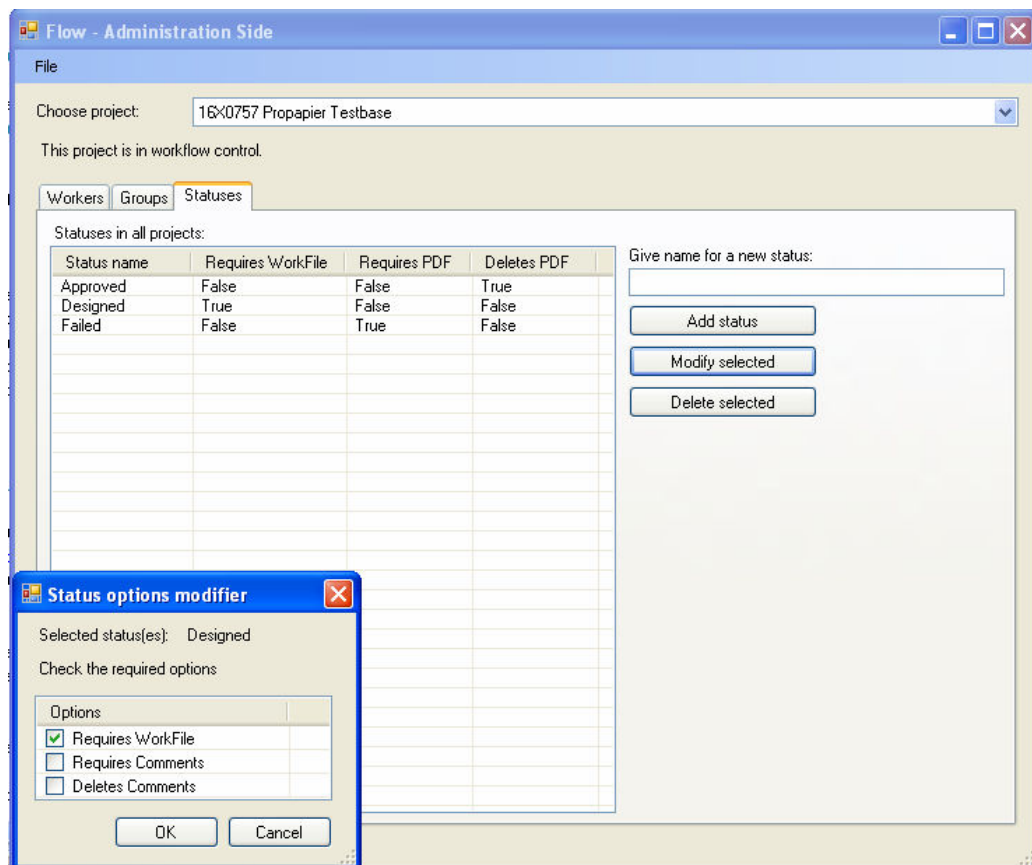
Kuva 11. Projektin valinnan alasvetovalikko käyttäjä- ja hallintatilassa.

Flow-sovelluksessa projektalista on nähtävillä alasvetovalikossa (combo box), josta haluttu projekti voidaan valita. Alasvetovalikon tiedot muodostuvat projektikoodista, joka on käytössä dokumentinhallintajärjestelmä JPDocissa ja projektin vapaamuotoisesta kuvauksesta. Projektalista muodostetaan jäsentämällä HTML-sivun lähdekoodi, joka luodaan dynaamisesti JPDocin rajapinnan palvelua kutsumalla. Kuvassa 11 ainoastaan yksi on projekti määritelty Flow-projektiksi.

4.3 Hallintatila

Dokumenttitilojen luominen ja muokkaaminen

Dokumenttitilojen luominen onnistuu kirjoittamalla tekstikenttään halutun dokumenttitilan nimi ja painamalla *Add status* -painiketta. Dokumenttitila lisätään vaihtoehtoihin, ellei sen nimestä ole jo olemassa. Dokumenttitilojen poistaminen tapahtuu valitsemalla yksi tai useampi dokumenttitila hiiren vasemmalla näppäimellä ja painamalla *Delete selected* -painiketta. Dokumenttitiloja on mahdollista poistaa ainoastaan, mikäli yhdelläkään dokumentilla ei ole ollut tilana kyseessä olevaa dokumenttitilaa.



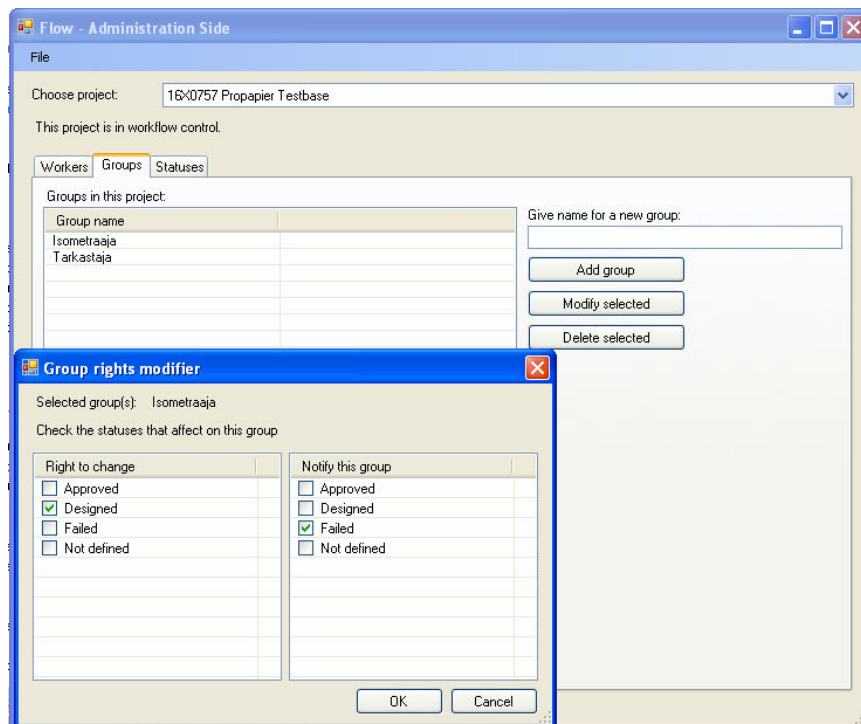
Kuva 12. Dokumenttitilojen muokkaaminen tapahtuu hallintatilassa.

Dokumenttitilojen muokaus tapahtuu valitsemalla yksi tai useampi dokumenttitila hiiren vasemmalla näppäimellä ja painamalla *Modify selected* -painiketta. Aukevasta dokumenttitilan ominaisuuksienmuokkausikkunasta on mahdollista muuttaa

kolmenlaisia tila-asetuksia (kuva 12). Mikäli dokumenttitila vaatii työtiedoston, kysytään käyttäjältä aina työ- ja katselutiedoston sijainti, kun hän vaihtaa jonkin dokumentin tilan valituksi dokumenttitilaksi.

Jos dokumenttitila vaatii kommentteja, näytetään käyttäjälle erillisellä ikkunalla kommentointivaihtoehdot, jotka sisältävät tekstikommenttien kirjoittamisen, kommentoidun katselutiedoston osoittamisen tai molemmat. Valitsemalla viimeinen *kommenttien poisto* -kohta, sovellus poistaa dokumenttiin liitetyt kommentit ja kommentoidun katselutiedoston viitteen tietokannasta, kun käyttäjä valitsee kyseisen dokumenttitilan.

Työryhminen luominen ja muokkaaminen

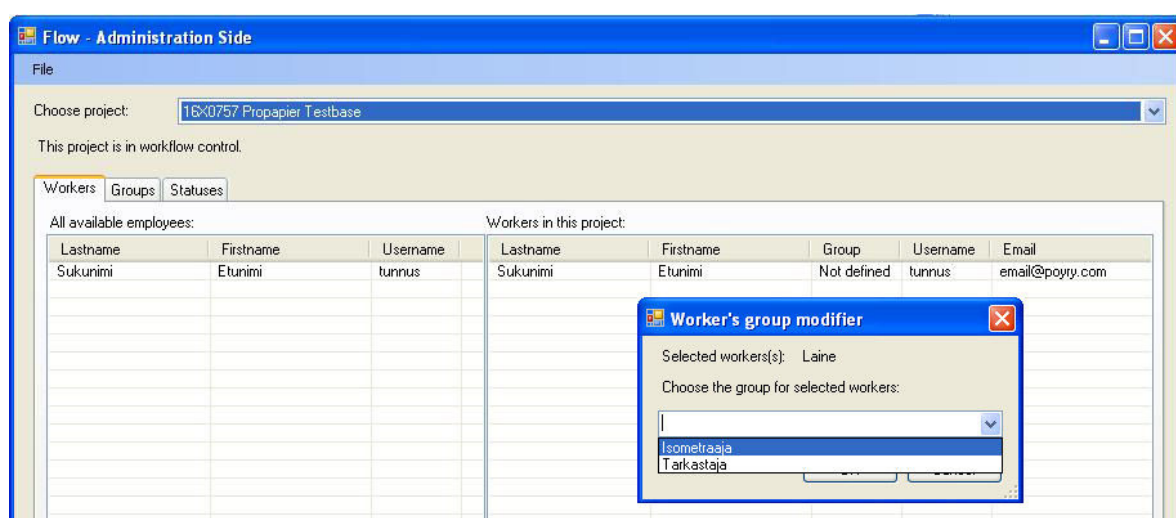


Kuva 13. Työryhmien oikeuksien ja ilmoitettavien dokumenttitilojen muokkaaminen hallintatilassa.

Työryhmien luominen tapahtuu kirjoittamalla tekstikenttään uuden työryhmän nimi ja painamalla *Add group* -painiketta, jolloin ryhmä lisätään kyseiseen projektiin, ellei sen nimistä työryhmää ole jo olemassa. Työryhmien nimet eivät näy Flow-sovelluksen käyttäjäpuolella, joten ne voidaan nimetä täysin vapaasti riippumasta todellisesta

merkityksestä tai kielestä. Kun yhden tai useamman työryhmän oikeuksia halutaan muuttaa, valitaan ryhmät ja painetaan *Modify selected* -painiketta, jolloin oikeuksienmuokkausikkuna aukeaa (kuva 13). Jotta projektin dokumentinhallinta ei vaarannu, on työryhmän poistaminen mahdollista ainoastaan, mikäli siihen ei kuulu työntekijöitä. Oletusarvoisesti määritellään projektiasetukset hallintatilassa ennen projektin toiminnan aloittamista, mutta on kuitenkin varauduttava kesken projektin ilmentyviin muutostarpeisiin.

Työntekijöiden sijoittaminen projektiin ja työryhmiin



Kuva 14. Esimerkki työntekijöiden sijoittamisesta projektiin ja työryhmiin.

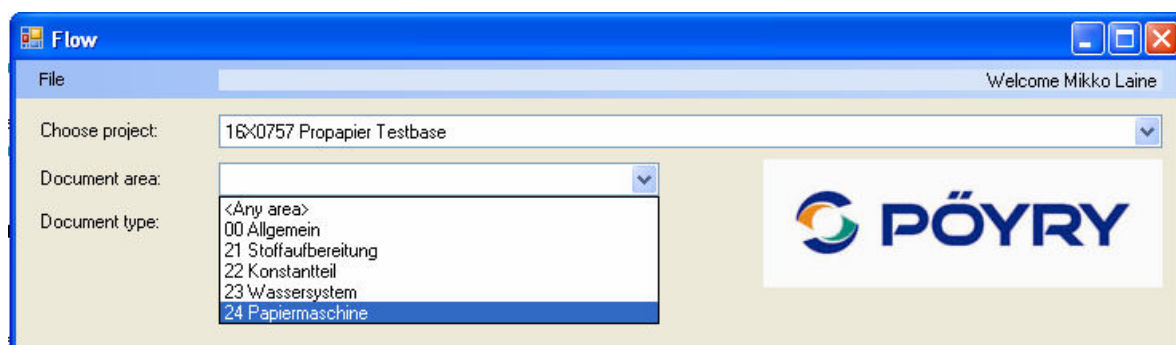
Työntekijöiden sijoittaminen työryhmiin tapahtuu joko vasemmalla sijaitsevan työntekijälistauksen puolella painalla hiiren oikean näppäimen puolelta *Add to project* -vaihtoehto tai hiiren tuplapainalluksella. Vasemman ja oikean puoleisissa ikkunoissa toimii yhden tai useamman henkilön valinta pitämällä CTRL-näppäintä pohjassa.

Oikean puoleisen ikkunan ruudusta on mahdollista hiiren tuplapainalluksella poistaa valittu henkilö projektista. Hiiren oikean näppäimen takaa aukeavasta valikosta mahdollista vaihtaa henkilön työryhmää, lähettää työntekijälle sähköpostia tai poistaa työntekijä projektista (kuva 14). Todellisuudessa työntekijää ei poisteta projektista vaan hänen mukanaolonsa projektissa poistetaan.

4.4 Käyttäjätila

Dokumenttistauksen rajaaminen

Koska projekteissa voi olla periaatteessa rajaton määrä ja käytännössä useita tuhansia dokumentteja, on dokumenttistausta syytä rajata, jotta tietojen hakemiseen kuluva aika pysyy käyttäjän kannalta siedettävissä rajoissa. Jokaisella dokumentilla on määritelty dokumenttityyppi ja -alue, joihin se kuuluu. Projektin alueet voidaan esimerkiksi jakaa projektissa eri henkilöille, tai tietty työryhmä tekee ainoastaan tiettyntyyppisiä dokumentteja.



Kuva 15. Dokumenttistauksen rajaaminen onnistuu valitsemalla dokumentin alue- ja tyyppisuodattimia.

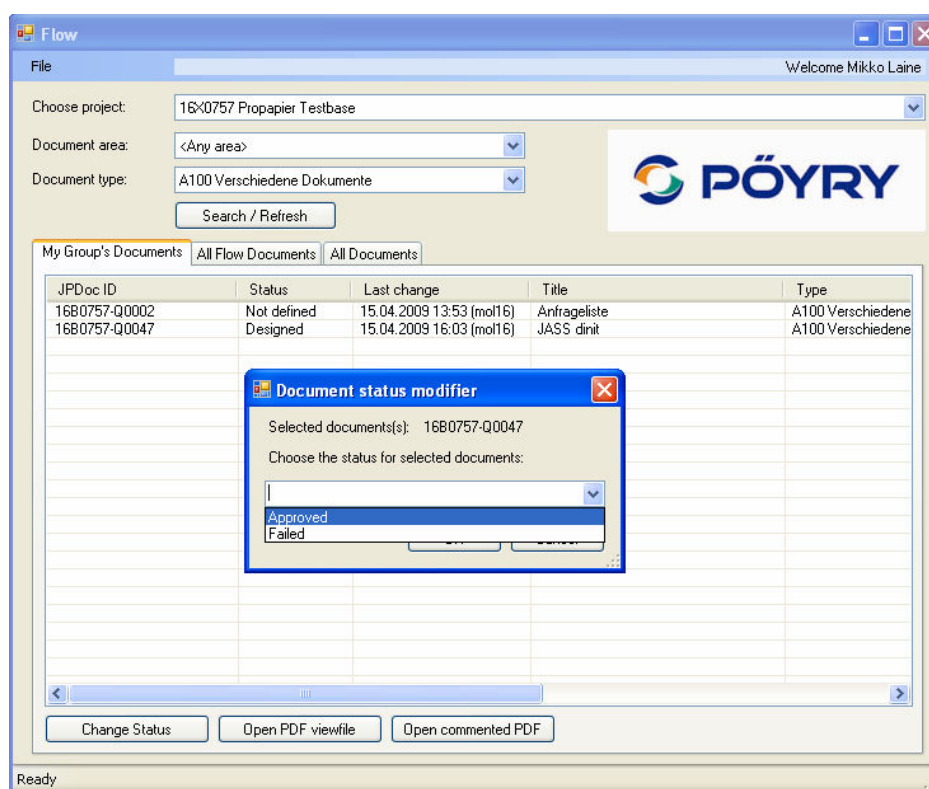
Käyttäjät voivat valita alasvetovalikosta haluamansa suodatusvaihtoehdot, joiden käyttäminen nopeuttaa tietojen hakemista merkittävästi (kuva 15). Mikäli käyttäjä ei aseta hakuehtoja lainkaan, kysyy sovellus käyttäjältä vahvistusta ennen haun suorittamista ja varoittaa mahdollisesti pitkään kestävästä odotusajasta.

Uusien työnkulkudokumenttien lisääminen

All documents -välilehdellä käyttäjällä on mahdollisuus lisätä hiiren oikean näppäimen painalluksen jälkeen aukeavan valikon kautta valittu dokumentti tai valitut dokumentit työnkulku-dokumentteihin, mikä tarkoittaa, että niiden tila tallennetaan Flow-sovelluksen tietokantaan. *All documents* -välilehdellä näkyvät dokumentit ja niiden

tiedot ovat peräisin JPDocista. Flow-sovelluksen suunnitteluvaiheessa rajattiin sen toiminta siten, että uusia dokumentteja ei voida luoda työkalun avulla JPDociin, vaan ainoastaan siellä jo olevia dokumentteja voidaan ottaa työnkiertoon. Kun dokumentti siirretään työnkiertoon, tulee sen tilaksi *määrittelemätön* (Not defined).

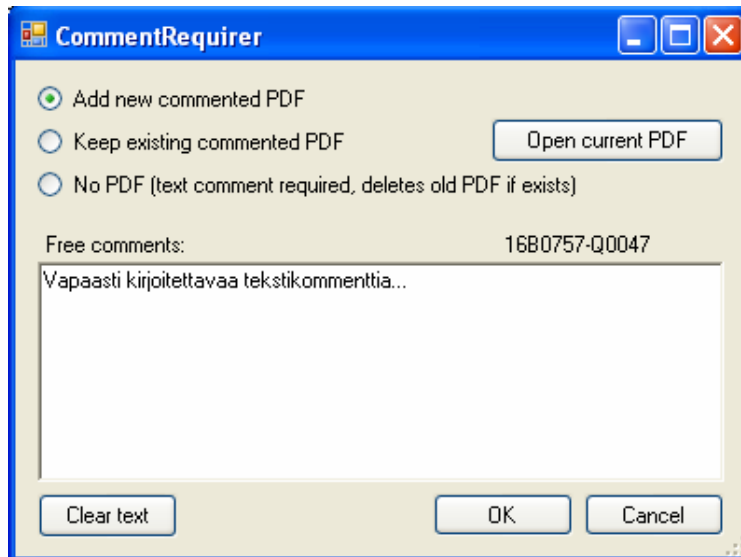
Työnkulkudokumenttien tilan vaihtaminen



Kuva 16. Dokumentin tilaa vaihtaessa, näyttää sovellus ainoastaan hallintatilassa kyseisen käyttäjän edustamalle työryhmälle määritellyt tilavaihtoehdot.

Kirjautuneelle käyttäjälle näkyy *My Group's Documents* -välilehdellä ainoastaan dokumentit, jotka koskevat hänen edustamaansa työryhmää. Näiden dokumenttien tilaa voidaan vaihtaa hiiren oikean näppäimen painalluksesta aukeavasta valikosta. Valinnan jälkeen ruudulle aukeaa uutta dokumenttitilaa tiedusteleva ikkuna (kuva 16). Toinen vaihtoehto tilanvaihtamiselle on *All Flow Documents* -välilehdeltä tilanmuutoksen peruminen, jolloin dokumentin tila muuttuu määrittelemättömäksi. Tätä toimintaa käytetään dokumenttien uudistamisessa ja päivittämisessä riippumatta edellisen version sijainnista työnkulussa.

Työnkulkudokumenttien kommentointi

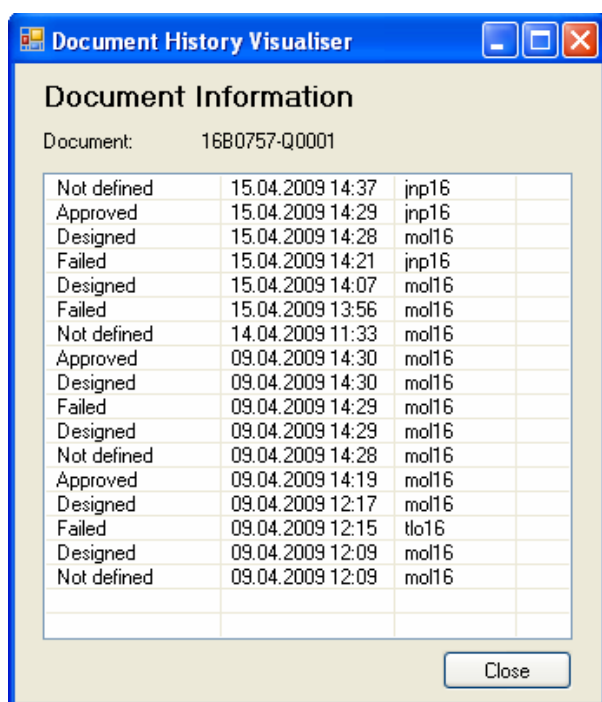


Kuva 17. Kommenttien lisääminen dokumenttiin onnistuu joko tekstimuotoisena tai PDF-muodossa.

Käytännössä vain dokumentin tarkastajille aukeaa kuvassa 17 näkyvä ikkuna, jossa vaaditaan kommentoimaan valittua dokumenttia. Todellisuudessa ikkuna aukeaa tilanvaihdon vahvistamisen jälkeen, mikäli valitulle dokumenttille on määritelty *vaatii kommentteja* -ominaisuus. Tarkastajalle on annettu mahdollisuus jättää edellinen kommentoitu katselutiedosto dokumentin liitteeksi, mikäli esimerkiksi edellisiä korjauksia ei toteutettu asianmukaisesti. Mikäli edellistä kommentoitua katselutiedostoa ei ole olemassa, on vaihtoehto harmaalla lukittuna, eikä sitä voi valita.

Aina auki olevina vaihtoehtoina ovat uuden kommentoidun katselutiedoston lisääminen, jolloin käyttäjältä kysytään sen sijaintia ja tekstikommenttien täyttäminen. Mikäli tekstikenttä on tyhjä, poistetaan tietokannan tekstikommentit valitun dokumentin kohdalta. Muuten ne korvataan tekstikentän sisällöllä. Tarkastajalla on mahdollisuus kirjoittaa tekstikommentteja myös lisäksi kommentoidulle katselutiedostolle.

Työnkulkudokumenttien historia



The screenshot shows a window titled 'Document History Visualiser' with a 'Document Information' section. The document ID is '16B0757-Q0001'. Below this is a table with columns for status, date and time, and user name. The table contains 17 rows of data, with the most recent status being 'Not defined' at 15.04.2009 14:37 by user 'jnp16'. A 'Close' button is located at the bottom right of the window.

Status	Date and Time	User
Not defined	15.04.2009 14:37	jnp16
Approved	15.04.2009 14:29	jnp16
Designed	15.04.2009 14:28	mol16
Failed	15.04.2009 14:21	jnp16
Designed	15.04.2009 14:07	mol16
Failed	15.04.2009 13:56	mol16
Not defined	14.04.2009 11:33	mol16
Approved	09.04.2009 14:30	mol16
Designed	09.04.2009 14:30	mol16
Failed	09.04.2009 14:29	mol16
Designed	09.04.2009 14:29	mol16
Not defined	09.04.2009 14:28	mol16
Approved	09.04.2009 14:19	mol16
Designed	09.04.2009 12:17	mol16
Failed	09.04.2009 12:15	llo16
Designed	09.04.2009 12:09	mol16
Not defined	09.04.2009 12:09	mol16

Kuva 18. Dokumentin tilanvaihtohistoriaa voi tarkkailla näyttöikkunassa, joka aukeaa valitsemalla "View Document History" valikosta, joka aukeaa käyttäjän painettua dokumentin kohdalla hiiren oikeata näppäintä.

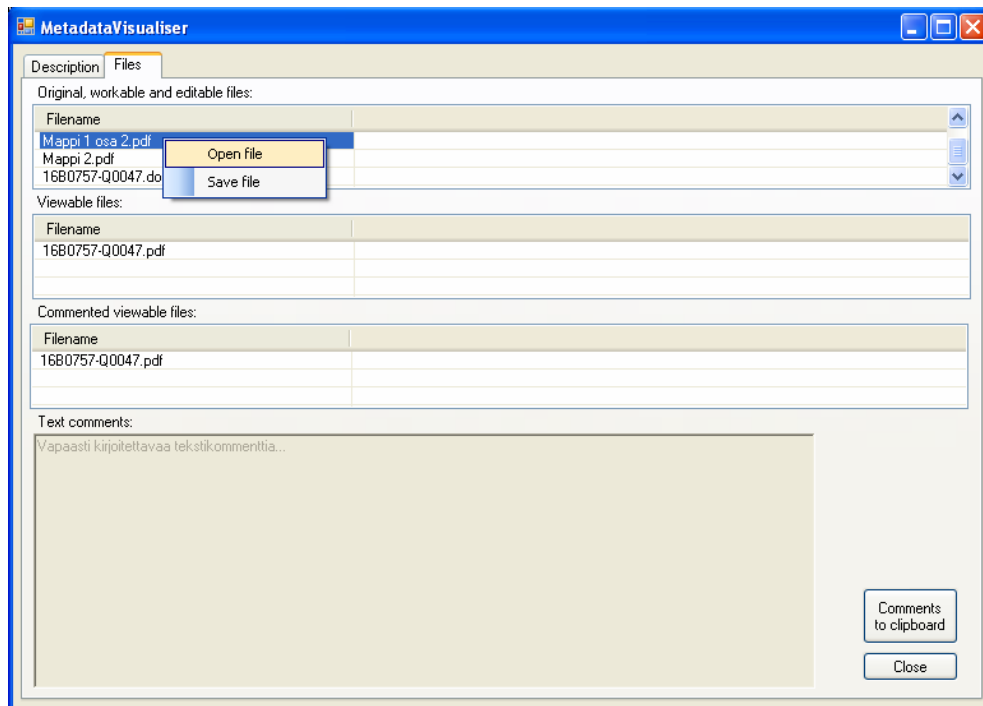
Työnkulkudokumenttien historiaikkuna tarjoaa näkymän dokumentin tilanvaihtohistoriaan. Tilanvaihdot ovat aikajärjestyksessä siten, että viimeisin tilanvaihto on ylimpänä. Tiedoista on näkyvissä vaihdettu dokumenttitila, tilanvaihdon päivämäärä ja kellonaika sekä tilanvaihdon tekijän käyttäjätunnus (kuva 18).

Metatiedon katselu ja muokkaus

The screenshot shows the 'MetadataVisualiser' application window. It has a blue title bar and a menu bar with 'Description' and 'Files' tabs. The main area is a form with various input fields and labels. The 'Project' field is set to '16X0757 Propapier Testbase'. The 'Title' field contains 'JASS dinit'. The 'Iss company' is 'JPH'. The 'Author' is 'Kalle Rasimäki'. The 'Phase' is '0 Registriert'. The 'Area' is '00 Allgemein'. The 'Class' is empty. The 'Dco type' is 'A100 Verschiedene Dokumente'. The 'Position' is empty. The 'Scale' and 'Size' fields are empty. The 'Add info' field is empty. The 'Jp doc no' is '16B0757-Q0047'. The 'Cl docno' is empty. The 'Info 1' through 'Info 4' fields are empty. The 'webpub' field is empty. There are 'Edit metadata' and 'Close' buttons at the bottom right.

Kuva 19. Metatiedon näyttöikkuna on hyvin paljon esikuvansa Document Explorerin vastaavan toiminnon kaltainen. Metatiedot on merkitty harmaalla, millä korostetaan niiden koskemattomuutta.

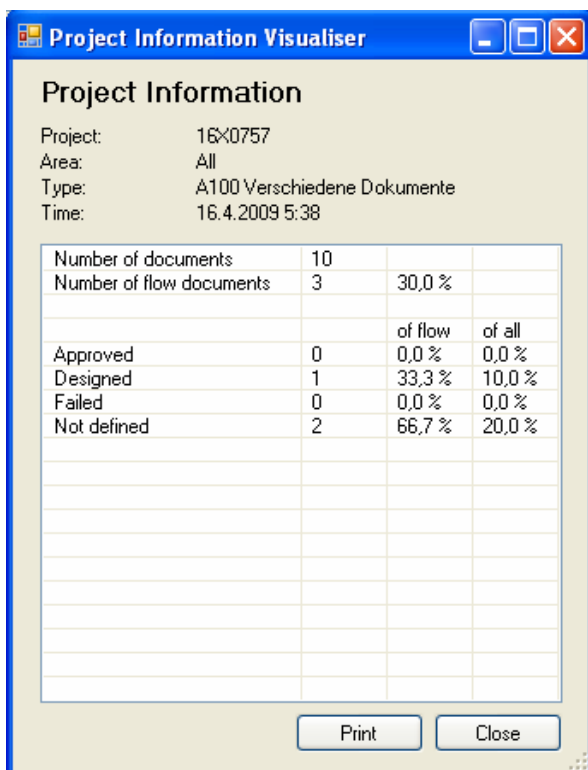
Jokaisesta käyttäjätilan ikkunasta on mahdollista siirtyä hiiren kaksoispainalluksella tai oikean näppäimen painalluksen jälkeen aukeavan valikon kautta katselemaan valitun dokumentin metatietoja (kuva 19). Teknisesti ei ole estettä usean dokumentin metatiedon avaamiseksi samalla kertaa, mutta käytettävyyden kannalta usean dokumentin valinta on poistettu käytöstä. Metatiedot sisältävät kaiken dokumenttiin liitetyn informaation tiedostojen lukuun ottamatta, kuten dokumentin tunnuksen, kuvauksen, muokkauspäivän ja tilan.



Kuva 20. Metatiedon näyttöikkunan *Files*-välilehdellä näkyy kaikki dokumenttiin liitetyt tiedostot ja tekstikommentit.

Metatiedon katselusta tehtiin lähes identtinen Document Explorerin vastaavan katselutilan kanssa. Ainoana merkittävänä eroavaisuutena on se, että metatiedon muokkaaminen on mahdollista ainoastaan Document Explorerin kautta. Tätä toimintoa varten Flow-sovellukseen tehtiin erillinen painonäppäin valitun dokumentin avaamiseksi Document Explorer -käyttöliittymässä. Metatiedon katselutilassa on myös toinen välilehti, nimeltään *Files*, josta voidaan tarkastella, avata ja tallentaa dokumenttiin liitettyjä tiedostoja (kuva 20). Lisäksi nähtävillä on dokumenttiin mahdollisesti liitetyt tekstipohjaiset kommentit. Document Explorerissa on nähtävillä ainoastaan työ- ja katselutiedostot, mutta Flow-sovelluksessa on nähtävillä näiden lisäksi myös tekstikommentit ja kommentoidut PDF-tiedostot.

Projektitietojen raportointi



The screenshot shows a window titled "Project Information Visualiser" with a blue title bar. The main content area is titled "Project Information" and contains the following text:

Project: 16X0757
 Area: All
 Type: A100 Verschiedene Dokumente
 Time: 16.4.2009 5:38

Below this text is a table with the following data:

Number of documents	10		
Number of flow documents	3	30,0 %	
		of flow	of all
Approved	0	0,0 %	0,0 %
Designed	1	33,3 %	10,0 %
Failed	0	0,0 %	0,0 %
Not defined	2	66,7 %	20,0 %

At the bottom of the window are two buttons: "Print" and "Close".

Kuva 21. Projektitietojen raportti-ikkuna on tarkoitettu projektin hallinnosta vastaaville henkilöille.

Kustakin projektista on saatavilla raportti kahdella eri tavalla, joista toinen on ajateltu karkeasti suunnittelutyötä tekeville ja toinen johtavissa tehtävissä toimiville.

Käyttäjätilassa *All flow documents* -välilehdeltä voidaan avata erilliseen ikkunaan numeerinen yhteenveto projektin kaikista dokumenteista ja niiden tiloista (kuva 21).

Tämä ilmaisumuoto on erittäin hyödyllinen johtotehtävissä oleville työntekijöille, jotta he voivat nopeasti saada yleiskuvan projektin etenemisestä. Siinä verrataan halutut hakuehdot täyttävien työnkulkudokumenttien määrän suhdetta koko dokumenttien määrään ja vastaavasti eri dokumenttitilojen osuuksia molemmista edellä mainituista kokonaismääristä.

5 Koevaihe

5.1 Koevaiheen kuvaus

Sovelluskehityksessä ensimmäinen koevaihe on toiminnallisuuden varmistaminen kehitysympäristössä sovelluksen kehittäjän toimesta. Luonnollisesti kehittäjä pyrkii mielikuvituksensa rajoissa simuloimaan todellisten käyttäjien toimintaa sovellysympäristössä, yleensä siinä kuitenkaan onnistumatta. Myös Flow-ohjelman koevaiheen ensimmäinen porras oli yhden henkilön koekäyttö.

Alkuvaiheessa tietoresurssit olivat paikallisia XML-tiedostoja ja tietokannan paikalliskopioita, jotta sovelluksen suorittamat prosessit olisivat mahdollisimman toimintavarmoja. Ensimmäinen havaittu merkittävä ongelma oli riittävän virheentarkistuksen puute useasta kohdasta, sillä sovelluskehityksessä on otettava huomioon yhteyden katoaminen tietolähteeseen missä prosessin vaiheessa tahansa. Lisäksi suuressa projektissa voi tapahtua muutoksia käytettäviin tietoresursseihin erittäinkin pienessä ajassa.

Korjaustoimenpiteiden jälkeen kyettiin muuttamaan tietoresurssit yksi kerrallaan paikallisista todellisiksi. Varsin pikaisella aikataululla yhteydet saatiin toimimaan ja dokumenttitilojen muokkaaminen oli mahdollista siten, että myös JPDoc-dokumentinhallintajärjestelmä havaitsi sille tarkoitetut muutokset ilman virheilmoituksia. Toiminnallisuus, joka jätettiin tarkoituksella keskeneräiseksi, on tiedostojen poistaminen ohjelmallisesti SharePoint-dokumenttikirjastosta. Tiedostojen poistaminen ei toistaiseksi ole toiminnassa, mutta viittaukset niihin poistetaan SQL-tietokannasta, joten Flow-sovellus ei noteeraa ylimääräisiä kommentoituja katselutiedostoja, jotka sijaitsevat dokumenttikirjastossa. Tämän toiminnallisuuden toteuttamista ei pidetty välttämättömänä sovelluksen koekäyttämiseksi.

Seuraavassa vaiheessa Flow-sovellusta käytettiin kahden käyttäjän välillä. Sovelluksen päätarkoituksellisessa toiminnassa ei havaittu välittömästi ongelmia, mutta *metatietojen muokkaus* -nappia painamalla sovellus jumiutui. Syyksi selvisi käyttäjän kirjautuminen Windows-käyttöjärjestelmään ylläpitäjän tunnuksella, joten Document Explorerin

käynnistys ei Flow-sovelluksesta onnistunut. Tämä tarkoitti myös, että ohjelma olisi jumiutunut työ- ja katselutiedostoja päivitettäessä eli aina tapahtumissa, jotka olisivat pyrkineet käyttämään Document Explorerin COM-objektia.

Havaittujen virheiden korjaamisen jälkeen sovelluksen toiminnassa ei havaittu virheitä kahdessa eri testitilaisuudessa, joista jälkimmäinen oli sovelluksen toiminnan esitystilaisuus. Tietokoneeseen, johon on asennettu Flow-sovellus, on toistaiseksi mahdollista ottaa etäyhteydellä yhteys mistä tahansa riippumatta maantieteellisestä sijainnista ja nähdä dokumenttien kommentoidut katselutiedostot, mikä oli asetettu insinööriyön tavoitteeksi.

Mikäli halutussa kohdesijainnissa datayhteyden nopeus on riittävä ja käyttäjien tiedot löytyvät Active Directorystä, on mahdollista käyttää Flow-sovellusta ongelmitta. Tiedostettu ongelma on projektit, joissa JPDoc-tietojärjestelmät replikoituvat. Replikoinnista johtuvien tietoviiveiden takia tieto Flow-sovelluksen käyttämässä SQL-tietokannassa saattaa olla uudempaa kuin paikallisessa JPDocissa, joka ei ole vielä replikoitunut ajan tasalle. Flow-sovellusta voidaan kehittää edelleen tutkimaan JPDocien replikoitumisajankohtia, mutta sen toiminnallisuuden lisääminen ei kuulu tämän insinööriyön sisältöön.

5.2 Saavutetut hyödyt

Saavutetut edut ovat tässä vaiheessa vielä teoreettisia, sillä Flow-sovellusta ei ole käytetty todellisessa projektissa. Sovelluksen monipuolisuuden ja projektiehtojen muokattavuuden ansiosta se käy työkaluksi mihin tahansa dokumentinkäsittelyprosessiin riippumatta käytettävistä suunnittelujärjestelmistä tai -työkaluista. Tavoitteena ollut paperiversioiden tekemisen välttäminen onnistui, ja dokumenttien kommentointi PDF-katselutiedostoihin sekä niiden tallennus omaan tietokannasta riippumattomaan tietovarastoon toimivat.

Ajankäytön tehokkuuden parantumista ei voida teoreettisuuden lisäksi arvioida tarkkaan siksi, että edellisestä työtavasta ei, ainakaan vielä, ole tehty tarkkoja tehokkuus-

tutkimuksia. Loogisesti ajateltuna alun mahdollisten vaikeuksien jälkeen tarkastusprosessin tulisi edetä huomattavasti hallitummin ja nopeammin, sillä jokaista dokumenttia voidaan käsitellä itsenäisenä riippumatta muista dokumenteista ja Flow-sovelluksessa näkyy jatkuvasti projektidokumenttien tilat muokkaustietoineen reaaliajassa.

5.3 Tulevaisuuden kehityskohteet

PDF-piirtotyökalujen integroiminen

Kommentoidut katselutiedostot ovat nykyisessä versiossa PDF-tiedostoja, joita kommentoidaan maksullisen Adobe Acrobat -version toiminnoilla, mikä aiheuttaa laajemmassa mittakaavassa kustannuksia ohjelmalisenssien lisähankintojen ja uusimisien muodossa. Toki suurelle osalle käyttäjistä ilmainen Adobe Reader on toiminnoiltaan riittävä, mutta toisaalta kaikilla työntekijöillä tulisi olla valmius kyetä kommentoimaan dokumentteja, sillä tarvetta siihen ei välttämättä voida ennakoida.

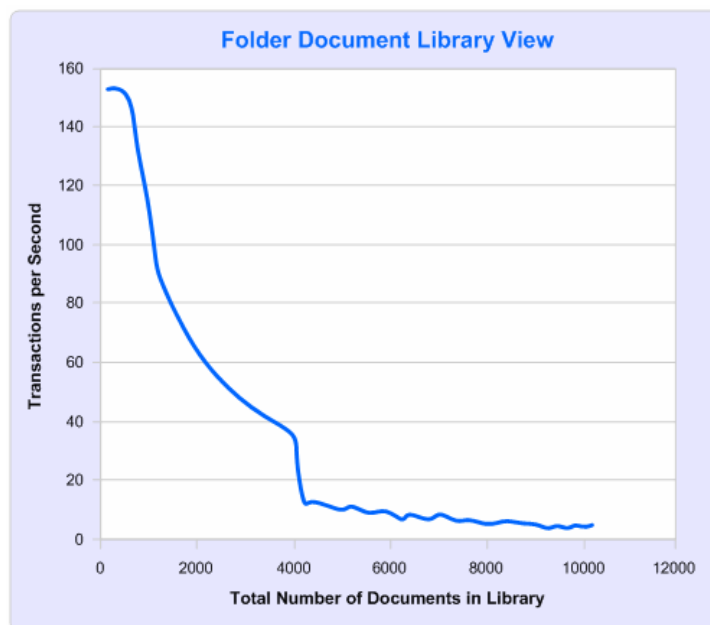
Taulukko 1: Adobe Acrobat -versioiden verolliset hinnat Suomessa huhtikuussa 2009 [33].

Ohjelmaversio	Päivityshinta	Täysi hinta
Acrobat 9 Standard	140,30 €	425,78 €
Acrobat 9 Pro	230,58 €	681,98 €
Acrobat 9 Pro Extended	328,18 €	999,18 €

Ohjelma- ja ohjelmistolisenssien tilaaminen on suuressa yrityksen monimutkainen prosessi, mikä tarkoittaa tilanteeseen varautumatta suurta mahdollisuutta menettää tehokasta työaikaa ja hidastaa työvaiheen valmistumista. Tuhansien työntekijöiden yrityksessä ohjelmistokustannukset ovat erittäin suuria. Pöyryllä käytetään vuosittain merkittävä osa budjetista ohjelmistolisenssien hankintaan. Acrobat-tuotevalikoimasta ja halvimmalla Standard-version hankkiminen normaalihinnalla tuhannelle työntekijälle maksaa yli 400 000 euroa (taulukko 1).

PDF-tiedostomuoto on kommentointiin ihanteellinen, mutta piirtotyökalujen integroiminen osaksi Flow-sovellusta olisi myös huomattavasti käyttäjäystävällisempää, sillä silloin käyttäjän ei tarvitsisi erikseen tallentaa tiedostoja ja osoittaa niiden sijaintia sovellukselle. Tällä hetkellä ongelmaa voidaan myös kiertää aktivoimalla kommentointityökalut maksullisella versioilla ja siten ne toimivat myös Adobe Readerissä. [34.]

SharePoint-dokumenttikirjaston optimointi



Kuva 22. Dokumenttien määrän vaikutus transaktioiden nopeuteen [35].

Flow'n koeversiossa kaikkien projektien dokumentit sijaitsevat samassa dokumenttikirjastossa. Se ei suoranaisesti aiheuta ongelmaa, sillä kunkin tiedoston nimi muodostuu SQL-tietokannassa pääavaimena käytettävästä DocumentID:stä ja JPDocin dokumentin pääavaimesta, joten jokainen tiedostonimi dokumenttikirjastossa on yksilöllinen. Ainoa todellinen haitta on suorituskyvyn lasku, mikäli tiedostojen määrä kasvaa erittäin suureksi (kuva 22). Kaikki SharePoint-sivujen kuormituslaskelmat kuitenkin perustuvat SharePointin oman käyttöliittymän ja siihen rakennetun hakukoneen käyttöön. Flow-sovellus käsittelee dokumenttikirjastoa ainoastaan suorittamalla suoria pyyntöjä palvelimelle, joten resursseja jää oletettavasti vielä runsaasti käytettäväksi, sillä esimerkiksi taustalla tehtävät listojen indeksoinnit voidaan

poistaa. Microsoftin ilmoittamien tietojen mukaan dokumentteja voi olla kirjastossa viisi miljoonaa ja suositeltava listaosioiden eli -rivien maksimimäärä näkymässä on 2 000.

Rajojen tullessa vastaan on mahdollista vähentää dokumenttikirjastossa olevien tiedostojen määrää muodostamalla jokaiselle projektille oma dokumenttikirjasto. Avaintietona voisi toimia tietokannassa oleva ProjectID ja siten jokainen projekti myös ”tietäisi” ilman muutoksia oman dokumenttikirjastonsa nimen ja sijainnin. Tällöin dokumenttilistaukset pysyisivät kohtuullisissa rajoissa kaikissa tapauksissa, myös tulevaisuuden mahdollisissa kehityskohteissa, joissa kenties hyödynnetään myös SharePointin omia työkaluja. Kokeiluversiossa ei tätä optimointia nähty koekäyttämisen kannalta rajoittavana tekijänä, joten päädyttiin siihen, että tiedostojen tallentaminen ja lukeminen SharePointista riittää vielä tässä vaiheessa. Ennen laajamittaista käyttöönottoa dokumenttikirjastoiden käyttöä on kuitenkin syytä optimoida.

Käyttäjätunnistus

Käyttäjätunnistuksessa hyödynnetään JPDoc-järjestelmän rajapintaa. Kutsumalla rajapinnan funktioita käyttäjätiedoilla, jotka eivät ole voimassa, palautuu HTTP-virheilmoitus ja vastaavasti oikeilla tiedoilla palautuu ehjä XML-dokumentti vastauksena palvelimelta. Todellisuudessa Flow-sovellus ei siis itse tarkista käyttäjätunnuksen ja salasanan oikeellisuutta, vaan se tiedustelee asiaa rajapinnalta ja suorittaa siitä johtopäätöksenä joko käyttäjän hyväksymisen tai hylkäämisen.

Pöyryllä on tulevaisuudessa tavoitteena saada yhdistettyä eri tyyleillä ja järjestelmillä toteutetut käyttäjien tunnistamiseen käytetyt järjestelmät. Tällöin olisi mahdollista ensimmäisen kirjautumisen jälkeen hyödyntää syötettyjä käyttäjätietoja kaikissa käytettävissä järjestelmissä. Käytännössä avaintietona käytettäisiin Windows-käyttöjärjestelmään kirjautuessa käytettävää tunnusta. Toimintatapa tunnetaan akronyymillä *SSO* (Single Sign-On), ja siinä oleellinen tekijä on eri järjestelmien välinen luottamussuhde. Pöyry pyrkii kaikin keinoin yhdistämään toimintojansa, ja toimintaa kuvataankin nykyaikaisella kehitysnimellä One Pöyry Com. SSO-toiminnon

integroiminen osaksi Flow-sovellusta parantaisi käyttäjäystävällisyyttä, sillä ohjelma käynnistyisi ilman käyttäjänimen ja salasanan pyytämistä. [7; 8.]

Rajapinta JPDociin

Rajapinnan kyselyistä hyödynnetään tällä hetkellä monia ennen tietojen esittämistä Flow-sovelluksen käyttöliittymässä. JPDociin rajapinta on toiminnoiltaan vajavainen, sillä sitä ei ole alun perin suunniteltu kuin kapeaan käyttötarkoitukseen. Jokaisen rajapinnan kyselyn tuloksena muodostetaan XML-dokumentti, joka jäsennetään tietojen keräämiseksi oikeille objekteille. XML-dokumenttiformaatti on erinomainen tiedonvaihto ja -siirtokeino sovelluksien ja tietojärjestelmien välillä, sillä se on täysin laitteisto- ja ympäristöriippumaton. Monikäyttöisyyden kustannuksena on kuitenkin tehokkuus. XML-dokumentissa yksinkertaisenkin asian ilmaisuun tarvitaan aina alku- ja loppuelementit sekä niiden sisältö.

Laajamittaisen käytön edellytyksenä olisi vähintään olemassa olevan rajapinnan korjaaminen sellaiseksi, että osittain päällekkäisiltä kyselyiltä vältyttäisiin dokumenttilistausta muodostaessa. Vielä tehokkaampi vaihtoehto olisi Document Explorerin COM-objektin toiminnallisuuden lisääminen myös tiedon hakemiseksi JPDoci-tietojärjestelmästä. Nykyisellään rajapinta soveltuu ainoastaan tietojen viemiseen.

Lokalisointi

Pöyryn virallinen kommunikointikieli on englanti. Siitä johtuen myös Flow-sovelluksen ohjelmakoodi, sen kommentit ja käyttöliittymä on kirjoitettu englannin kielellä.

Joissakin projekteissa on kuitenkin tarve työskennellä jollakin muulla kielellä asiakkaan tai muun merkittävän tahon vaatimuksesta. Erikoiskielten tapauksissa monilla projektin parissa työskentelevillä henkilöillä on vaihtelevan kokoisia ymmärtämisongelmia projektin aikana, mikä laskee työtehokkuutta.

JPDoc mahdollistaa projekteittain kielen vaihtamisen, mutta ainoastaan projektin dokumenttien metatietojen kieli vaihtuu ja sekin vain monivalintakohdissa.

Monikielisyys helpottaisi käyttäjien asemaa, sillä kukin voisi käyttää itselleen sopivinta kieltä. Käytännössä tämä tarkoittaisi Flow-sovelluksen käyttämän tietokannan muuttamista monimutkaisemmaksi, sillä kielitaulun lisäksi jokaiselle asian kuvaukselle tulisi oma taulu ja kullekin käyttäjälle oma kielitaulun viiteavain. Edellä mainitun toiminnon lisääminen ei olisi valtava työurakka, ellei maailmassa olisi useita eri kieliä, joissa on erikoiskirjaimistoja. Hyviä ongelmia aiheuttavia esimerkkejä ovat jo skandinaavisten kirjainten lisäksi kyrilliset aakkoset sekä kiinalais-japanilaiset merkistöt. Nykyisellä Flow-sovelluksen toiminnalla, jossa yhdistetään lukuisia eri tekniikoita, ei ole järkevää ryhtyä suunnittelemaan lokalisointia, mutta tulevia järjestelmiä kehitettäessä se on otettava vakavasti harkittavaksi.

Dokumenttilistan järjestäminen

Lainasana *sorttaus* on vakiinnuttanut paikkansa suomalaisessa tietokonetermistössä.

Sillä siis tarkoitetaan listan järjestämistä jonkin määriteltävän tai osoitettavan asian mukaisesti. Loogiseksi järjestelytoiminnoksi on muodostunut listan otsikkopalkin painaminen hiirellä, jolloin käyttäjä olettaa alla olevan listan järjestyvän kyseisen sarakkeen ehdoilla luonnolliseen järjestykseen. Luonnollinen järjestys tarkoittaa yleensä numeerista tai aakkosellista järjestystä, ellei ohjelmakoodiin ole toteutettu erillistä vertailumetodia.

Microsoft Visual Studio -kehitysympäristö antaa sovelluskehittäjän päättää, ovatko listojen otsikkorivin painettavia vai eivät. Toistaiseksi Flow-sovelluksessa otsikkorivit on määritelty painettaviksi, mutta niillä ei ole lainkaan toiminnallisuutta, mikä saattaa hämmentää käyttäjää. Lähitulevaisuuden tavoitteena on kuitenkin kyseisen toiminnallisuuden lisääminen, sillä se parantaisi käyttäjän mahdollisuuksia paikallistaa haluttu tai halutun tyyppinen dokumentti nopeasti ja pienellä vaivalla. Nykyisessä versiossa dokumentit on järjestetty niiden dokumentti-id:n perusteella ja listojen järjestäminen on hoidettava Microsoft Excelissä, johon on kuitenkin rakennettu suora export-toiminto.

6 Yhteenveto

Insinööriyössä suunniteltiin ja toteutettiin dokumentinkäsittelyprosessin hallintatyökalun koeversio, joka yhdistää monien järjestelmien tietoresursseja. Se kykenee hakemaan tietoja Microsoftin Active Directorysta ja Lotus Notes -tietojärjestelmässä sijaitsevasta dokumentinhallintajärjestelmästä. Sovellus tallentaa tietoja SQL-relaatiotietokantaan ja SharePoint-palvelimen dokumenttikirjastoon. Sovelluksella voidaan ladata Internetissä tai suojattujen yhteyksien takana olevia tiedostoja kirjautuneen käyttäjän tietokoneelle. Lisäksi se kykenee hyödyntämään dokumentinhallintajärjestelmän COM-rajapintaa lisätessä tiedostoja tai avatessa sen käyttöliittymä halutun dokumentin kohdalta.

Sovelluksen tarkoitus oli käytännössä vähentää dokumentinkäsittelyprosessin nykyisiä ongelmia, joita globalisoituminen ja hajautetut projektit aiheuttavat. Työtapojen yhtenäistämisen vaatimuksena on PDF-komentointityökalujen käyttö, mikä saattaa monille olla aluksi ongelmallista.

Sovellus kehitettiin Microsoft Visual Studion integroidussa kehitysympäristössä C#-ohjelmointikielellä. Sitä suoritetaan itsenäisenä Windows-sovelluksena ja useista kommunikointikeinoistaan ja -kohteistaan huolimatta sen käyttö ei vaadi muutoksia, eikä vaikuta nykyisen dokumentinhallintajärjestelmän toimintaan. Sovellusta on koekäytetty vasta pienessä ja rajatussa ympäristössä. Vasta laajempi käyttö mahdollistaa paremmin työtehokkuuden parantumisen tutkimisen ja mittaamisen. Koekäyttäjille sovelluksen ensivaikutelma on ollut erittäin positiivinen, mutta muiden järjestelmien ja tiedonsiirtokapasiteettien rajoitukset ja niiden vaihtelut aiheuttavat vielä runsaasti kehitettävää ennen mahdollisen laajemman maailmanlaajuisen käyttämisen aloittamista.

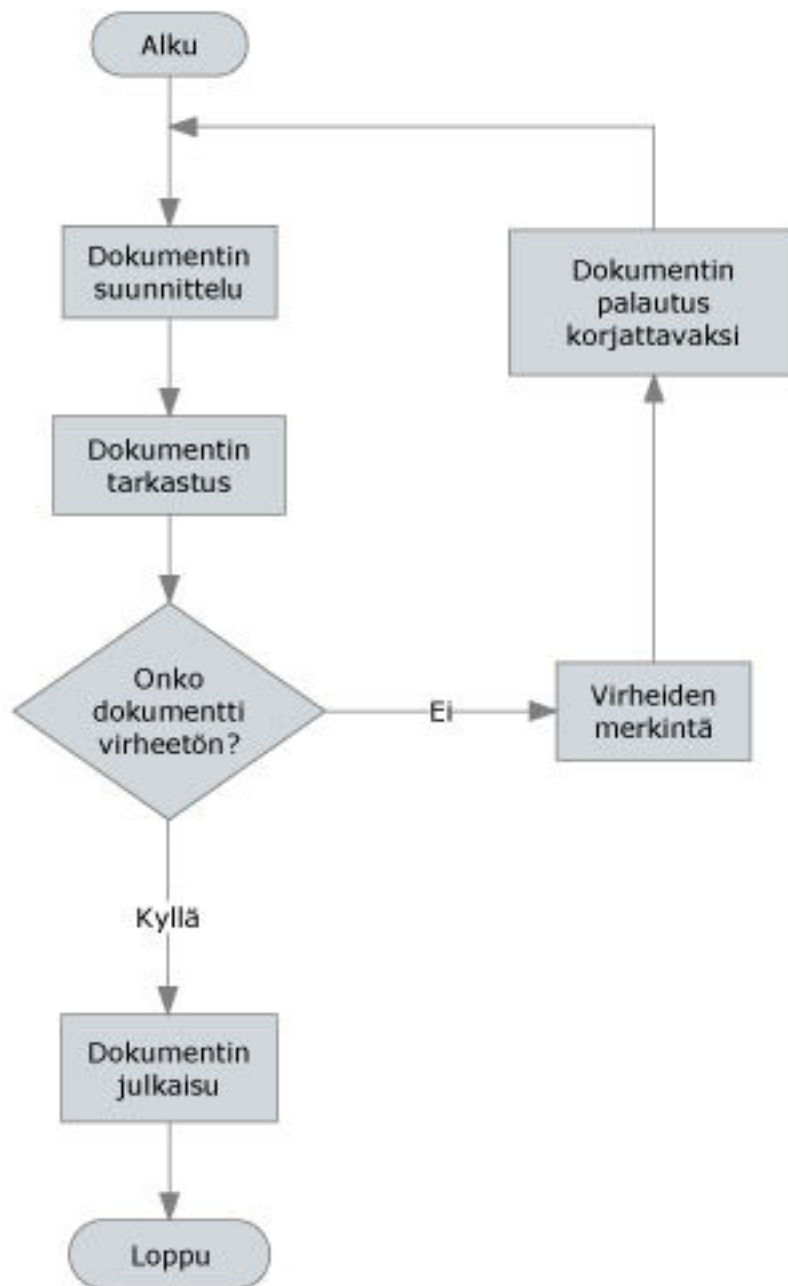
Kaiken kaikkiaan insinööriyön tekeminen todellisessa työympäristössä oli erittäin mielekästä ja haastavaa. Käytäntö eroaa usein teoriasta -lausahdus tuli hyvin tutuksi projektin edetessä, kun teoriassa yksinkertaiset ja pienet lisäykset aiheuttivat useiden päivien ongelmanratkomishetkiä, jotka kuitenkin opettivat paljon uusia asioita.

Lähteet

- 1 Salminen, Airi. Hajautettu hypermedia. (WWW-dokumentti.) Jyväskylän yliopisto, tietojenkäsittelytieteiden laitos.
<<http://users.jyu.fi/~airi/opetus/tkod54/kasitteet.html>>. Päivitetty 3.10.1997.
Luettu 20.4.2009.
- 2 ISO suomen kielioppi. (WWW-sanakirja). Kotimaisten kielten tutkimuskeskus.
Luettu 20.4.2009.
- 3 Peltopuro, Jani. Jaosjohtaja, Pöyry Industry Oy. Keskustelut 15.10.2008-20.4.2009.
- 4 Kallio, Juha-Matti. Suunnitteluinsinööri, Pöyry Industry Oy, Vantaa. Keskustelu 3.11.2008.
- 5 Utela, Heikki. Suunnittelija, Pöyry Industry Oy, Vantaa. Keskustelu 12.11.2008.
- 6 Virolainen-Borg, Kirsi. Suunnitteluinsinööri, Pöyry Industry Oy, Vantaa. Keskustelut keväällä 2009.
- 7 Äyräs, Jaakko. IT-spesialisti, Pöyry Application Services Oy, Vantaa. Keskustelut 1.10.2008-20.4.2009.
- 8 Niskanen, Veli-Veikko. IT-spesialisti, Pöyry Application Services Oy, Vantaa. Keskustelu 14.4.2009.
- 9 Allen, R. & Hunter, L. Active Directory Cookbook. 2nd Edition. Sebastopol: O'Reilly Media, 2006.
- 10 Miettinen, Sami. IT-arkkitehti, Pöyry Application Services Oy, Vantaa. Keskustelu 18.3.2008.
- 11 The History of Notes and Domino. (WWW-dokumentti.) IBM.
<<http://www.ibm.com/developerworks/lotus/library/lS-NDHistory/>> Päivitetty 20.12.2005. Luettu 6.2.2009.
- 12 Robinson, S., Nagel, C., Glynn, J., Skinner, M., Watson, K. & Evjen, B. Professional C#. 3rd Edition. Indianapolis: Wiley Publishing, Inc., 2004.
- 13 Microsoft Visual Studio. (WWW-dokumentti.) Wikipedia.
<http://en.wikipedia.org/wiki/Microsoft_Visual_Studio>. Päivitetty 18.4.2009.
Luettu 18.4.2009.
- 14 .NET Framework. (WWW-dokumentti.) Wikipedia.
<http://en.wikipedia.org/wiki/.NET_Framework>. Päivitetty 19.4.2009. Luettu 20.4.2009.

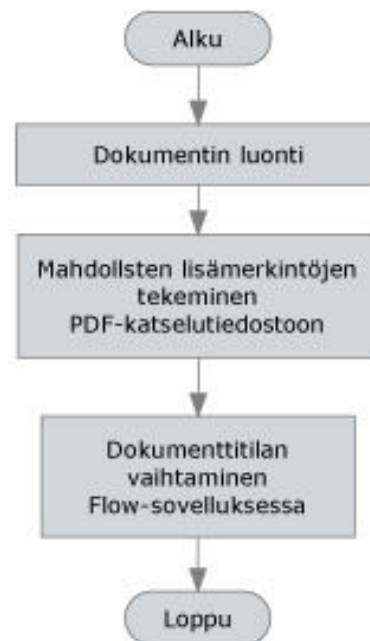
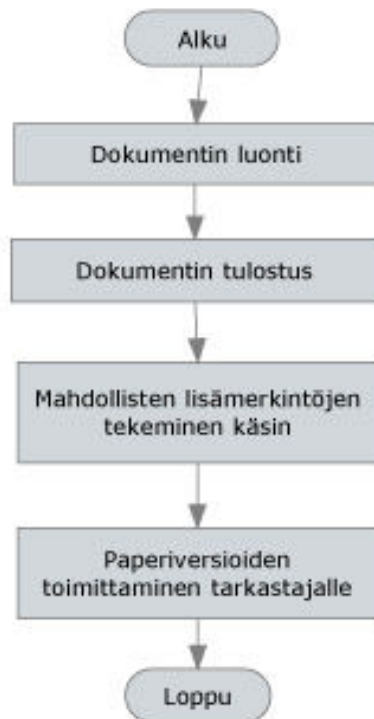
- 15 Deep Inside C#: An Interview with Microsoft Chief Architect Anders Hejlsberg. (WWW-dokumentti.) O'Reilly Windows DevCenter. <<http://www.windowsdevcenter.com/lpt/a/2273>>. Päivitetty 8.1.2000. Luettu 18.4.2009.
- 16 Adobe - PDF: The Global Standard For Trusted, Electronic Documents and Forms. (WWW-dokumentti.) Adobe. <<http://www.adobe.com/pdf/about/history/>>. Luettu 20.4.2009.
- 17 Portable Document Format. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Portable_Document_Format>. Päivitetty 20.4.2009. Luettu 20.4.2009.
- 18 SQL. (WWW-dokumentti.) Wikipedia. <<http://en.wikipedia.org/wiki/SQL>>. Päivitetty 19.4.2009. Luettu 20.4.2009.
- 19 Sheldon, Robert. SQL A Beginner's Guide. 2nd Edition. Emeryville: McGraw-Hill/Osborne, 2003.
- 20 Robbins, Thom. Introducing SharePoint Web Services. (WWW-dokumentti.) .NET Developer's Journal. <<http://dotnet.sys-con.com/node/47341>>. Päivitetty 7.12.2004. Luettu 19.4.2009.
- 21 Windows SharePoint Services 3.0 Overview. (WWW-dokumentti.) Microsoft. <<http://technet.microsoft.com/en-us/windowsserver/sharepoint/bb684453.aspx>>. Luettu 20.4.2009.
- 22 Working with SharePoint document libraries. (WWW-dokumentti.) Microsoft. <<http://office.microsoft.com/en-us/sharepointtechnology/HA011412871033.aspx>>. Luettu 20.4.2009.
- 23 Ekholm, Niklas. Sovelluskehittäjä, Pöyry Application Services Oy, Vantaa. Keskustelu 20.4.2009.
- 24 Wesley, Addison. LDAP Directories Explained: An Introduction and Analysis. Boston: Pearson Education, Inc., 2003.
- 25 MSDN .NET Framework Developer Center. (WWW-dokumentti.) Microsoft. <[http://msdn.microsoft.com/en-us/library/system.string_members\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.string_members(VS.71).aspx)>. Luettu 20.4.2009.
- 26 MSDN Microsoft Developer Network (WWW-dokumentti.) Microsoft. <[http://msdn.microsoft.com/en-us/library/ms695276\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms695276(VS.85).aspx)>. Luettu 20.4.2009.

- 27 SQL Server Developer Center. (WWW-dokumentti.)
<[http://msdn.microsoft.com/en-us/library/aa933198\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa933198(SQL.80).aspx)>. Microsoft.
Luettu 20.4.2009.
- 28 Greenwich Mean Time (GMT). (WWW-dokumentti.)
<<http://www.greenwichmeantime.com/what-is-gmt.htm>>. World Wide Portals.
Päivitetty 24.3.2009. Luettu 20.4.2009.
- 29 MSDN .NET Framework Developer Center. (WWW-dokumentti.) Microsoft.
<[http://msdn.microsoft.com/en-us/library/
system.data.sqlclient.sqlconnection.aspx](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.aspx)>. Luettu 20.4.2009.
- 30 MSDN .NET Framework Developer Center. (WWW-dokumentti.) Microsoft.
<[http://msdn.microsoft.com/en-us/library/
system.data.sqlclient.sqldataadapter\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldataadapter(VS.71).aspx)>. Luettu 20.4.2009.
- 31 Kostamo, Timo. Sovelluskehittäjä, Pöyry Application Services Oy, Vantaa.
Keskustelut 23.2.2009-20.4.2009.
- 32 Hook, Brian. Write Portable Code. San Francisco: No Starch Press, Inc., 2005, s.
2.
- 33 Adobe Store, Finland. (WWW-kauppasivusto.) Adobe.
<[https://store2.adobe.com/cfusion/store/html/index.cfm?store=OLS-
FI&event=displayCatalog&catalogOID=1858553](https://store2.adobe.com/cfusion/store/html/index.cfm?store=OLS-FI&event=displayCatalog&catalogOID=1858553)>. Luettu 16.4.2009.
- 34 Acrobatin ja Readerin vertailu. (WWW-dokumentti.) Adobe.
<<http://www.adobe.com/fi/products/acrobat/reader.html>>. Päivitetty 2008.
Luettu 16.4.2009.
- 35 Plan for software boundaries. (WWW-dokumentti.) Microsoft.
<<http://technet.microsoft.com/en-us/library/cc262787.aspx>>. Päivitetty
12.6.2008. Luettu 16.4.2009.

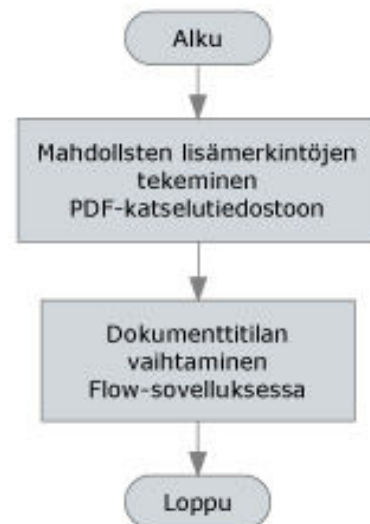
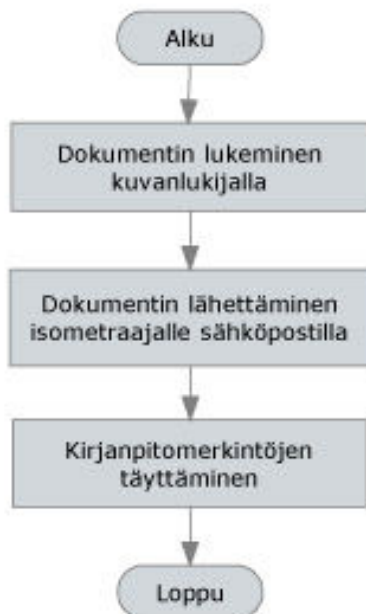
Liite 1: Perustarkastusprosessin vuokaavio

Liite 2: Flow-sovelluksen aiheuttamat muutokset työprosessin kulkuun tutkitussa projektissa

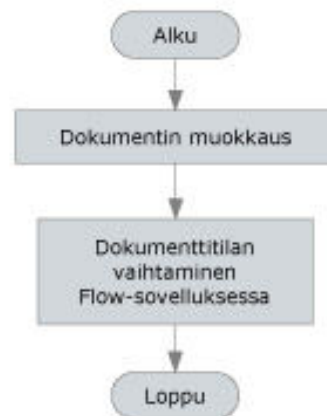
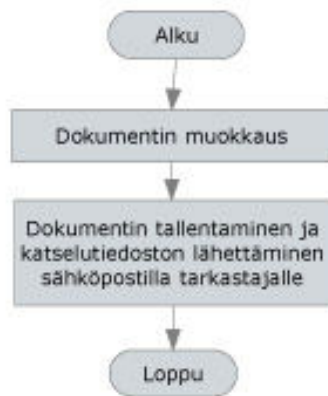
SUUNNITTELIJAN TYÖVAIHE:



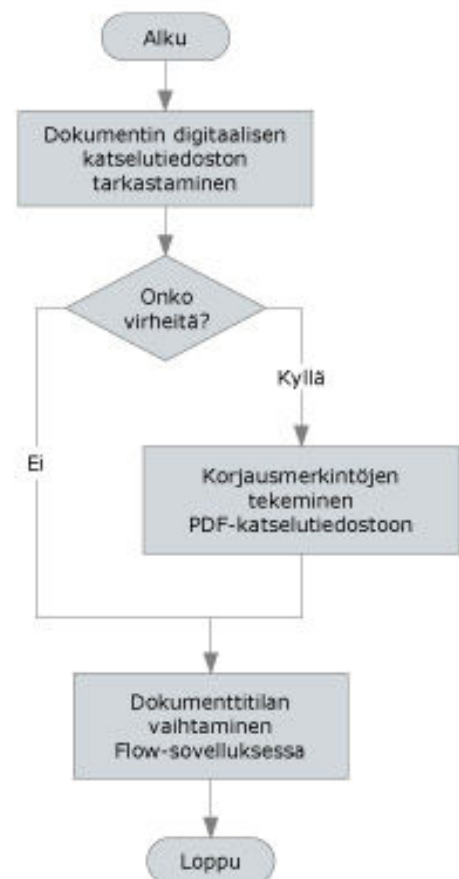
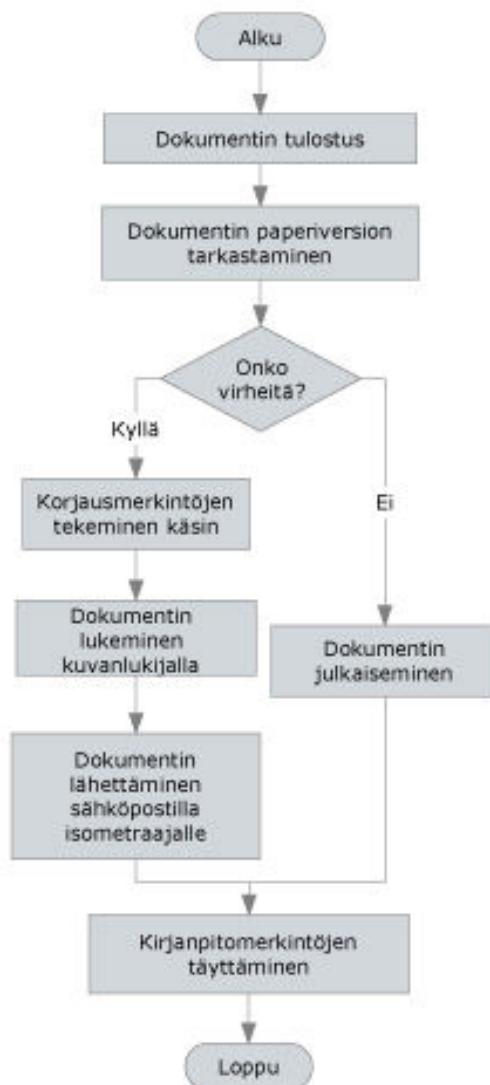
TARKASTAJAN VÄLITYÖVAIHE:



ISOMETRAAJAN TYÖVAIHE:



TARKASTAJAN TYÖVAIHE:



Liite 3: Dokumentin metatiedon XML-listaus

```

<?xml version="1.0" encoding="iso-8859-1" ?>
- <getnumber>
  <Messages />
  <DocID>16B0757-02001</DocID>
  <Issuer>JPH</Issuer>
  <Area>21</Area>
  <Type>P230</Type>
  <Name1>Auflösung und Dickstoffreinigung</Name1>
  <Name2 />
  <Phase>2</Phase>
  <Contr />
  <IPart />
  <IRev>B</IRev>
  <OPart />
  <ORev />
  <ODate>4.2.2008</ODate>
  <RDate>16.1.2009</RDate>
  <Pos />
  <Title1_2 />
  <Title2_2 />
  <RStat />
  <Auth>K. Tenhola /RMK</Auth>
  <ODType />
  <DDPrel />
  <DDBind />
  <DDFin />
  <AInfo />
  <Scale />
  <Size />
  <Info1 />
  <Info2 />
  <Info3 />
  <Info4 />
  <Info5 />
  <NFinal />
  <NoArch>Yes</NoArch>
  <RecDate />
  <CIDoc />
  <Areamame />
  <Typename />
  <Classname />
  <Phasename />
  <NumSet />
  <Errors />
  <Status>No base updates</Status>
  <Edited>19.1.2009 12:17:41</Edited>
</getnumber>

```


Liite 4: Keskeisten dokumentti- ja projektiluokkien toteutetut metodit

Project
Class

+ Fields

+ Properties

- Methods

- addDocumentsToWorkflow
- addGroup
- findServer
- getFlowDocuments
- getNotDefinedGroupID
- getProjectID
- getStatuses
- initializeDocumentListview
- isWorkflowProject
- loadAdditionalInfoToDocument
- loadAreas
- loadClassXML
- loadDocuments
- loadGroups
- loadTypes
- loadWorkers
- printAreas
- printDocuments
- printMyDocuments
- printTypes
- printWorkflowDocuments
- Project (+ 1 overload)
- removeGroup
- ToString
- updateWorkers

Document
Class

+ Fields

+ Properties

- Methods

- addCommentedPdf
- addExtraInfo
- addInfo
- addTextComments
- addToWorkflow
- addWorkfileAndViewfile
- Document
- getCommentedPdfAsListItem
- getHistoryAsListViewItems
- getJpDocFile
- getListItem (+ 1 overload)
- getPhaseNumber
- getPreviousEditor
- getSharePointFile
- getStatus
- getTextComments
- getTime
- getViewfilesAsListViewItems
- getWorkfilesAsListViewItems
- hasCommentedPdf
- hasTextComments
- isInWorkflow
- isMyDocument
- isValidUrl
- JpDocFileDownloader_DoWork
- JpDocFileDownloader_RunWorkerCompleted
- openCommentedPdf
- openViewfile (+ 1 overload)
- openWorkfile (+ 1 overload)
- removeAllComments
- removeCommentedPdf
- removeTextComments
- saveCommentedPdf
- saveJpDocFile
- SharePointFileDownloader_DoWork
- SharePointFileDownloader_RunWorkerComple...

Liite 5: Vuokaavio sovelluksen toiminnasta dokumenttitilavaihdossa

