



TAMPEREEN TEKNILLINEN YLIOPISTO

**JOUNI NURMILO**  
**MICROSOFT DYNAMICS AX 2009 ENTERPRISE PORTAL**  
**-SOVELLUSKEHITYS**

Diplomityö

Tarkastaja: Tommi Mikkonen  
Tarkastaja ja aihe hyväksytty  
Tieto- ja sähkötekniikan  
tiedekuntaneuvoston  
kokouksessa 7.10.2009

# TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma

**NURMILO, JOUNI:** Microsoft Dynamics AX 2009 Enterprise Portal

-sovelluskehitys

Diplomityö, 65 sivua

Toukokuu 2011

Pääaine: Hajautetut ohjelmistot

Tarkastaja: professori Tommi Mikkonen

Avainsanat: Microsoft Dynamics AX 2009, Enterprise Portal, User controller, käyttäjäohjain, web-käyttöliittymä, sovelluskehitys

Tässä diplomityössä käsitellään Microsoft Dynamics AX 2009 -toiminnanohjausjärjestelmän Enterprise Portal nimisen web-käyttöliittymän toimintaa ja ohjelmointia. Webin käyttö on ollut nousemassa johtavaksi trendiksi myös hyötyohjelmien maailmassa, ja siihen tarkoitukseen Enterprise Portal on luotu. Enterprise Portal -web-käyttöliittymä on uudistettu AX:n 2009 versioon täydellisesti edellisistä versioista, ja sen seurauksena myös sovelluskehitys tälle alustalle on täysin uutta.

Diplomityötä aloitettaessa ei Enterprise Portal ohjelmoinnista ollut saatavilla juuri mitään tietoa. Tästä muodostuikin työlle ongelma, eli kuinka Enterprise Portal toimii ja miten sitä ohjelmoidaan. Erityisen merkittävässä roolissa työn kannalta on juuri miten ja mihin Enterprise Portal sovelluksia ohjelmoidaan.

Työtä tehdessä selvisi, että Enterprise Portal -sovelluskehitystä tulee tehdä kolmessa eri ympäristössä: AX-palvelin, Visual Studio ja SharePoint. Työssä käydään lävitse eri ympäristöjen osalta tehtävät toimenpiteet. Tämän lisäksi lopuksi on toteutettu esimerkkisovellus, missä luodaan yksinkertainen kohteiden hallintajärjestelmä Enterprise Portal -ympäristöön olemassa olevien taulujen päälle.

Enterprise Portal osoittautui hyvin toimivaksi ja helpoksi tavaksi toteuttaa web-käyttöliittymä Microsoft Dynamics AX 2009 -toiminnanohjausjärjestelmälle. Joitain ongelmia tai kummallisuuksia Enterprise Portal -järjestelmästä löytyi, mutta kokonaisuutena Enterprise Portal osoittautui oikeaksi tavaksi toteuttaa web-käyttöliittymä.

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**NURMILO, JOUNI:** Microsoft Dynamics AX 2009 Enterprise Portal Application Development

Master of Science Thesis, 65 pages

May 2011

Major: Distributed Software

Examiner: Professor Tommi Mikkonen

Keywords: Microsoft Dynamics AX 2009, Enterprise Portal, User controller, web user interference, application development

This thesis is about how Microsoft Dynamics AX 2009 enterprise resource planning system's Enterprise Portal web user interface operates and how it can be programmed. Using the web has been a rising trend also on utility programs and Enterprise Portal is made for that purpose. Enterprise Portal web user interface is totally redesigned in the AX 2009 version, and thus also application development for this platform is totally new.

At the begin of the work carried out for the thesis, there were very little information about Enterprise Portal development. This became the central problem to be solved in the thesis: How the Enterprise Portal works and how it is developed. One especially important role for the work is how and where Enterprise Portal application development is done.

During the study it was found out that Enterprise Portal application development has to be done with three different environments: AX-server, Visual Studio and SharePoint. The thesis introduces the different environments and tasks done there for the Enterprise Portal. Towards the end of the thesis, there is an example application for simple asset management for Enterprise Portal build on existing tables.

To summarize the results, Enterprise Portal proved to be a simple and well-functioning way to build web user interface for Microsoft Dynamics AX 2009 - enterprise resource planning system. Some problems and oddities were identified, but besides of those Enterprise Portal framework turned out to be the right way for web user interface implementation.

# ALKUSANAT

Tämä työ on tehty kevään 2009 ja 2011 välisenä aikana Tampereen teknillisen yliopiston ohjelmistotekniikan laitokella hajautetut järjestelmät pääaineeseen. Työn teettäjänä toimi Solteq Oyj.

Haluan kiittää työn ohjaajana toiminutta professori Tommi Mikkosta työn aikana saamastani ohjauksesta tuesta ja vihjeistä. Solteq Oyj:tä mahdollisuudesta diplomityön tekemiseen sekä joustavasta työskentelymahdollisuudesta opintojen aikana.

Lisäksi haluan kiittää vaimoani Johanna Nurmiola ja vanhempiani saamastani tuesta ja siskoani Hanna Nurmiola työn oikoluvusta.

Tampereella 18. toukokuuta 2011

Jouni Nurmiola  
jonurmi@uusikaupunki.fi

## SISÄLLYS

1.	Johdanto .....	1
2.	Microsoft Dynamics AX .....	3
2.1.	Käyttöliittymä .....	4
2.2.	Teknologia.....	5
2.3.	Application Object Tree (AOT).....	6
2.4.	AX-ohjelmointi .....	7
2.5.	X++ .....	8
3.	Enterprise Portal.....	10
3.1.	Käyttöliittymä .....	10
3.2.	Tekniikka.....	11
3.3.	Sivun käsittely.....	12
3.4.	.NET Business Connector .....	13
3.5.	Roolikeskukset .....	13
3.5.1.	Pinot.....	14
3.5.2.	Pikalinkit.....	15
3.5.3.	Business overview .....	16
3.5.4.	Työluettelo .....	17
3.5.5.	Raportit .....	17
3.6.	Microsoft SharePoint Services.....	18
4.	Enterprise Portal -sivun luonti .....	19
4.1.	AX.....	19
4.1.1.	Web-solmu.....	19
4.1.2.	Data Set.....	20
4.1.3.	Tiedon hakeminen.....	21
4.1.4.	Metodikutsut .....	22
4.1.5.	Tiedon rajaaminen .....	23
4.1.6.	Navigointi .....	24
4.2.	Enterprise Portal sivun luominen SharePoint ympäristössä.....	25
4.2.1.	Sivun luominen .....	26
4.2.2.	Ohjaimen lisääminen sivulle.....	27
4.2.3.	Ohjainten välinen vuorovaikutus .....	27
4.2.4.	Ohjaimen poistaminen sivulta .....	28
4.2.5.	Sivun koodin vieminen ja tuominen .....	29
5.	Visual Studio-kehitys .....	31
5.1.	Käyttäjäohjaimen perusteet.....	31
5.2.	Kehitysympäristö .....	32
5.2.1.	Kehityksen työnkulku .....	33
5.2.2.	Komponenttien ominaisuudet.....	35
5.3.	Yleisiä toimintoja käyttäjäohjaimissa .....	36
5.3.1.	Eventit eli tapahtumat .....	36

5.3.2.	Käyttjäohjaimen tilan määrittäminen.....	37
5.3.3.	Data Set-kutsut.....	39
5.3.4.	Käännöstekstit.....	39
5.3.5.	Uudelleen ohjaaminen .....	40
6.	AX-sidonta .....	41
6.1.	AxDataSource .....	41
6.2.	AxGridView .....	43
6.3.	AxForm - lomake .....	46
6.4.	AxToolbar - työkalurivi .....	48
6.5.	AxLookup - poimintaikkuna .....	51
7.	Esimerkkisovellus .....	54
7.1.	Kohteiden hallinta .....	54
7.2.	Käyttjäohjaimen katselutila ja ulkoasu.....	55
7.3.	Lisäys ja muokkaus ominaisuuksien lisääminen.....	57
7.4.	Linkit käyttjäohjaimiin .....	58
7.5.	Grid-näkymä .....	59
7.6.	SharePoint-sivut .....	59
8.	Arviointi .....	61
9.	Yhteenveto .....	63
	Lähteet.....	65

# 1. JOHDANTO

Internet ja webin käyttö ovat jo vuosia tehneet tuloaan myös hyötyohjelmistomaailmassa. Käytännön toteutukset ovat kuitenkin jääneet aikaisemmin lähinnä yksittäisiin toimintoihin tai erittäin kevyisiin järjestelmiin, jotka ovat soveltuneet pienille yrityksille. Microsoft on kuitenkin uuden Dynamics AX 2009-toiminnanohjausjärjestelmän myötä ottanut ison askeleen eteenpäin paremman web-käytön kanssa uudistamalla Enterprise Portal -järjestelmänsä.

Enterprise Portal -järjestelmän sivut toimivat SharePoint-julkaisualustalla, ja koostuvat AX:n palvelimelta tietonsa hakevista ja suorittavista erilaisista komponenteista. Erityisen merkityksellinen näistä Enterprise Portalin komponenteista on käyttäjäohjain (englanniksi User Controller), joka on web-vastine AX:n Windows-sovelluksen näytöille. Näillä käyttäjäohjaimilla AX:n toimintalogiikan vieminen mahdollistetaan web-käyttöliittymään, ja sen pitäisi olla suhteellisen helppoa ja vaivatonta.

Diplomityön tavoitteena on selvittää kuinka Microsoft Dynamics AX 2009-toiminnanohjausjärjestelmän Enterprise Portal -lisäosa toimii, ja kuinka ohjelmoidaan sen käyttäjäohjaimia. Tämän lisäksi pyritään arvioimaan onko kyseinen laajennos järkevä tapa toteuttaa web-käyttöliittymää AX 2009-järjestelmälle.

Ennen työn aloitusta oletuksena oli, ettei Enterprise Portal -ohjelmointi vaatisi juuri AX-ohjelmointi osaamista. Tämä oletus osoittautui kuitenkin nopeasti vääräksi, ja selvisi, että AX-ohjelmoinnista olisi hyvä ymmärtää ainakin perusteet, kuten myös AX:n toimintafilosofiasta. Työtä tehdessä selvisi, että joitain kriittiset ohjelmaolioita on pakko luoda AX:n kantaan, ja usein niihin täytyy myös kirjoittaa AX-ympäristön ohjelmakoodia.

AX-ohjelmointi tapahtuu pääasiassa englanninkielisessä tilassa. Koska AX:sta on kuitenkin olemassa suomenkielinen käännös, käytetään diplomityössä pääasiassa suomenkielisiä termejä. Ohjelmaolioissa ja ohjelmointitilassa käytetään kuitenkin englanninkielisiä termejä, joten työssä käytetään myös näitä termejä etenkin viitattaessa suoraan ohjelmaolioihin.

Diplomityössä perehdytään Microsoft Dynamics AX 2009 järjestelmän sekä Enterprise Portalin toimintaan. Työssä käydään lävitse AOT:ssa ja SharePointissa tehtävät Enterprise Portalin kannalta välttämättömät ohjelmointi sekä säätötoimet. Lisäksi syvennytään Enterprise Portal käyttäjäohjaimien ohjelmointiin Visual Studiassa sekä käydään lävitse Enterprise Portalin omat komponentit AX-toimintaan.

Työn rakenne on seuraava: toisessa luvussa tutustutaan Microsoft Dynamics AX -käyttöjärjestelmään yleisellä tasolla, sekä syvennytään hieman siinä käytettyyn

teknologiaan. Luvussa kolme perehdytään Enterprise Portal -käyttöjärjestelmään, sen käyttöliittymään, tekniikkaan sekä kuinka se on yhteydessä AX-palvelimelle. Lisäksi luvussa kolme perehdytään roolikeskusnäkömään, sekä näkymän sisältämiin SharePoint Web Parteihin. Luvussa neljä syvennyttään Visual Studion ulkopuolella tapahtuvaan käyttäjäohjainkehitykseen. Luvussa käydään lävitse AX:n AOT-palvelimella tehtävät toimenpiteet Enterprise Portal -järjestelmää varten, sekä SharePoint-ympäristössä tapahtuva sivunluominen ja kehittäminen. Luvussa viisi syvennyttään Visual Studiassa tapahtuvaan käyttäjäohjainkehitykseen yleisellä tasolla, sekä käydään lävitse Visual Studiassa tapahtuvaa koodausta yleisten käyttötapauksen mukaan. Luvussa kuusi käsitellään Enterprise Portal -laajennuksen tärkeimmät komponentit sekä kuinka niitä käytetään. Luvussa seitsemän tehdään esimerkkisovellus kohteiden hallintaan. Luvussa kahdeksan arvioidaan Enterprise Portalin käyttökelpoisuutta ja luku yhdeksän on työn kokoava yhteenveto.

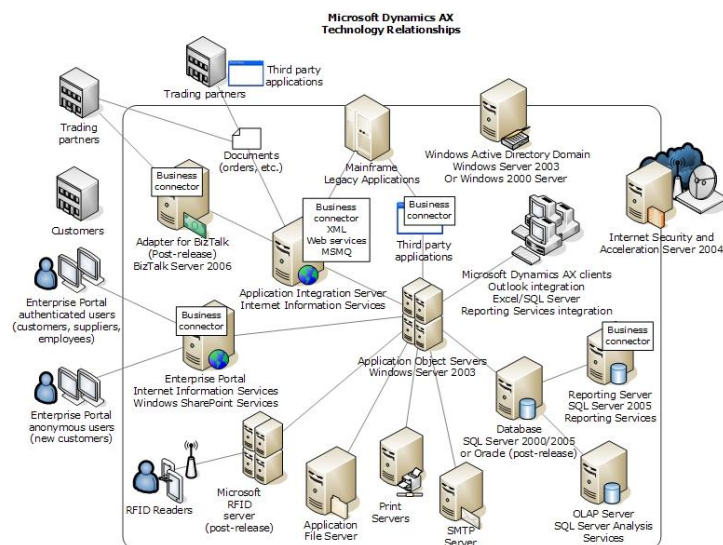


## 2. MICROSOFT DYNAMICS AX

Microsoft Dynamics AX on Microsoftin Dynamics tuoteperheen toiminnanohjausjärjestelmä, joka on tarkoitettu erityisesti keskisuurten yritysten käyttöön. Se on toteutettu mukautuvaksi ja integroitavaksi järjestelmäksi, joka on suunniteltu toimimaan yhdessä muiden Microsoftin ohjelmistojen kanssa. Dynamics AX on suunniteltu yli 1000 käyttäjän ohjelmistoksi, mutta se soveltuu hyvin myös huomattavasti pienempien käyttäjäryhmien ympäristöksi. Dynamics AX on suunniteltu toimimaan monikansalliseen ja -kieliseen toimintaympäristöön, ja ympäristöä voidaan mukauttaa käyttäjäryhmiin perustuvien roolien sekä käyttäjäkohtaisiin asetuksiin. [3]

Dynamics AX 2009-toiminnanohjausjärjestelmä koostuu useista valmiista moduuleista, jotka käsittelevät oman alueensa toimintoja. Dynamics AX 2009:n on hankittavissa muun muassa seuraavat moduulit: finanssihallinta, toimitusketjunhallinta, kauppa ja logiikka, projekti, asiakassuhteidenhallinta ja henkilöstöhallinta. Liiketoiminnan tarpeiden muuttuessa voidaan Dynamics AX-järjestelmää muuttaa asentamalla siihen uusia moduuleita, tai muokkaamalla jo olemassa olevia osia paikallisilla räätälöinneillä. [1]

Dynamics AX 2009:ää voidaan käyttää kahden erilaisen valmiin käyttöliittymän kautta. Toinen käyttöliittymä on perinteinen Windows-sovellus, joka on asennettuna käyttäjän koneelle. Toinen käyttöliittymä on puolestaan Microsoft Sharepoint-julkaisujärjestelmän päälle luotu Enterprise Portal, joka toimii verkon ylitse normaalien Internet sivujen tapaan, eikä vaadi erillisten ohjelmistojen asentamista käyttäjän koneelle toimiakseen. [1]



Kuva 2.1: Microsoft Dynamics AX 2009 arkkitehtuurikuva

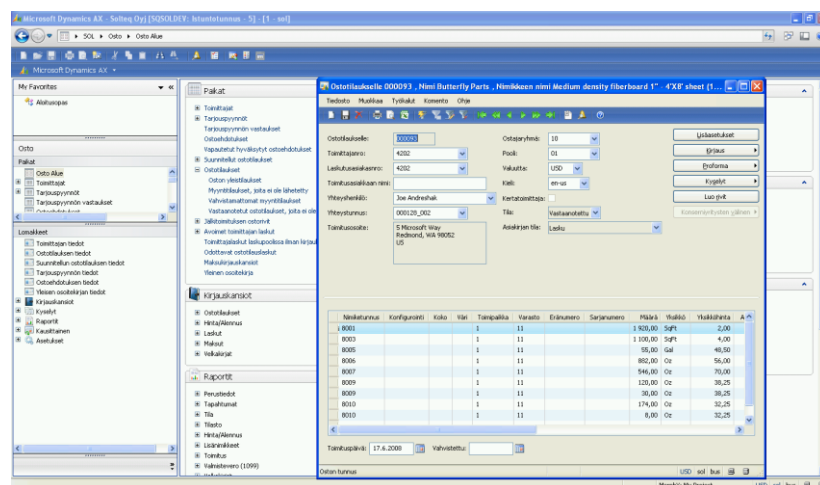
Kuvassa 2.1 on esitetty Microsoft Dynamics AX 2009 järjestelmän arkkitehtuuri. Kuvassa keskellä on Application Object Serverit, eli AX-palvelimet. Ne ovat yhteydessä tietokantapalvelimeen, joka voi olla joko SQL Server tai Oraclen palvelin. Tietokantapalvelimet taas tarjoavat muun muassa raportointipalveluita. Peruskäyttöliittymänä on AX-asiakas, joka näkyy kuvassa AX-Clients ryppäänä. Asiakkaat voivat hyödyntää myös muita Microsoftin tuotteita työskennellessään AX:n parissa. Yhteyksiä ulkomaailmaan hallinnoidaan Business Connector liittymällä, jonka takaa löytyy myös Enterprise Portal, jota voidaan käyttää kirjautuneena tai kirjautumattomana. Tämän lisäksi AX-palvelimet toimivat yhdessä monien muiden kuten posti- ja tulostinpalvelinten kanssa.

## 2.1. Käyttöliittymä

Kun käyttäjä käynnistää AX-asiakasohjelman, aukeaa hänelle ensimmäisenä käyttäjän rooliin perustuva roolikeskus-sivu, mikäli Enterprise Portal ja roolikeskukset on asennettu kyseiseen AX-instanssiin. Roolikeskuksen tarkoitus on sisältää yhdessä näkymässä käyttäjän kannalta mielenkiintoisimmat tiedot visuaalisena esityksenä.

Navigoiminen AX-asiakkaassa tapahtuu valitsemalla käytettävän moduulin näytön vasemmassa laidassa sijaitsevalta moduulilistalta. Moduulilistalla näkyvät kaikki kyseiseen AX-instanssiin asennetut moduulit, joita käyttäjällä on oikeus käyttää. Moduulin valitseminen aukaisee moduulikohtaisen yleiskatsausnäytön, jolla on linkkejä moduulin sisältämiin näyttöihin, toimintoihin tai raportteihin. AX:n modulaarisuus näkyy siis ohjelman lävitse hyvin selvästi myös käyttäjälle.

AX:ssa tietoja käsitellään tai katotaan joko joukkona tietoja, jolloin tietueet ovat yleensä listatut grid-taulukon muodostamaan yleisnäkymään, tai yksittäisinä tietueina omalla ikkunallaan. Näyttö voi sisältää myös molempia komponentteja. Tällainen näyttö voi olla esimerkiksi laskun esityksen tapauksessa, jolloin laskun tiedot voidaan näyttää yksittäisenä ja laskuun liittyvät rivit omassa gridissä, jossa on kaikki laskuun liittyvät rivit nähtävillä.

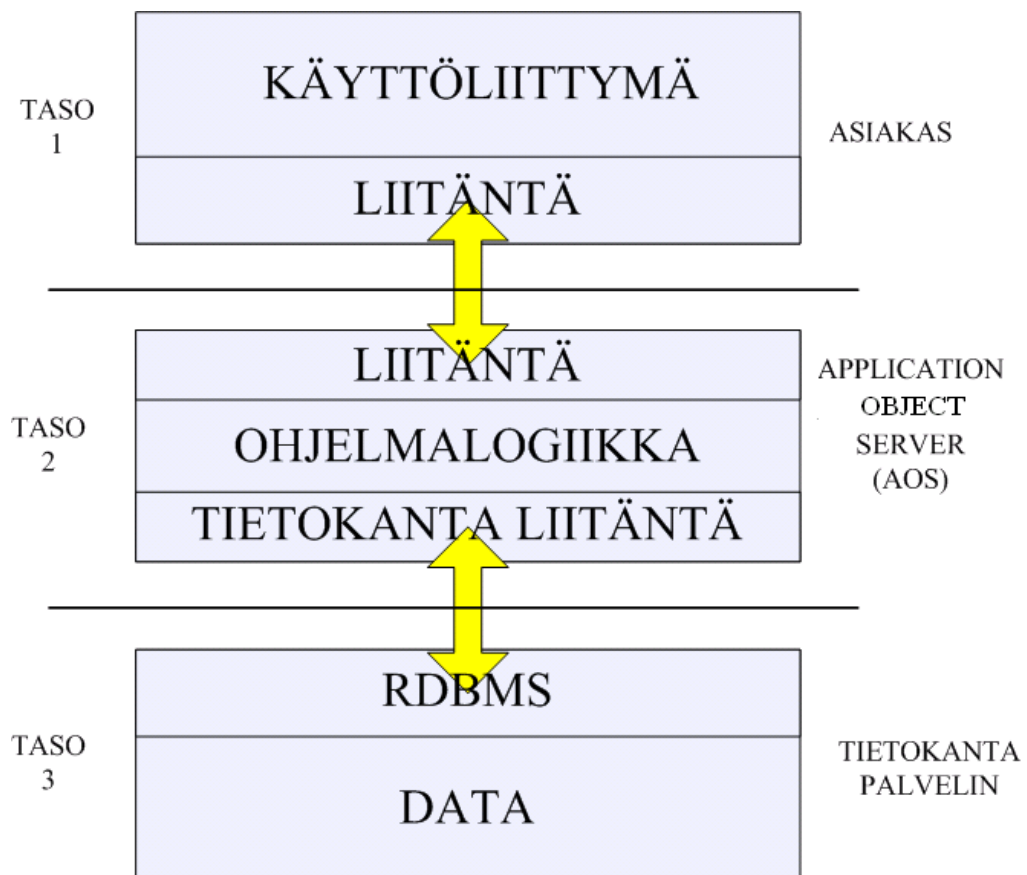


Kuva 2.2: Windows-käyttöliittymä

Kuvassa 2.2 on Windows-käyttöliittymä. Taustalla on pääikkuna, jossa on auki erään moduulin etusivu-ikkuna. Ikkunan vasemmassa laidassa on moduuliin liittyviä pikalinkkejä lomakenäytöille sekä paikkoihin. Oikealla puolella on moduulin etusivu, joka sisältää linkit kaikkiin moduulin näyttöihin, raportteihin sekä asetuksiin. Ikkunan päälle on avattu eräs lomake. Lomakkeet avautuvat AX-asiakkaassa usein uuteen ikkunaan. Lomakenäytön yläosassa on yksittäisen tietueen tietoja kuvaava ryhmänäkymä (englanniksi Group) ja alaosassa on tietuekokonaisuuden näyttämiseen suunniteltu grid-näkymä.

## 2.2. Teknologia

Dynamics AX 2009 perustuu teknisesti kolmikerrosarkkitehtuuriin, jossa alimmalla kerroksella on tietokanta, välissä toimintalogiikka ja päällimmäisenä käyttöliittymä.



**Kuva 2.3:** AX:n kolmikerrosmalli [2, AOS Overview]

Kuvassa 2.3 on esitetty Microsoft Dynamics AX käytetty kolmikerrosarkkitehtuuri pääpiirteittäin. Kuvasta on nähtävissä, että ensimmäisessä kerroksessa on älykäs asiakasohjelma, toinen kerros on Application Object Server (AOS), joka sisältää toimintalogiikan, ja kolmannessa kerroksessa on tietokantapalvelin. Dynamics AX 2009:n perusominaisuus on yhden tietokannan arkkitehtuuri, jota käytetään kaikissa organisaatioon kuuluvissa yrityksissä. [4]

Kuvan 2.3 ensimmäisessä kerroksessa oleva asiakaskerros toteuttaa käyttöliittymän. Asiakaskerros vastaa ulkoasun lisäksi myös näyttöjen ja raporttien suorittamisesta. Asiakaskerroksen toiminnot ajetaan samalla koneella kuin mihin asiakassovellus on asennettu, ja tarvittavat tiedot noudetaan AOS:sta.

Application Object Serveriä (AOS) käytetään olioiden, luokkien ja tiedon jakamiseen. AOS:n tarkoituksena on toimia datan sekä asiakasrajapinnan erottavana tekijänä (kuva 2.3), ja siinä ajetaan ohjelmistologiikkaa, kuten luokkien ja kyselyiden (queries) suorittamista. Tällä erottelulla on pyritty parantamaan ohjelman suorituskykyä sekä uudelleenkäytettävyyttä ja ylläpidettävyyttä, koska dataa ei tarvitse jokaisessa sovelluksen osassa erikseen määritellä. [1, Application Object Server]

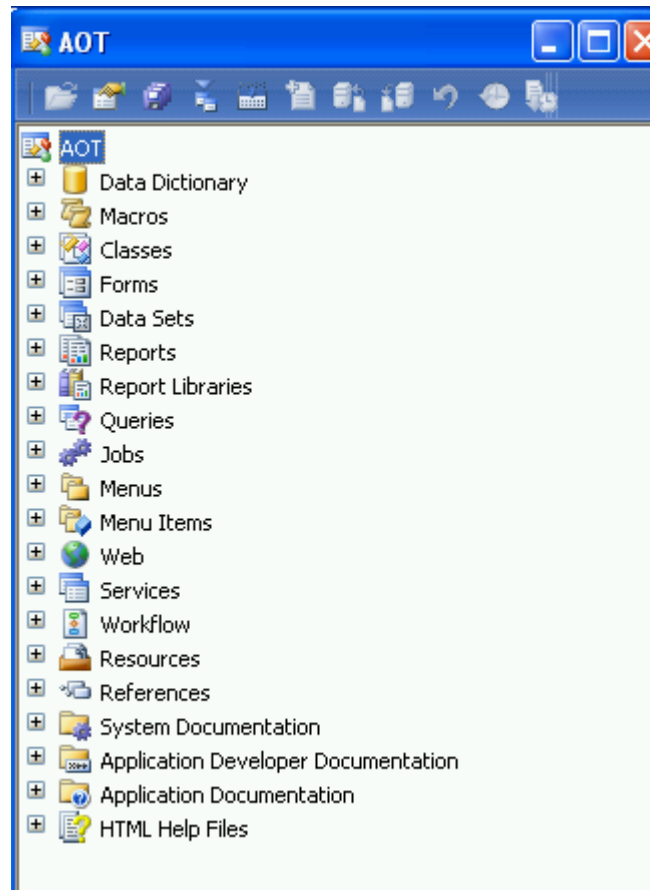
Kuvan 2.3 kolmannessa kerroksessa oleva tietokantapalvelin sisältää datan ja tietokannan taulut. Tietokantapalvelinta kutsutaan aina AOS:n kautta ja näin saadaan aikaan muun muassa tietoturvasempi ympäristö, kun AOS hoitaa kutsujen toteuttamisen, ja tietojen rajoittamisen annettujen parametrien mukaisesti.

### 2.3. Application Object Tree (AOT)

AOT on lyhenne sanoista Application Object Tree, eli sovellusobjektipuu. Se toimii AX-sovelluskehittäjälle hierarkkisenä tapana esittää AX:n sovellusobjektit ja sisältää kaikki AX:ssa käytetyt sovellusobjektit järjesteltynä aihealueittain aukeavien solmujen alle. Kunkin solmun alta löytyy, joko lisää päätasoon liittyviä alitasoja, tai solmun olioita ja näihin liittyviä toimintoja. Esimerkiksi Data Dictionary, jonka alla on muun muassa taulut, näkymät, laajennetut tietolistat, perus enumit, lisenssikoodit, asennus- (englanniksi configuration) ja turvallisuusavaimet. Tai näyttöjä ja raportteja muodostavia kokonaisuuksia, esimerkiksi Forms, jonka alle tulee lista olemassa olevista näytöistä. Ja avattaessa jonkin näytön solmu, nähdään siihen liittyvät metodit, tietolähteet ja suunnittelumallit, joiden alle on lisätty niihin liittyviä objekteja ja kokonaisuuksia.

Kun jokin objekti valitaan, voidaan siihen liittyvät ominaisuudet tuoda näkyville Properties-valikkoon. Tässä valikossa voidaan säätää objektiin liittyviä ominaisuuksia. Eri objekteilla on erilaisia ominaisuuksia, mutta yleisimpiä ja tärkeimpiä ominaisuuksia ovat: Name, eli nimi, joka yksilöi AX-kehittäjälle objektin, ja Label, eli näkyvä nimi, joka on AX-asiakkassa näytettävä kyseistä objektia kuvaava käännettävä nimitys. Koska AX on suunniteltu tukemaan useita eri kieliä, annetaan label -kenttään, kuten muillekin AX:n käyttöliittymässä näkyville teksteille, best practicen mukaisesti viittaus Label-käännöstauluun [2, Top Best Practices to Consider] .

ConfigurationKey kertoo minkä asennuksen mukana kyseinen objekti tulee käytettäviin. SecurityKey [1] määrittää millainen turvallisuusavain käyttäjällä tulee olla, jotta hän näkisi kyseisen objektin. NeededAccessLevel määrää mikä käyttöoikeustaso käyttäjällä tulee olla, jotta kyseinen objekti näkyisi hänelle.

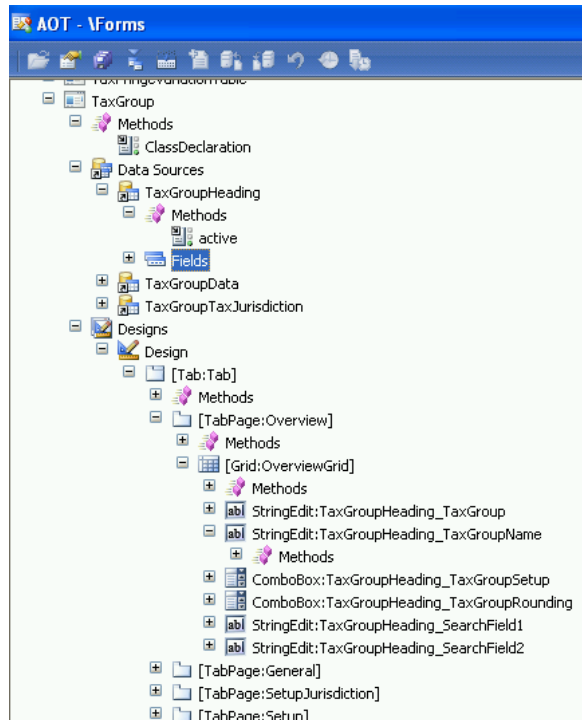


*Kuva 2.4: Application Object Tree*

Kuvassa 2.4 näkyy AOT:n rakenne. Näistä solmuista Enterprise Portal -toteutuksen kannalta olennaisimpia ovat Data Sets-solmu, sekä koko Web -solmu, jonka kaikkia alakohtia voidaan hyödyntää Dynamics AX 2009:n Enterprise Portal ratkaisussa. Navigointi luodaan käyttämällä Web Menus-solmua navigointivalikoiden luontiin ja Web Menu Items-solmua luomaan yksittäisiä navigointilinkkejä, joita lisätään Web Menus-solmun alla oleviin valikkorakenteisiin.

## 2.4. AX-ohjelmointi

AX-ohjelmoinnissa pyritään yleiskäyttöisyyteen ja uudelleen käytettävyyteen, jolloin suurin osa näytöllä näkyvistä tapahtumista suoritetaan alemmalla tasolla kuin itse näytöllä. Ohjelmalogiikkaa on myös taulutasolla, joka sisältää muun muassa uuden tietueen perustamisen, tiedon eheyden tarkistuksia, tiedon talletuksessa ja päivityksessä tapahtuvia arvojen asettamista tai tauluun kohdistuvia funktioita. Toinen yleiskäyttöisyyteen pyrkivä paikka on sijoittaa funktioita ja metodeja luokkiin, joita voidaan kutsua mistä tahansa AX:n sisältä tai Business Connectorin lävitse.



*Kuva 2.5: AX-näytön AOT-solmu*

Kuvassa 2.5 nähdään kuinka AX-asiakkaan näytöt on sijoitettu Forms-solmun alle. Jokainen näyttö koostuu kolmesta osasta: Methods, metodeista tai funktioista, jotka on sidottu kyseiseen näyttöön, Data Sources -tietolähteistä, jotka sisältävät kyseisellä näytöllä käytetyt AX:n taulut ja Designs-solmusta, joka sisältää näytön ulkoasun määrittäykset. [1]

Jokaisen kohdan alle voidaan kirjoittaa omia metodeja. Myös järjestelmän omia metodeja voidaan kuormittaa, lisäämällä omaa koodia ajettavaksi metodin yhteydessä. Tällöin voidaan esimerkiksi hakea koodilla toisiin kenttiin arvoja tai suorittaa ajonaikaista arvojen oikeellisuuden tarkistusta. [1]

## 2.5. X++

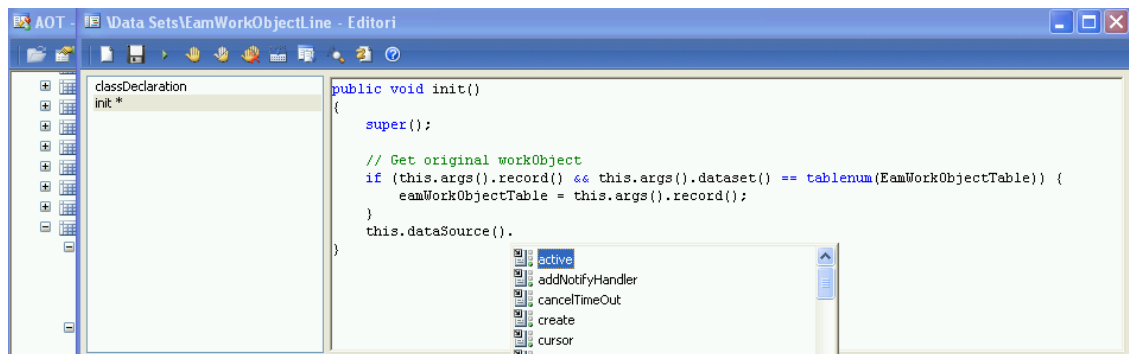
Microsoft Dynamics AX toteutuskielenä käytetään X++-nimellä kulkevaa ohjelmointikieltä. Se on kehitetty AX:n edeltäjää AXAPTA-järjestelmää varten, ja ottamaan erityisesti huomioon toiminnanohjausjärjestelmän vaatimukset. X++ on oliopohjainen kieli, joka käyttää aiemmin esitettyä layer-tekniikkaa toteutuksissa. Se muistuttaa hyvin paljon C++-, Java- ja C#-kieliä koodin syntaksin ja ulkoasun osalta.

X++-funktion erikoisuutena on, että funktion alussa esitellään funktiossa käytettävät muuttujat. Nämä muuttujat eristetään muusta ohjelmakoodista rivillä, joka sisältää pelkän puolipisteen. Puolipisteen jälkeen kirjoitetaan ohjelmalogiikka normaalisti. [1]

### Ohjelma 2.1: X++-esimerkki

```
// Esimerkkiohjelma toteuttaa raporttiajon saatujen
// argumenttien perusteella
static void Example_Job1(Args _args)
{
    Args args;
    ReportRun reportRun;
    ;
    args = new args(ReportStr(EPCustRevenue));
    reportRun = classFactory.reportRunClass(args);
    reportRun.init();
    reportRun.run();
}
```

Ohjelmassa 2.1 on esimerkki X++-koodista. Ohjelmakoodissa haetaan EPCustRevenue niminen raportti, jonka ajo voidaan käynnistää esimerkin koodilla, mistä tahansa ohjelmassa. Huomattavaa koodissa on AX:n vaatima puolipiste erottamaan muuttujien esittelyn ja ohjelmakoodi toisistaan.



**Kuva 2.6:** X++-ohjelmointiympäristö

Kuvassa 2.6 on esitetty X++-ohjelmointiympäristö. AX-ohjelmointi tapahtuu AX-asiakkaassa varsin alkeellisessa ympäristössä. Saman AOT-olion metodit aukeavat yhdelle ohjelmointinäytölle, jossa eri metodit on listattu näytön vasemmalle laidalle. Näytön oikea puoli on koodieditorin puoli, joka on käytännössä yksinkertainen tekstieditori. Sen ominaisuuksiin kuuluvat ohjelmakoodin väritys, ja taulujen tai luokkien sisäisiä metodeja ja kenttiä hakevalla ennakoiva tekstinsyöttö.

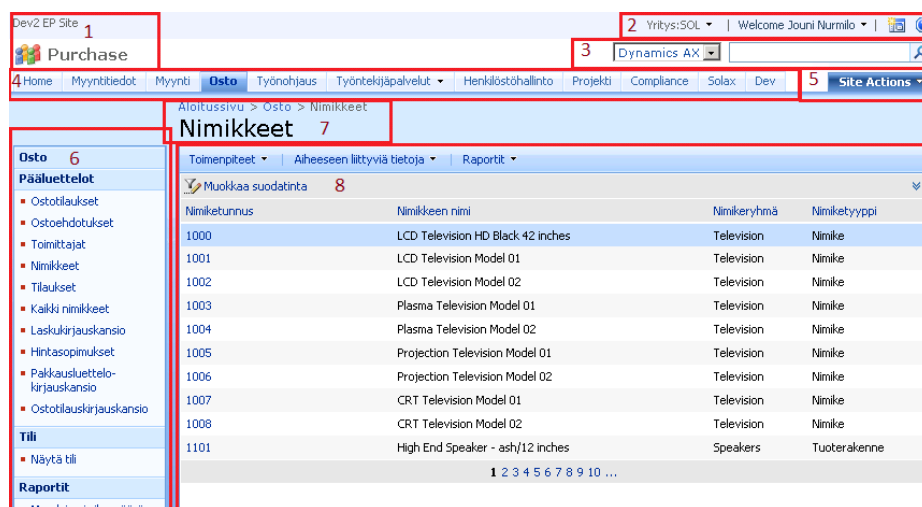
## 3. ENTERPRISE PORTAL

Enterprise Portal on Microsoft Dynamics AX 2009 järjestelmän lisäosa, joka mahdollistaa web-portaalin luomisen olemassa olevalle AX-asennukselle. Enterprise Portal toimii Microsoft SharePoint Servicen [6] tai Microsoft Office SharePoint Server (MOSS) [6] alustalla. Enterprise Portalin sivupohjat luodaan SharePointilla ja pohjiin voidaan lisätä Enterprise Portal komponentteja, jotka ovat yhteydessä Dynamics AX:n AOS-palvelimeen. AX:n kolmikerrosarkkitehtuurissa Enterprise Portal sijoittuu ensimmäiselle eli käyttöliittymän tasolle ja sen liitännä AOS-palvelimelle toimii .NET Business Connector -komponentti [1, .NET Business Connector].

### 3.1. Käyttöliittymä

Enterprise Portal on AX:n web-käyttöliittymä, joten sitä käytetään Internet selaimen kautta. Enterprise Portalin käyttö ei vaadi erillisten lisäosien asentamista koneelle, ja sitä voidaan käyttää missä tahansa, mistä on yhteys Enterprise Portal -palvelimelle. Enterprise Portalin käytön selainvaatimukseksi on ilmoitettu joko Internet Explorer 6.0 Service Pack 2 tai uudempi tai Mozilla Firefox 2.0 tai uudempi selain [5]. Enterprise Portalin käyttö onnistuu kuitenkin myös muilla selaimilla, joissa on JavaScript-tuki.

Enterprise Portal -sivut ovat rakenteellisesti hyvin samankaltaisia keskenään. Sivun yläosassa on käyttäjänasetukset, haku sekä moduulien välinen navigointi. Sivun vasemmassa reunassa on moduulikohtainen pikavalikko moduulin sisäiseen navigointiin ja valikon oikealle puolelle jäävä tila käytetään itse sivun sisällön esittämiseen eli esimerkiksi käyttäjäohjaimiin ja raportteihin.



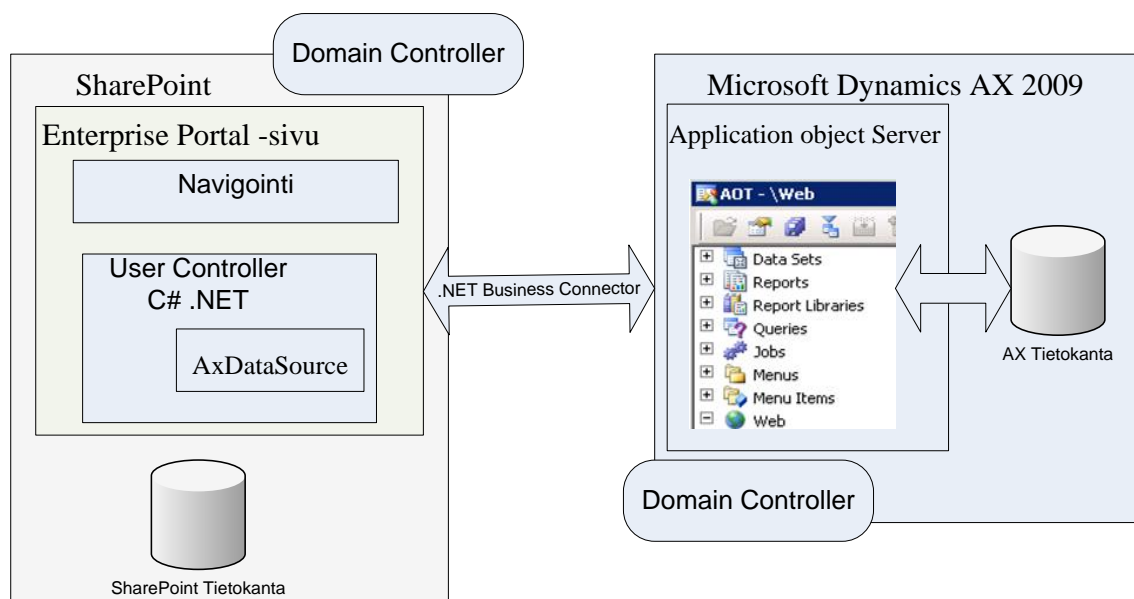
Kuva 3.1: Enterprise Portal -sivun rakenne



Kuvassa 3.1 on esitetty Enterprise Portal -sivun rakenne. Kuvaan on eroteltu suorakaiteella alueita, jotka on numeroitu. Näillä alueilla on eroteltu Enterprise Portal -sivun rakenteen osat toisistaan. Kuvan 3.1 alue 1 sisältää Enterprise Portal sivuston otsikon ylhäällä, sekä moduulin nimen alempana. Alueella 2 on käyttäjätiedot, AX:n yrityksenvalinta, käyttäjän kirjautumisvalikko ja uuden ikkunan sekä ohjeet avaavat ikonit. Alueella 3 on haku sivuston sisällä. Alue 4 sisältää moduulivalikon, josta voidaan valita mitä moduulia käytetään. Alueella 5 on valikko, jolla käyttäjä voi muokata sivuston sisältöä. Alue 6 sisältää moduulikohtaisen pikavalikon. Alueella 7 on Enterprise Portal -sivun polku sekä otsikko. Alue 8 sisältää Enterprise Portal -sivun sisällön. Kuvassa 3.1 alueella on SharePointiin rakennettu AX-kohtainen AxToolBar-web part, sekä käyttäjäohjain, jossa grid-listaus nimikkeistä.

### 3.2. Tekniikka

AX:n kolmikerrosarkkitehtuurissa Enterprise Portal sijoittuu ylimmälle käyttöliittymäkerrokselle. Se on integroitu Dynamics AX:n AOS-palvelimelle .NET Business Connector yhteydellä. Enterprise Portal toimii SharePoint-alustalla omalla palvelimellaan.



**Kuva 3.2:** Enterprise Portal -toteutukseen liittyvät teknologiat [1]

Kuvassa 3.2 on vasemmalle merkitty SharePoint-ympäristö, joka sisältää Enterprise Portal -sivut. Enterprise Portal -sivut koostuvat SharePoint-ympäristöön rakennetuista web-osista. Kuvassa 3.2 esitetystä osa-alueista navigointia voi toteuttaa useilla erilaisen navigointinäkömän luovilla web-osilla kuten Dynamics Left Navigation [2, Dynamics Left Navigation], Dynamics QuickLaunch [2, Dynamics QuickLaunch], Web Menu [2, Web Menu], ja Dynamics Toolbar [2, Dynamics Toolbar]. Nämä osat hakevat tietonsa määrätystä AX:n Web Menu Item -elementiltä.

Käyttäjöohjaimet tuodaan Enterprise Portal -sivuille Dynamics User Control web-osalla [2, User Controls Overview] ja ne ovat Enterprise Portal -kehittäjien itsensä toteuttamia Microsoft Visual Studio -kehitysympäristössä C#-kielellä ja .NET-ominaisuuksia käyttäen. Käyttäjöohjaimet sisältävät usein joko listamaisen useita tietueita sisältävän grid-elementin [2, AxGrid] tai lomakemaisen yhden tietueen tiedot sisältävän form-elementin [2, AxForm]. Käyttäjöohjaimet ovat yhteydessä AOS-palvelimen Data Set -elementteihin oman AxDataSource-elementtinsä [2, AxDataSource] kautta. Data Set -elementit tarjoavat siinä määritetyn joukon AX:n tauluja sekä metodeja käyttäjöohjaimelle käytettäväksi. Enterprise Portal -sivut voivat omien web-osiensa lisäksi sisältää myös SharePointin web-osia sekä HTML, JavaScriptiä tai muita web-toteutustekniikoita. SharePoint tallentaa luodut sivut omaan SharePoint -tietokantaansa. [1]

Enterprise Portal ja AX:n AOS-palvelin kommunikoivat keskenään käyttäen .NET Business Connector -komponenttia [1, NET Business Connector]. Lisäksi molempien ympäristöjen kirjautumisia ja käyttöoikeuksia hallinnoidaan omilla Domain controller -järjestelmillä [7]. .NET Business Connector -yhteys muistaa SharePointiin kirjautuneen käyttäjän tiedot ja käyttää AOS-palvelinta samoilla käyttäjätiedoilla. Tämä mahdollistaa tiedon ja oikeuksien tarkistamisen käyttäjakohtaisesti sekä tiedon hakemisen käyttäjöoikeuksien mukaisesti. .NET Business Connector käyttää AX:n AOS -palvelimen elementtejä, kuten AOT:n Web-solmun sisältö ja Data Sets-solmuja, jolloin AX toiminnallisuuksien toteuttaminen Enterprise Portalin kautta on mahdollista, samalla tavalla kuin ne tehtäisiin AX-asiakkaassa. [1]

### 3.3. Sivun käsittely

Enterprise Portal sivun käsittely tapahtuu seuraavasti: [1, Page Processing]

1. Käyttäjä komentaa selaimen ottamaan yhteyden Enterprise Portal sivulle.
2. Selain ottaa yhteyden Internet Information Services (IIS)-palvelimelle.
3. Tunnistustasosta riippuen IIS-palvelin tunnistaa käyttäjän.
4. IIS-tunnistuksen jälkeen Windows SharePoint Services Internet Server Application Programmin Interface (ISAPI) tarkistaa, onko käyttäjällä oikeus käyttää sivua.
5. SharePoint tunnistuksen jälkeen sivukutsu ohjataan SharePointin ASP.NET -sivunkäsittelijälle.
6. Sivunkäsittelijä hakee Web part-sivun tiedot Windows SharePoint Services sisällön tietokannasta. Nämä tiedot sisältävät mm. sivun template ID:n, käytetyt web-osat ja niiden ominaisuudet, sekä palvelimelle tallennetun sivun ulkoasun. Tämän jälkeen Windows SharePoint Services käsittelee sivun ja luo sekä alustaa sivun Web-osat.
7. Alustettaessa Enterprise Portalin Web-osat, Enterprise Portal alustaa istunnon Enterprise Portal Web frameworkilla .NET Business Connectorin läpi.

8. Web-rajapinta tarkistaa Dynamics AX tunnistuksen ja kutsuu web-rajapinnan web-käsittelijöitä, jotka käsittelevät Enterprise Portalissa käytettyjä web-osat.
9. Dynamics User Control Web Part ajaa käyttäjäohjaimen, johon se osoittaa. Käyttäjäohjain ottaa yhteyden Dynamics AX:n .NET Business Connectorin läpi ja esittää Web-osan HTML:n.
10. Web-sivu koostaa kaikki Web-osien palauttamien HTML:t ja esittää sivun käyttäjän selaimelle.
11. Enterprise Portal istunto päättyy.

### 3.4. .NET Business Connector

.NET Business Connector on osa Microsoft AX -kehitysympäristöä, jonka avulla luodaan sovelluksia, jotka integroituvat AX:n AOS-palvelimeen. Sitä käytetään muun muassa Enterprise Portalin ja Application Integration Frameworkin (AIF) [1, s.581] yhteyksissä AOS-palvelimelle. Lisäksi se mahdollistaa ulkopuolisten sovellusten tai oman käyttöliittymän tekemisen AOS-palvelimen päälle. [10, Business Connector architecture]

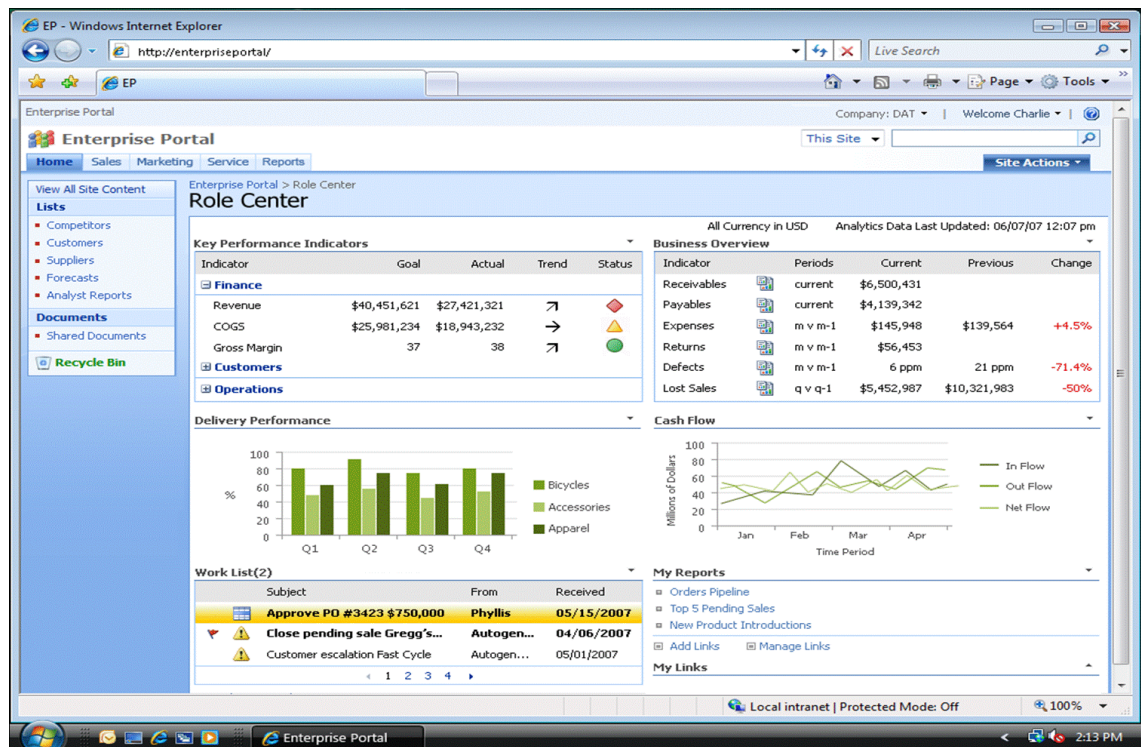
Business Connectoria käytettäessä käyttäjä ensiksi tunnistautuu Axapta.Login -metodilla AOS-palvelimelle Dynamics AX -käyttäjänä. Palvelin tarkistaa, että käyttäjällä on riittävät oikeudet käyttää Business Connectoria. Kun käyttäjän kirjautuminen on varmistettu, muodostetaan AX-käyttäjälle istunto. Tämän jälkeen .NET Connectorin kautta voidaan käsitellä AX:n dataa ja käyttää AOT:n luokkia ja X++-toimintalogiikkaa käyttäjän käyttöoikeuksien mukaisesti. [1, s. 351]

### 3.5. Roolikeskukset

Roolikeskukset ovat Enterprise Portal -tekniikan päälle luotuja SharePoint-sivuja, jotka ovat määritelty AX:n asetuksissa erityisiksi aloitussivuuksi. Roolikeskukset aukeavat AX-käyttäjille ensimmäisenä näkymänä mentäessä Enterprise Portal -sivustolle tai avattaessa AX-asiakasohjelma.

Roolikeskus sivulta käyttäjän on tarkoitus nähdä heti itseään koskevat mielenkiintoisimmat ja tärkeimmät asiat. Roolikeskukset luodaan käyttäjäryhmäkohtaisesti, ja niille on kerätty erilaisilla mittareilla, raporteilla ja pinoilla ryhmien kannalta mielenkiintoisimmat tiedot järjestelmästä. Lisäksi jokaisella käyttäjällä on mahdollisuus muokata roolikeskussivuaan oman mielensä mukaisiksi, jolloin sivut vastaavat parhaiten kyseisen käyttäjän tarpeita.

Roolikeskussivuja varten on luotu omia erillisiä Share Point web parteja, jotka toimivat pääasiallisesti tiedon visualisoimisen apuvälineinä. Osaan elementeistä voidaan liittää linkkejä AX-asiakkaan näkymiin, jolloin käyttäjä voi asiakkaan puolella elementtiä klikatessaan siirtyä suoraan kyseistä asiaa koskevaan näkymään. [1, Role Centers]



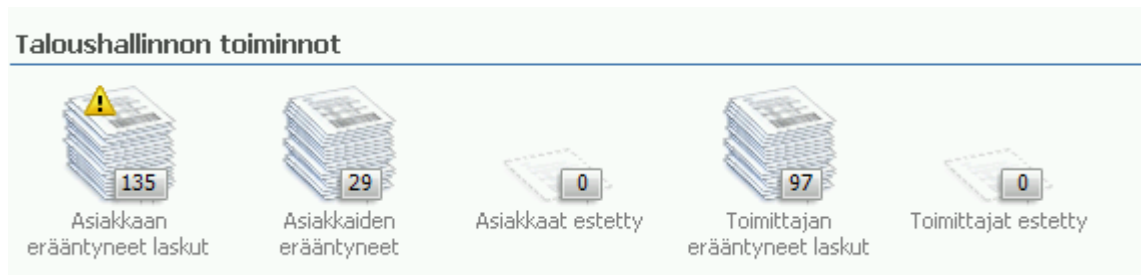
*Kuva 3.3: Roolikeskusnäky*

Kuvassa 3.3 on kuvakaappaus yhdestä roolikeskusnäkyästä. Kuvasta on nähtävissä roolikeskusten yleinen ilme. Roolikeskuksesta on näkyvissä kuusi elementtiä, joista ylimmäiset kaksi elementtiä esittävät AX-tietokannasta kerättyjä tietoja Business Overview -elementin näkymissä, seuraavat kaksi elementtiä ovat visuaalisia raporttinäkymiä. Alhaalla vasemmalla on AX:n työlistaan perustuva elementti ja alhaalla oikealla on pikalinkkinäkymä, johon on kerätty käyttäjän omia raportteja.

Roolikeskuksia varten on Enterprise Portaliin luotu useita erilaisia web part-osaia, jotka ovat yhteydessä AX-järjestelmän eri osiin. Osa web parteista on yhteydessä raportointipalvelimeen, toiset taas AX: kantaan. Osa web parteista käyttää vain omien taulujensa tietoja, kun taas toiset hakevat tietoa kaikista AX:n tauluista.

### 3.5.1. Pinot

Pino- eli cue-elementti, on visuaalinen tapa esittää tietyllä haulla saatavaa määrää tietueista AX:n tietokannassa. Tällaisella pinolla voidaan kuvata esimerkiksi hyväksyttävien laskujen määrää, suoritettava tilassa olevia töitä tai vanhentuneet myyntitarjouksia.



**Kuva 3.4:** Pinot ovat visuaalinen tapa kuvata dataa.

Kuvasta 3.4 käy ilmi, että pinot kuvaavat käyttäjän työpöydälle kertyvää pinoa töistä. Näissä pinoissa on ilmoitettu numeerisesti töiden määrä. Lisäksi pinoille voidaan asettaa jokin tietty raja-ehto, jonka täytyttyä pinossa näkyy varoitusmerkki rajaehdon täyttymisen merkiksi. [1, Role center Cue]

Pelkän visuaalisen tiedon lisäksi pinoihin voidaan liittää linkki näyttöön, joka liittyy tietueisiin, joiden määrää pino kuvastaa. Tämä linkki luodaan automaattisesti siihen näkymään, josta pinon kysely on tehty. Mikäli kyseiseen AX:n Menu Item -elementtiin on lisätty ”WebMenuItem” -arvo, luodaan linkki myös Enterprise Portal puolella, WebMenuItem:ssa määritettyyn sivuun. [1, Role center Cue]

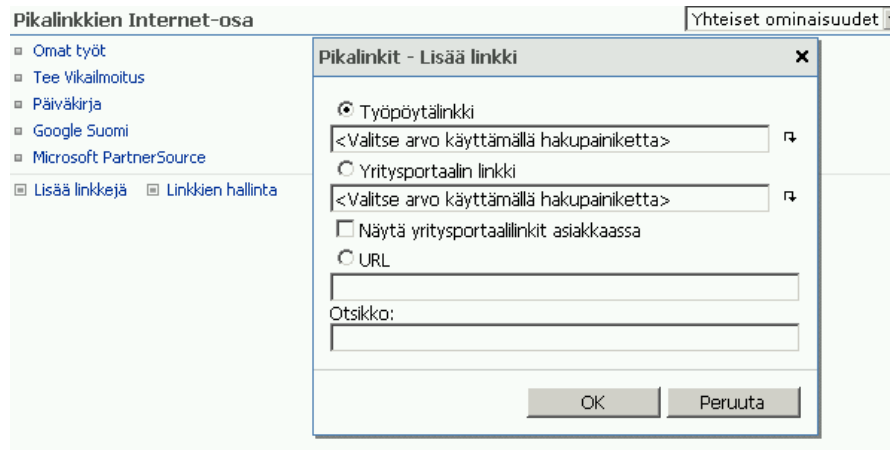
Pinoja voidaan luoda AX-asiakaan miltä tahansa listanäytöltä. Pinon luominen tapahtuu luomalla näytöllä suodatus ja sitten tallentamalla kyseinen suodatus pinoksi. Tämän jälkeen suodatusta voidaan käyttää Enterprise Portal:n pino elementissä, kuten mitä tahansa pinoa. [1, Role center Cue]

Pinoja voidaan muokata AX-asiakaan ikkunassa, joka löytyy Perustiedot/Asetukset/Roolikeskus/Muokkaa pinoja -linkin takaa. Tässä ikkunassa nähdään kaikki AX:ssa käytössä olevat pinot, sekä voidaan muokata niiden rajausta. Tätä kautta voidaan myös tuoda/viedä pinoja tiedostoihin tai AOT:n myöhempää käyttöä varten. AOT:n viedyt pinot löytyvät Resources solmun alta tunnuksella Cue\_<pinon nimi>.

Pinojen tiedot tallennetaan AX:n tietokannassa kahteen tauluun: CueQuerySecurity ja CuesQuery. CuesQuery-tauluun tallennetaan pinon tiedot, kuten hakujono ja hälytyksen raja-arvot. CueQuerySecurity-tauluun on tallennettuna mihin profiileihin pino liittyy eli minkä profiilin jäsenet voivat nähdä kyseisen pinon.

### 3.5.2. Pikalinkit

Pikalinkkien web part on tarkoitettu linkkien näyttämiseen. Siinä esitetään pääkäyttäjien määrittämiä yleisiä linkkejä, sekä käyttäjän itsensä lisäämiä henkilökohtaisia suosikkilinkkejään. Elementtiin on mahdollista tehdä kolmenlaisia linkkejä: Normaaleja web-linkkejä osoitteeseen, linkkejä Enterprise Portalin sivuihin sekä linkkejä AX-asiakaan näyttöihin. Pikalinkit web part on esitetty kuvassa 3.5.



*Kuva 3.5: Pikalinkkien lisäysnäkyessä on mahdollista valita minkä tyyppinen linkki lisätään.*

Linkkien näkyminen riippuu käytetystä alustasta. AX:n asiakkaassa näytettävässä roolikeskuksessa näkyvät AX-asiakkaan linkit, web-linkit sekä erikseen määritellyt Enterprise Portal linkit. Enterprise Portalin puolella linkkistasta on piilotettu AX-asiakkaan linkit, sillä asiakkaan linkit eivät toimi selaimen kautta, vaikka samalle koneelle olisi asennettuna AX-asiakas.

Pikalinkit on tallennettuna AX:n tietokantaan SysQuickLinks- ja SysQuickLinksOrder-tauluihin. Näistä SysQuickLinks-tauluun tallennetaan yhteiset ja yleiset linkit. Taulussa on UserId, kenttä jolla kuvataan minkä käyttäjän linkistä on kyse. Mikäli kyseinen kenttä on tyhjä, on kyseessä yhteinen linkki. Linkeistä määritellään myös mihinkä ryhmään ne kuuluvat ja linkin osoite sekä otsikko. SysQuickLinksOrder taulu taas sisältää linkkien ryhmäkohtaisen järjestyksen. Tässäkin taulussa on UserId-kenttä, jolla kuvataan mikäli kyse on käyttäjäkohtaisesta järjestyksestä.

### 3.5.3. Business overview

Kuvassa 3.6 on esitetty Business overview-elementti. Elementin tarkoituksena on tarjota nopea katsaus yrityksen avaintietoihin ja vertailla näitä tietoja ennalta annettuihin rajoihin. Näitä tietoja voidaan esitellä kahdella erilaisella tavalla: KPI-lista (Key Performance indicators), joka on kuvassa 3.6 vasemmalla puolella, toinen tapa esittää tietoja on Business Overview, joka on kuvassa 3.6 oikealla puolella.

Key Performance Indicators					Business Overview				
Indicator	Goal	Actual	Trend	Status	Indicator	Periods	Current	Previous	Change
<b>Finance</b>					Receivables	current	\$6,500,431		
Revenue	\$40,451,621	\$27,421,321	↗	🔴	Payables	current	\$4,139,342		
COGS	\$25,981,234	\$18,943,232	→	🟡	Expenses	m v m-1	\$145,948	\$139,564	+4.5%
Gross Margin	37	38	↗	🟢	Returns	m v m-1	\$56,453		
<b>Customers</b>					Defects	m v m-1	6 ppm	21 ppm	-71.4%
<b>Operations</b>					Lost Sales	q v q-1	\$5,452,987	\$10,321,983	-50%

*Kuva 3.6: Business Overview elementti näyttää avaintietoja.*

Business overview-elementit käyttävät Analysis Servicen kuutioita [1], jotka sijaitsevat tietokantapalvelimella. Business Overview kuutioita ja mittaustapoja luodaan Business Intelligence Development Studiolla, joka toimitetaan Analysis Servicen mukana. Koska näiden mittareiden luominen tapahtuu erillisellä ohjelmalla, ja on varsin monimutkaista, ei niihin perehdytä tarkemmin tässä työssä.

### 3.5.4. Työluettelo

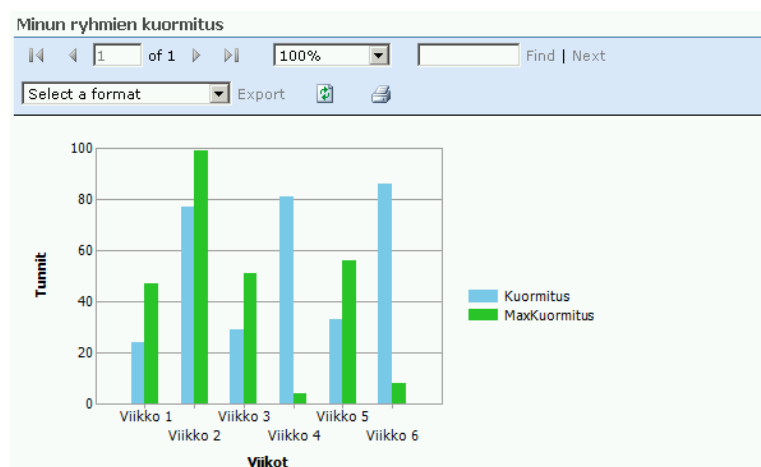
Kuvassa 3.7 on esitetty työluettelo-elementti (Unified Worklist). Elementin tarkoituksena on näyttää käyttäjälle hänelle tehtäväkseen osoitetut työt ja varoitukset. Elementin näyttämät tiedot perustuvat AX:ssa määritettyihin asetuksiin. Ne kertovat mitkä työt käyttäjälle näytetään tai minkä raja-arvon jälkeen käyttäjälle näytetään varoitus.

Työluettelo							
Näytä		Ryhmittelyperuste					
!	Aihe	Tietuetyyppi	Tunnus	Eräpäivä	Alkaen	Luontipäivämäärä	Yritys
	Workki	Työtilaus	Työn Id: APP007135,	7.6.2010 10:51:05		7.6.2010 10:52:13	sol
	Workki	Työtilaus	Työn Id: APP007060,	4.6.2010 11:26:31		4.6.2010 11:26:52	sol
	Workki	Työtilaus	Työn Id: APP007042,	4.6.2010 9:31:58		4.6.2010 9:34:31	sol
	Workki	Työtilaus	Työn Id: APP007039,	4.6.2010 17:07:34		3.6.2010 17:08:03	sol
	Workki	Työtilaus	Työn Id: APP006910,	3.6.2010 10:49:08		3.6.2010 10:49:34	sol

*Kuva 3.7: Työluettelo*

### 3.5.5. Raportit

Kuvassa 3.8 on esitetty Dynamics Report Server Report -web partilla esitetty raportti roolikeskuksessa. Tämä web partin tarkoituksena on esittää roolikeskuksessa Microsoft SQL Server Reporting Servicen raportteja. Nämä raportit voivat olla joko graafisia kuten pylväs- tai piirakkadiagrammeja, tai pelkkää tekstiä sisältäviä yleiskatsauksia tai syväanalyysiä toiminnasta.



*Kuva 3.8: Dynamics Report Server -raportti*

Roolikeskuksissa näytettävien raporttien pitää olla Reporting Servicen raportteja, ja niille tulee olla määriteltynä OLAP tai OLTP yhteys datan saamiseen. Raportit luodaan Dynamics AX:n raportin kehitystyökaluilla ja niihin viitataan AOT:n Menu Item-linkillä, joka on SQLReportLibraryReport muotoinen.

Raporteille voidaan antaa joko kiinteät parametriarvot tai käyttäjälle voidaan tarjota mahdollisuus syöttää itse parametrit, joiden perusteella raportti generoidaan. Lisäksi parametrejä voidaan välittää raportille post-metodilla URI:ssä muodossa <raportin\_nimi>.Parameters.<parametrinNimi>=<parametrinArvo>.

### **3.6. Microsoft SharePoint Services**

SharePoint on kokoelma tuotteita ja teknologioita, jotka auttavat käyttäjiä luomaan, organisoimaan, levittämään ja ylläpitämään web-pohjaisesti tallennettua tietämystä. Se tarjoaa standardin sivupohjan sivun esitystavaksi ja tukee näin helposti omaksuttavan ja yhtenäisen ulkoasun sekä navigoinnin läpi sivuston. SharePointia käytetäänkin usein yrityksen ryhmäkohtaisten sivujen tai intra- ja extranet-ratkaisujen toteuttamiseen.

SharePoint toimii kahden hieman toisistaan poikkeavan alustan päällä: Windows SharePoint Services (WSS) ja Microsoft Office SharePoint Server (MOSS). Näistä WSS käytetään luotaessa pieniä sivustoja ryhmien väliseen ja yrityksen sisäiseen viestintään, kun taas MOSS laajentaa WSS:ä mahdollisuuksilla laajempaa portaalimaiseen yritystason julkaisemiseen, yrityshakuun, bisnes prosessien automatisointiin sekä BI-tietojen raportointiin ja analysointiin. [7]

Microsoft Sharepoint on Enterprise Portalin toiminta-alusta. Enterprise Portal -sivusto on myös täydellinen SharePoint -sivusto. SharePointia käytetään luotaessa Enterprise Portal -sivuja ja määriteltäessä Enterprise Portal -sivujen ulkoasua sekä lisättäessä sivuille elementtejä. [1, s.17]



## 4. ENTERPRISE PORTAL -SIVUN LUONTI

Enterprise Portalin kehitys tapahtuu kolmessa eri ympäristössä: AX:n omassa ohjelmointiympäristössä, Visual Studiossa sekä SharePointissa. AX:n puolella luodaan muun muassa ohjainten käyttämät rajaukset, taulut, navigointi ja sivustohierarkia. Myös sitä käytetään kaikkien Enterprise Portal elementteihin liittyvien metadatan ja datan tallennuspaikkana. Visual Studiossa ohjelmoidaan Enterprise Portalin esitystason komponentit C#-kielellä .NET-tekniikkaa hyväksi käyttäen. SharePoint ympäristö taas toimii Enterprise Portalin web-sivujen luontialustana. [1], Enterprise Portal Development Tools]

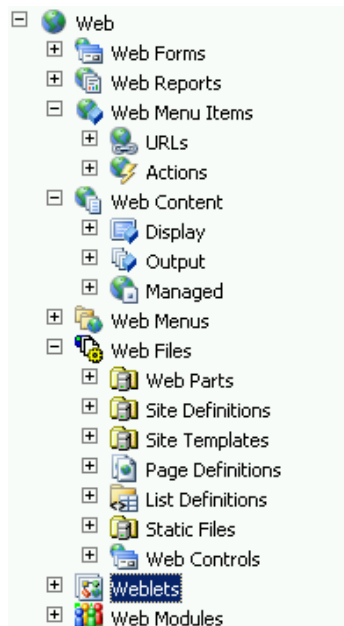
### 4.1. AX

Käyttäjähajaimet perustuvat AX:n tietokannasta saatujen tietojen esittämiseen AX:n metatietojen perusteella, joten AX:ssa tapahtuva kehitystyö on merkittävässä osassa Enterprise Portalin käyttäjähajaimien kehitystä. AX:n tietokanta sisältää Enterprise Portalin kannalta paljon muutakin merkittävää tietoa kuin puhdas data, ja niihin tietoihin syvennyttään tarkemmin tässä luvussa.

#### 4.1.1. Web-solmu

AOT:n Web-solmun alle on keskitetysti tallennettu kaikki Enterprise Portalin tarvitsema metatieto, kuten sivuston hierarkkinen rakenne, käyttäjähajainten vaatimat oikeudet tai otsikkotiedot. Lisäksi Web Files -solmun alle on tallennettu käyttäjähajainten, SharePoint-sivujen ja SharePoint-sivuilla käytetyt muut elementit Enterprise Portalin hallittua asennusta varten.

Kuvassa 4.1 on esitetty AOT:n web-solmu ja sen alasolmu. Näiden kahdeksan alasolmun alla on kaikki pelkästään web-käyttöliittymään liittyvä tieto. Web Forms, Web Reports ja Weblents solmut sisältävät AX 4.0:n Enterprise Portalissa käytettyjen ohjainten ja raporttien koodit. Näitä ominaisuuksia voidaan käyttää edelleen AX:n 2009 versiossa, mutta seuraavasta AX:n versiosta ne on poistettu, täten niiden ohjelmointiin ei syvennyttä tarkemmin. Näiden komponenttien ohjelmointi on hyvin samanlaista, kuin AX:n asiakkaan näyttöjen tai raporttien tekeminen ja se tapahtuu täysin AX:n omassa ympäristössä. [1]



**Kuva 4.1:** AOT:n Web-solmu ja sen alaselmät

Web Menu Items -solmun alle luodaan Enterprise Portalissa käytetyt linkit. Linkit voivat olla joko URL-tyyppisiä, jolloin ne viittaavat toiseen sivuun, tai Action-tyyppisiä, jolloin ne viittavat AX:n luokkakutsuun. Web Menus -solmun alle on luotu hierarkkinen kokoelma linkkivalikkoja, jotka koostuvat Web Menu Item -tyyppisistä olioista. Web Menu Items- ja Web Menus -solmujen käyttöön perehdytään tarkemmin alakohdassa 4.1.6. [1]

Web Content-solmun alle lisätään käyttäjäohjaimen, Web Form-, Web Report- ja raportti-elementtien metadata määritykset. Käyttäjäohjaimet lisätään Managed-solmun alle Visual Studioon ”Add to AOT” -toiminolla. Managed-solmusta voidaan käyttäjäohjaimen ominaisuuksia muuttamalla asettaa mm. käyttäjäohjaimen vaatima turvallisuustaso sekä käyttäjäohjaimesta näytettävä otsikko SharePointissa.

Web Files -solmun alle on koottu Enterprise Portalissa käytetyt tiedostot. Solmun tarkoituksena on toimia säiliönä elementeillä, josta ne haetaan Enterprise Portalin asennuksen tai päivityksen yhteydessä. Solmuista ”Page Definitions”-solmun alle on koottu SharePoint-sivujen lähdekoodi, Web Controls-solmun alla on käyttäjäohjainten lähdekoodi ja Static Files sisältää mitä tahansa Enterprise Portalissa käytettyjä tiedostoja. Alisolmuista ainoastaan Page Definitions -oliot sisältävät SharePointissa näytettävää metadataa.

#### **4.1.2. Data Set**

Data Set -solmu AOT:ssa tarjoaa geneerisen datayhteyskerroksen, joka mahdollistaa ulkoisten esityskerrosten pääsyn Microsoft Dynamics AX:n tauluihin ja tietotyyppeihin. Enterprise Portal käyttää Data Settejä ensisijaisena tapanaan käsitellä AX:n dataa. [3]

Data Set on kokoelma tietokannan tietoa. Sillä voidaan rajata näkyviin vain tietyt tietokannan taulut ja näkymät. Data Setillä saadaan siis näkyviin vain valikoitu

tietokannan taulu tai joukko tauluja. Data Setin taulujen näyttäminen tukee myös taulujen liittämistä toisiinsa erilaisin join-menetelmin. [1, s. 250]

Data Setit yhdistävät AX-suunnittelijalle näytöistä [2, forms] tutut tavat tiedon ja ohjelman mallintamiseen, sekä ASP.NET:n tiedon sitomismenettelyyn (englanniksi data binding). Lisäksi Data Setit tarjoavat laajennetun mahdollisuuden tiedon varmistamiseen, sekä käsittelyyn X++ -ohjelmointikielen avulla. [1, s. 250]

AxDataSource -komponentti [2, AxDataSource] käyttää käyttäjäohjaimessa Data Settejä tiedon näyttämiseen, sekä manipuloimiseen, ja tarjoaa tätä kautta pääsyn kaikille ASP.NET elementeille, jotka tukevat tiedon sitomismenettelyä. Tämä mahdollistaa AX:n tietokannan taulujen ja tietojen käytön kaikissa sovelluksissa. [1, s.250]

.NET Business Connector [1, s. 349] käyttää tietokannan tietoa käyttäjän tunnistautumistietojen mukaan, joka mahdollistaa tiedon käsittelyn kuten normaalissa AX-asiakasohjelmassa. Tämä mahdollistaa tiedon automaattisen rajaamisen AX:n määritettyjen käyttöoikeuksien ja ehtojen mukaisesti, eikä käyttäjä voi ylittää valtuuksiaan ja nähdä tietoa, minkä pääsyyn hänellä ei ole oikeuksia.

### 4.1.3. Tiedon hakeminen

Tiedon hakemista AX:n kannasta varten tulee käyttää Data Set -elementtejä. Mikäli AX:ssa ei ole valmiina sopivaa Data Settiä, tulee sellainen luoda AOT:n Data Set-solmun alle. Data Setin luominen tapahtuu lisäämällä uusi Data Set solmun alle ja nimeämällä se ominaisuudet välilehdellä. [1, s. 250]

Data Settiin voidaan lisätä Data Sources -välilehden alle halutut taulut. Jokaiselle Data sourcelle voidaan valita pudotusvalikosta haluttu taulu, tai näkymä. Valittu taulu tulee nimetä best practicen mukaan samalla nimellä kuin siinä käytetty taulu on nimetty. Lisäksi käytettävä indeksi voidaan valita pudotusvalikosta. Tauluja voidaan valita useampia yhden Data Setin alle, ja ne voidaan liittää toisiinsa yhden suhde yhteen tai yhden suhde moneen -liitännöillä [2, Data Set Views]. Kahden tai useamman taulun yhteenliittymät luovat Data Settiin yksittäisen näkymän, joka koostuu liitetyistä tauluista liitosehtojen mukaisesti. [1, s. 250]

Data Sourcen käyttöoikeuksia ja mahdollisuuksia voidaan rajata ominaisuudet valikosta. Mahdollisia rajattavia oikeuksia ovat tiedon luominen, muokkaaminen ja poistaminen. Nämä voidaan määritellä päälle tai pois päältä AllowCreate-, AllowEdit- ja AllowDelete-parametreistä, joissa true-arvo tarkoittaa, että kyseinen toiminto on mahdollista ja false-arvo tarkoittaa että kyseinen toiminto ei ole tämän Data Setin kautta mahdollista tehdä taululle. Oikeuksia voidaan antaa myös kenttäkohtaisesti, jolloin voidaan rajata tiettyjä taulun kenttiä näkymättömiin tai rajoittaa näihin vain lukuoikeus. Tämä on hyödyllistä etenkin silloin, jos joitain taulun kenttiä ei haluta käyttäjän ylläpitävän, tai niihin haetaan tiedot automaattisesti tietuetta tallennettaessa. [1, s. 250]

Haetusta Data Setin näkymästä on aina tarjolla kaksi erilaista näkymää, joista toinen näkymä sisältää kaikki haetut rivit, ja toinen näkymä käsittää ainoastaan aktiivisen rivin. Kaikki rivit käsittävää näkymää käytetään käyttäjäohjaimen useita rivejä käsittelevissä elementeissä esimerkiksi AxGridView:ssä [2, AxGridView]. Kaikki rivit käsittävä

näkymäkään ei hae välttämättä kaikkia tietokannan rivejä, vaan tehokkuuden parantamiseksi sille välitetään sivutusparametri, joka rajaa haettavien rivien määrän asetettuun sivutus arvoon [2, Accessing data].

Aktiivisen rivin näkymä hakee vain yhden tietokannan rivin, joka haun hetkellä on ajankohtainen. Tämä näkymä erotetaan kaikki rivit käsittävistä näkymästä lisäämällä sen nimen loppuun `_Current`-lisän. Tätä tiedon esitystapaa käytetään näytettäessä tai käsiteltäessä yksittäisen rivin dataa, esimerkiksi `AxForm` -elementissä [1, `AxForm`].

#### 4.1.4. Metodikutsut

Data Set -elementin tarjoamaa dataa voidaan käsitellä tai rajata lisäämällä siihen metodeja. Metodit voivat olla täysin käyttäjän omia funktioita, joita kutsutaan Data Set-elementin lävitse käyttäjäohjaimesta, tai rajauksen voi toteuttaa kuormittamalla Data Set -elementin valmiita metodikutsuja. Kuormittamalla metodeja voidaan esimerkiksi tiedonrajaus toteuttaa jonkin kentän tietojen mukaan heti, kun kyseinen Data Set ladataan, tai siitä voidaan piilottaa tai tuoda esille tilanteen mukaan tarpeellisia kenttiä. [1, s. 250]

Oman metodikutsun Data Set-oliolle voi luoda valitsemalla haluttu Data source, tai koko Data Set-solmu, ja luomalla sille Method-välilehden alle uusi metodi. Metodikutsu voi sisältää mitä tahansa X++-koodia, kuten myös esimerkiksi normaaleille AX:n näytöille luodut metodikutsut. Myös kuormitettavat metodit luodaan kuten omat metodit, mutta niille tulee antaa kuormitettavan metodin kutsu. [1, s. 250]

Yksi tärkeimpiä kuormitettavia metodeja on `init`-metodi, jota kutsutaan kun Data Set-elementin luonti `new`-metodilla on valmis, ja sillä luodaan ajonaikainen kuva Data Set-elementistä. `Init`-metodissa voidaan rajata dataa lisäämällä raja-arvoja, tarkistaa sille välitetyjä argumentteja ja parametreja tai alustaa ja valmistella muuttujia tai query-ajoja [2, Database Queries]. `Run`-metodia kutsutaan heti `init`-metodin jälkeen, missä voidaan määrittää yksittäisten kenttien näkyvyys, sekä muokattavuutta tai muokata luotua query-ajoa. `Pack`-metodi, jota kutsutaan `run`-metodin jälkeen. Sitä voidaan käyttää yhdessä `unpack` -metodin kanssa tallentamaan objektin tilaa. Tätä metodia voidaan käyttää säilyttämään muuttujat Data Set -elementillä `postback`-kutsujen välillä. `Unpack`-metodia käytetään, mikäli Data Set -elementti on aiemmin pakattu `pack`-metodilla. Tällöin ei kutsuta `init`- tai `run`-metodeja vaan ainoastaan `unpack`-metodia kutsutaan. Muita mainittavia metodeja ovat `validateWrite`, jolla varmistetaan uuden tai muokatun rivin tietojen oikeellisuus, sekä `validateDelete` jolla varmistetaan että poistettava rivi voidaan poistaa. [1, s. 250] Muut kuormitettavat metodit voi tarkistaa Microsoft Dynamics AX:n SDK:sta `form`- solmun alta ”Methods on a Form Data Source” kohdasta [2, Accessing Data].

Kuormitettaessa olemassa olevia metodeja on lähes aina tarpeellista säilyttää myös alkuperäinen metodin toteutus ja lisätä siihen vain pieni pala omaa koodia. Sen sijaan että alkuperäistä koodia lähdetäisiin etsimään AOT:sta ja kopioimaan omaan kuormittavaan toteutukseensa, voidaan käyttää AX:n `super()` komentoa, joka viittaa systeemi luokaan, joka sisältää käytetyn metodin ja tätä kutsuttaessa suoritetaan

systemiluokan vastaava metodi, jonka jälkeen tai jota ennen voidaan suorittaa omaa X++-koodia [2].

#### 4.1.5. Tiedon rajaaminen

Data Set -elementin välittämää tietoa voidaan rajata AX:ssa käyttämällä siihen metodeja. Tiedon rajaaminen voidaan tehdä aina Data Set -elementin latauksen yhteydessä kuormittamalla init-metodin tai myöhemmin tehtävillä kutsuilla, joita varten voidaan luoda omat metodit.

Rajausta voidaan tehdä lisäämällä arvovälejä, käyttämällä QueryBuildDataSource-luokkaa [2, QueryBuildDataSource], joka mahdollistaa Query-hakujonojen [1] rakentamisen X++ -koodissa ja lisäämällä siihen QueryBuildRange-luokan olio [2], jolla voidaan antaa arvoväli haettaville riveille metodikutsussa.

#### *Ohjelma 4.1: Esimerkki ohjelmakoodi tiedon rajaamisen. [2]*

```
void MyTrips()
{
    QueryBuildDataSource qbds;
    QueryBuildRange qbr;
    ;
    // Haetaan Data Source
    qbds = FMTrips_q.dataSourceName(FMTrips_ds.name());

    // Poistetaan olemassa olevat arvovälit
    qbds.clearRange(fieldnum(FMTrips, EmpId));

    // Lisätään uusi arvoväli näyttämään to show the current
    // employee's trips.
    qbr = qbds.addRange(fieldnum(FMTrips, EmpId));
    qbr.value(SysQueryRangeUtil::currentEmployeeId());
    qbr.status(RangeStatus::Open);
    FMTrips_ds.executeQuery();
}
```

Arvovälin lisääminen Data Source -tietolähteelle on esitetty ohjelmassa 4.1. Ohjelmassa lisätään ensiksi käytettäviä luokkia vastaavat muuttujat metodin käyttöön. Tämän jälkeen QueryBuildDataSource-luokalle määritetään käytettävän Data Source:n nimi. Seuraavaksi tyhjennetään käytettävässä kentässä jo mahdollisesti olevat arvovälit. Tämän jälkeen luodaan QueryBuildRange-luokkaan uusi arvoväli ja annetaan sille käytettävä kenttä. Tämän jälkeen lisätään value -kenttään haluttu arvo, jota käytetään arvovälin asettamiseen sekä status-kenttään arvovälin näkyvyyden asettaminen. Lopuksi suoritetaan luotu kysely executeQuery-komennolla, joka rajaa Data Source:n rivit asetetun arvovälin mukaisesti. [2]

Asetettaessa käytetyn arvovälin näkyvyyttä voidaan näkyvyyttä rajata RangeStatus-arvolistalla. RangeStatus-arvolistassa on kolme arvoa: Open; jolloin asetettu arvovälin

voi nähdä sekä sitä voi muokata. Locked; jolloin arvovälin voi nähdä, mutta sitä ei voi muokata. Sekä Hidden; jolloin arvoväli on piilotettu, eikä sitä voi muokata. [2]

#### 4.1.6. Navigointi

Enterprise Portal sivujen linkkivalikot ja linkit luodaan ensisijaisesti AOT:n Web Menu ja Web Menu Items solmujen alle. Tähän luotuja valikkoja on mahdollista käyttää Enterprise Portal käyttäjäohjaimissa, AxToolBar-elementeissä sekä SharePointin AX-linkkilistoissa. Linkkien ja linkkivalikoiden ylläpitämisen AX:ssa on tarkoitus mahdollistaa valikoiden helppo monikäyttöisyys sekä ylläpidettävyys.

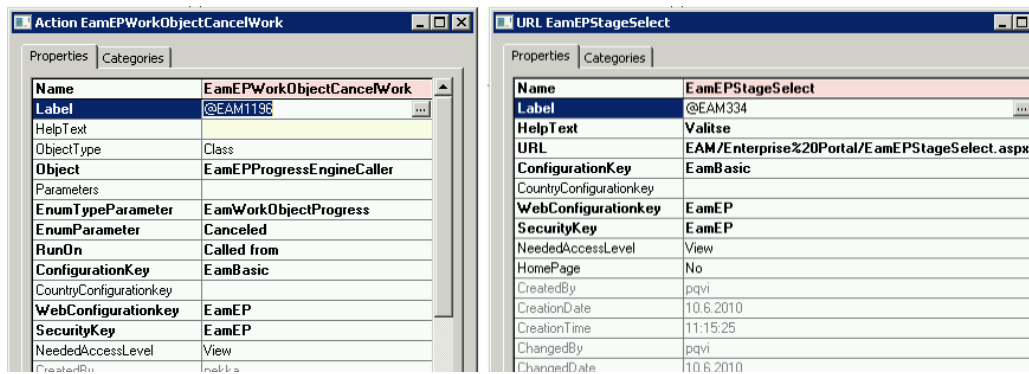
Web Menu- ja Web Menu Items -alasolmut muodostavat kokonaisuuden, joka mahdollistaa sivuston navigoinninluomisen ja linkkien tekemisen. Web Menus-solmuun luodaan Enterprise Portalissa käytettävät navigointivalikot. Valikot koostuvat pääsolmusta, joka toimii valikon hierarkkisena juurena, minkä alle luodaan valikon rakennetta. Juuren alle voidaan lisätä joko uusia alivalikoita tai valikon vaihtoehtoja, jotka toimivat linkkeinä sivuille tai luokkakutuihin. Valikon vaihtoehdot ovat Web Menu Item -tyyppisiä komponentteja, mitkä tulee olla määriteltyinä ennen kuin niitä voidaan liittää valikoihin. Web Menu -komponentteja voidaan käyttää ohjaimissa, jotka ovat tarkoitettu linkkikokoelmien näyttämiseen.



*Kuva 4.2: Web menun rakenne ja sen näkyminen AxToolBar-elementissä.*

Kuvassa 4.2 on esitetty AOT:n Web Menu -solmusta eräs valikkorakenne vasemmalla puolella ja oikealla puolella näkee, miltä kyseinen valikkorakenne näyttää AxToolBar-elementissä käytettynä. Alavalikot, jotka muodostavat AOT:ssa uuden solmun web menu -puussa, näkyvät AxToolBar-objektissa alavetovalikkoina, ja valikon vaihtoehdot näkyvät alavetovalikossa linkkeinä.

Web Menu Item -solmun alle on koottu valikon vaihtoehdot. Niitä voi olla kahta tyyppiä joko URL, joka viittaa internet-osoitteeseen, tai action, joka viittaa AX:n luokkakutsuun tai ohjelmapinoon (AOT:ssa Job [1]). Web Menu Itemiä on mahdollista käyttää linkin generoimiseen missä tahansa käyttäjäohjaimen elementissä WebLink-luokkaa hyväksi käyttäen.



**Kuva 4.3:** vasemmalla Action tyyppisen web menu item:n ominaisuudet ja oikealla URL.

Kuvassa 4.3 on esitetty molempien Web Menu Item -tyyppien ominaisuudet. Kuvassa vasemmalla puolella on Action-tyyppinen komponentti ja oikealla puolella on URL- tyyppinen komponentti. Molemmille elementeille yhteistä ovat AX:n perusominaisuudet, kuten asennusavaimet (Key), nimi (Name), nimilappu (Label) ja ohjeteksti (HelpText). Näistä nimilappu näkyy Enterprise Portalissa linkkitekstinä, kun valikkovaihtoehto esitetään kuten kuvassa 4.2, ja ohjeteksti esitetään linkin esitystavasta riippuen, joko tekstin alla kuten kuvassa 4.2 tai tooltip-tekstinä.

Kuvasta erottuu myös, että URL-tyyppiselle linkille voidaan antaa vähemmän arvoja. URL-arvolla kerrotaan mihinkä osoitteeseen linkki viittaa. Arvo voi olla joko absoluuttinen tai suhteellinen web-osoite. HomePage-arvoksi voidaan antaa joko Yes tai No. Yes-arvolla määritetään, että sivu voidaan liittää käyttäjän kotisivuksi eli roolikeskuksen aloitussivuksi.

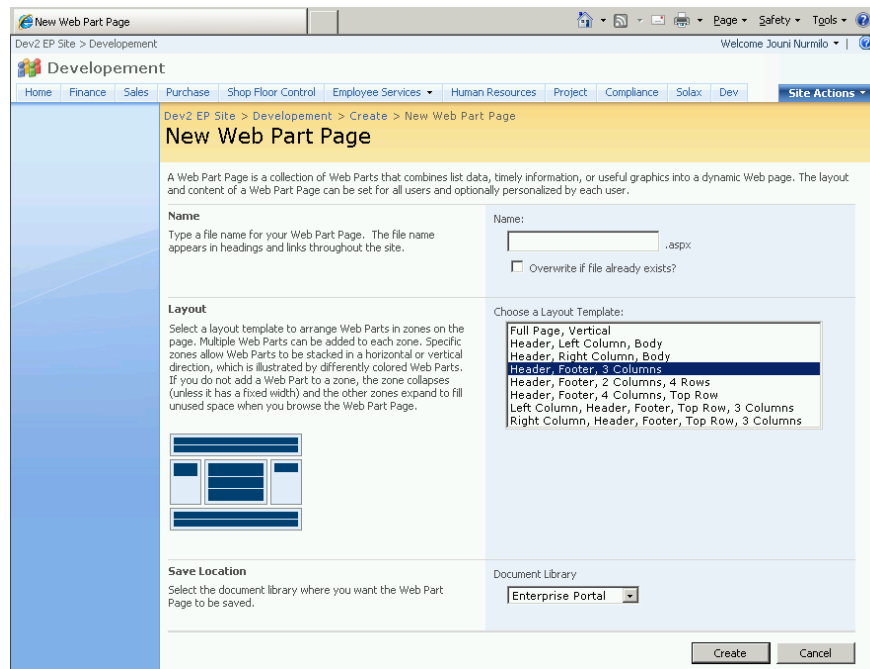
Action-tyyppiselle linkille annetaan arvoksi minkä tyyppiseen objektiin viitataan ObjectType-kenttään ja Object-kentässä kerrotaan mihinkä luokkaan tai jobiin kyseisellä linkillä viitataan. Action-tyyppiseen linkkiin voidaan määrittää Parameters-kentässä luokalle välitettävä joukko parametrejä. Lisäksi luokkakutsussa voidaan välittää luokalle EnumTypeParameter-arvossa määritetyn tyyppisen Enum tai laajennetun datatyyppin jokin arvo, joka on määritetty EnumParameter-kentässä. Lisäksi voidaan päättää, suoritetaanko luokkakutsu kutsuttaessa joko palvelimella, asiakkaalla vai kutsujassa (eli asiakkaassa tai palvelimella).

## 4.2. Enterprise Portal sivun luominen SharePoint ympäristössä

Enterprise Portalin web-sivut toimivat SharePoint-alustalla. Tämä tarkoittaa sitä, että sivujen sivupohja ja HTML-koodi generoidaan SharePointin kautta. Tämä tarkoittaa myös sitä, että Enterprise Portal sivujen koodi tallennetaan SharePoint-alustan omaan tietokantaan, josta sitä ajetaan. Tämän takia SharePoint sivut täytyy erikseen viedä AX:n tietokantaan, jotta sivut voidaan asentaa helposti uuden AX-asennuksen yhteydessä.

### 4.2.1. Sivun luominen

Enterprise Portal -sivut on helpointa luoda SharePoint-sivuston kautta uuden sivun. Tämä tapahtuu valitsemalla sivun oikeassa yläkulmassa olevasta Sita Action-pudotusvalikosta ”Create”-vaihtoehto. Tämän jälkeen sivun avautuvasta sivusta valitaan Web Pages-listan alta Web Part Page-linkki. Tämä valinta vie uuden Web Part-sivun luomiseen tarkoitettulle sivulle.



**Kuva 4.4:** Uuden web part-sivun luomisen sivu

Kuvassa 4.4 on esitettyä sivun luomiseen tarkoitettu lomake. Lomakkeessa annetaan uuden sivun nimi, jonka perään lisätään .aspx-kuvaamaan, että kyseessä on SharePoint-sivu. Alla on valintalaatikko, jonka merkitsemällä käyttäjä voi valita ylikirjoitetaanko palvelimella mahdollisesti olemassa oleva samanniminen tiedosto. On huomioitavaa, että sivu luodaan ja tallennetaan tähän lomakkeeseen annetulla nimellä siihen kansioon, mistä sivun luominen on aloitettu. Tämän kansion osoitteen voi tarkistaa myös lomake-sivun URL osoitteesta siten, että URL-osoitteesta poistetaan ”\_layouts/create.aspx” -osa lopusta ja palvelimen nimi alusta. Esimerkiksi jos sivun lisäämisen URL-osoite on ”http://axdemo/sites/DynamicsAx/Dev/\_layouts/create.aspx”, luodaan sivu ”sites/DynamicsAx/Dev/”-kansioon.

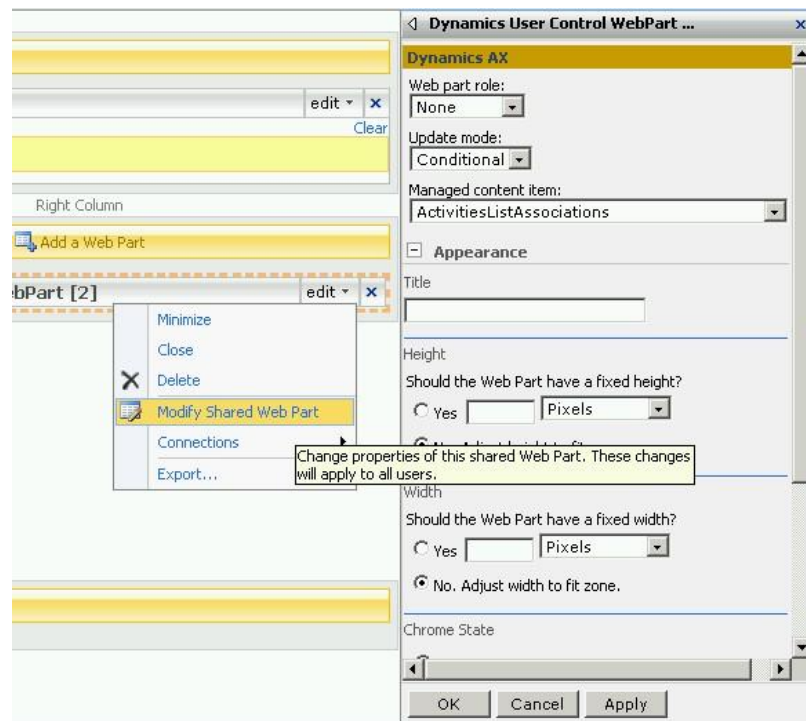
Kuvassa sivun nimen alla on Layout-valinta. Tästä listasta valitaan sivun sivupohja, mikä määrittää sivuston perusasettelun. Listan vieressä on kuva, jonka tarkoituksena on havainnollistaa miltä valittu asettelu näyttää. Sivujen asettelut löytyvät ”\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\1033\AXSITEDEF\DOCTEMP\SMARTPGS” -kansioista. Enterprise Portalin mukana tulee kahdeksan erilaista sivuasettelua valmiina.



Klikattaessa ”Create”-näppäintä luo WSS web-sivun annettujen valintojen mukaisesti ja valitulla nimellä.

#### 4.2.2. Ohjaimen lisääminen sivulle

Kun sivupohja on luotu, voidaan sivulle lisätä AX:n käyttäjäohjaimia tai mitä tahansa muita Web Part-objekteja. Käyttäjäohjaimen lisääminen tapahtuu valitsemalla klikkaamalla sivun editointitilassa näkyvää ”Add a Web Part” -palkkia halutussa kohtaa sivua. Tämän jälkeen aukeaa lisättävien Web Part -objektien ikkuna, jolta valitaan ”Dynamics User Control Web Part”-valinta lisättäessä käyttäjäohjainta sivulle. Kun käyttäjäohjaimen Web Part on liitetty sivulle, valitaan siihen haluttu käyttäjäohjain avaamalla kyseisen ohjaimen ”edit”-alasetusvalikko ja valitsemalla ”Modify Shared Web Part”-valinta, mikä aukaisee Web Part-ohjaimen asetusnäytön sivulle.



**Kuva 4.5:** Web Partin asetusikkuna

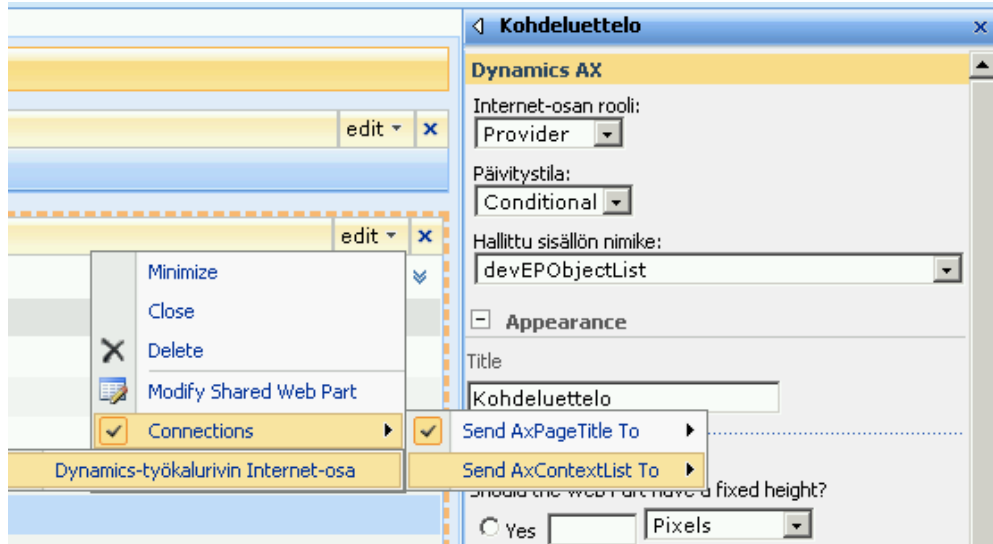
Kuvassa 4.5 on Web Partin asetusikkuna, sekä edellisessä vaiheessa käytetyt painikkeet. Web Part -objektiin valitaan haluttu käyttäjäohjain asetusikkunan ”Managed content item:” pudotusvalikosta. Tähän valikkoon on haettu AOT:n Web->WebContent->Managed -solmussa olevat ohjaimet. [2] Aetusikkunasta on mahdollista myös muun muassa määrittää käytetyn Web Part -elementin ulkoasu, koko, näytetäänkö otsikkotietoa ja onko ohjain pienennettävissä tai suljettavissa käyttäjän toimesta.

#### 4.2.3. Ohjainten välinen vuorovaikutus

Enterprise Portal web-sivu koostuu useasta eri web partista. Sivulla on usein ainakin käyttäjäohjain, otsikko ja työkalurivi. Mahdollisesti myös useampia

käyttäjähajaimia, joista yksi toimii tiedon tarjoajana, ja muut näyttävät tähän liittyvää tietoa. Eri web partit eivät automaattisesti osaa luoda yhteyksiä toisilleen, vaan niiden roolit ja yhteydet täytyy määrittellä erikseen.

Web partien tilan määrittäminen tapahtuu muokkaamalla niiden asetuksia SharePointin valikosta ja asettamalla ne joko ”Provider”- tai ”Consumer”-rooliin. Näiden roolien perusteella SharePoint tietää mihin osioihin ohjaimen kontekstia voidaan välittää. Välittäminen on mahdollista vain Provider-roolista Consumer-rooliin.

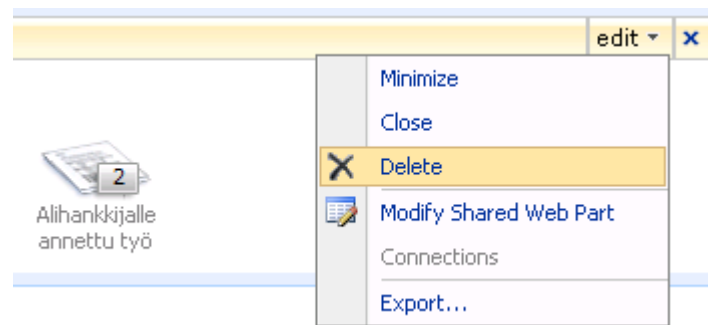


**Kuva 4.6:** Web partien välisten yhteyksien luominen

Kuvassa 4.6 on esitetty erään käyttäjähajaimen ominaisuudet välilehti oikealla sekä Connections-luettelo. Ominaisuudet välilehdeltä käyttäjähajaimelle on valittu rooliksi Provider, eli sitä käytetään tiedon välittämiseen eteenpäin. Kuvassa Connections-luettelossa on kaksi yhteyttä: AxPageTitle-sivun otsikko ja AxContextList-tiedon välitys. AxPageTitle-luettelon vieressä on merkki, joten käyttäjä ohjain on liitetty lähettämään tiedot AxPageTitle-web partille, joka näyttää sivun otsikon. Tämän lisäksi AxContextList-valinta on auki ja sieltä voidaan valita työkalurivi. Kun työkalurivi valitaan, lähettää käyttäjähajain sen aktiivisen rivin kontekstinaan työkaluriville, joka välittää sen eteenpäin käytettäessä työkalurivin linkkejä.

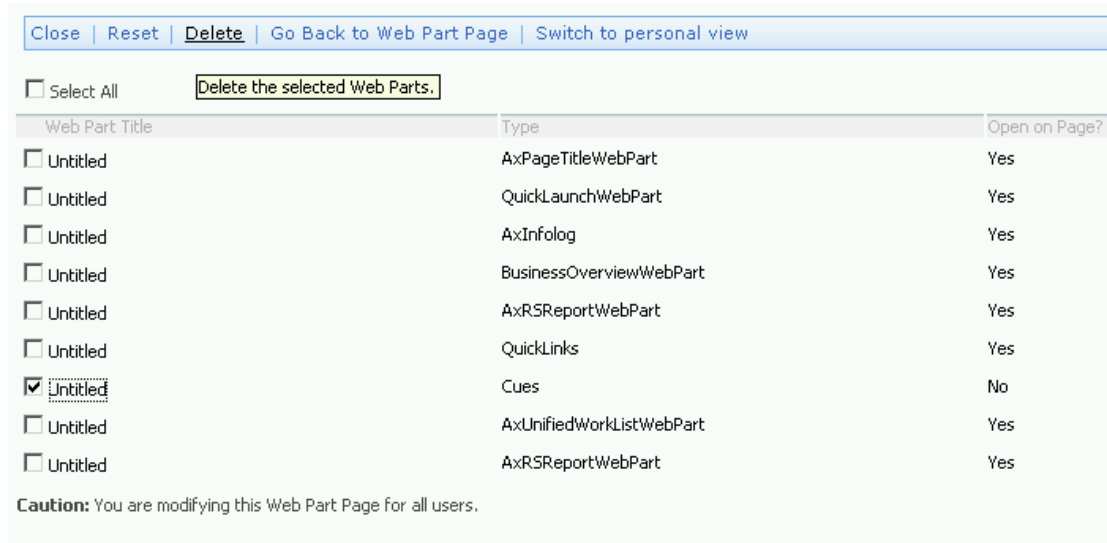
#### 4.2.4. Ohjaimen poistaminen sivulta

SharePointissa on kaksi tapaa poistaa ohjain näkyvistä. Ohjain voidaan joko sulkea Edit-valikon ”Close”-valinnasta, tai ohjaimen yläkulmassa olevasta ruksista, tai poistaa jolloin myös sen koodi poistetaan. Tämä on kuitenkin väärä tapa poistaa ohjain, sillä sen koodi jää sivun lähdekoodiin Enterprise Portal -asennusten kannalta on tärkeä poistaa käyttäjähajain, sillä mikäli käyttäjähajaimen sulkee, tulee se kuitenkin näkyviin sivulle tuotaessa sivu AX:sta.



**Kuva 4.7:** Web partin poistaminen tapahtuu Delete-valinnasta

Kuvassa 4.7 on esitetty web part -ohjaimen valikko. Ohjaimen poistaminen tapahtuu yksinkertaisesti valitsemalla web partin edit-listasta ”Delete”-vaihtoehto. Mikäli ohjain suljetaan ruksista tai ”Close”-valinnasta, ei se enää näy sivulla, mutta se jää SharePoint-sivun lähdekoodiin muistiin. Tällaisen suljetun komponentin poistaminen onnistuu vain SharePoint-sivun ohjainten listasta. Tähän listaan päästään käsiksi lisäämällä GET-parametriin, eli sivun osoitteen perään ”?contents=1”. Tämä hallintasivu on myös käyttökelpoinen apuvälinne, mikäli sivulla on ohjaimia, jotka estävät sivun avautumisen normaalisti.



**Kuva 4.8:** SharePoint-sivun web partien hallintasivu.

Kuvassa 4.8 on esitetty SharePoint-sivun web partien hallintasivu. Sivun oikeassa laidassa on nähtävillä, onko web part avoinna sivulla. Kuvassa näkyy, että yksi web part on suljettuna sivulla. Tämä web part ilmestyisi näkyviin sivulle, mikäli sivu tuodaan AX:n palvelimelta SharePointiin. Ohjain voidaan poistaa sivulta merkitsemällä se ja valitsemalla ”Delete”-linkki.

#### 4.2.5. Sivun koodin vieminen ja tuominen

Koska SharePoint ei ole AX:n osa, vaan oma erillinen ympäristönsä, täytyy siitä tuoda sivukoodit AOT-palvelimelle uudelleen asennusta varten. Sivujen vieminen

AOT-palvelimelle onnistuu Web Menu Item -olion ”Tuo”-toiminnon kautta. Sivujen viemiseen Enterprise Portal sivustolla taas on useampia tapoja. Sivun vieminen onnistuu joko yksittäin Page Definitions -olion kautta, moduuleittain Web Modules -solmun kautta, tai koko Enterprise Portal kerrallaan AxUpdatePortal-ohjelmalla [2, AxUpdatePortal].



**Kuva 4.9:** Sivun tuominen ja vieminen SharePointiin.

Kuvassa 4.9 on oikealla puolella Web Menu Items-solmun kautta tapahtuva sivun tuominen. Sivua tuodaan ”Tuo”-toiminnolla (englanniksi Import), jolloin AX hakee sivun koodin SharePoint-tietokannasta ja tallentaa sen AOT:n tietokantaan uudeksi Page Definitions -olioksi. Tämän olion vieminen takaisin SharePoint-sivustolle taas tapahtuu Page Definitions -olion ”Käytä sivua”-toiminnon (englanniksi Deploy Page) kautta. Tällöin AX vie sivun SharePoint-järjestelmään.

## 5. VISUAL STUDIO-KEHITYS

User Control -elementit perustuvat ASP.NET-koodiin, joka luodaan Microsoft Visual Studio-kehitystyökalulla. Visual Studio on .NET Business Connector-komponentilla yhteydessä AX:n AOT-sovellusobjektipuuhun. Tämän yhteyden kautta Visual Studio kykenee tunnistamaan muun muassa käytössä olevat Data Set-elementit, sekä niiden sisältämät taulut, kentät ja kenttien tyypit ja näiden tietojen avulla toteuttamaan automaattisia ASP.NET-koodin määrittämiä.

.NET Business Connector-yhteyttä käytetään myös käyttäjäohjaimien lisäämiseen tai hakemiseen AOT:sta. Kun User Controller on lisätty AOT:een, päivittää Visual Studio automaattisesti AOT:n tallennetun User Controller-koodin.

### 5.1. Käyttäjäohjaimen perusteet

User Controller -ohjain koostuu kahdesta tiedostosta. Sivun ulkoasu ja käytetyt komponentit ovat määritetyt .ascx -tiedostossa, joka taas periyttää tapahtuma- ja ohjelmakoodin .ascx.cs-tiedostosta. .ascx-tiedosto on XML-koodia, joka voi sisältää niin ASP.NET:n ohjaimien kutsuja, kuin perinteistä XHTML-koodia sivun ulkoasun määrittämiseen. .ascx.cs-tiedosto taas ohjelmoidaan C#-kielellä, ja se sisältää sivun ja sen elementtien tapahtumien (englanniksi event) koodin, sekä mahdollisesti muita apufunktioita.

Enterprise Portal kehitys vaatii Visual Studio 2008 ja sen Standard, Professional tai Team -version. Visual Studioon tulee olla asennettuna Visual C#- ja Visual Web Developer -komponentit. Lisäksi Dynamics AX:n asennuksesta on asennettava Enterprise Portal Tools-paketti, joka sisältää Visual Studio lisäosan, projektipohjat ja AX:n omat komponentit. [2, How to: Set Up Visual Studio for Enterprise Portal Development]

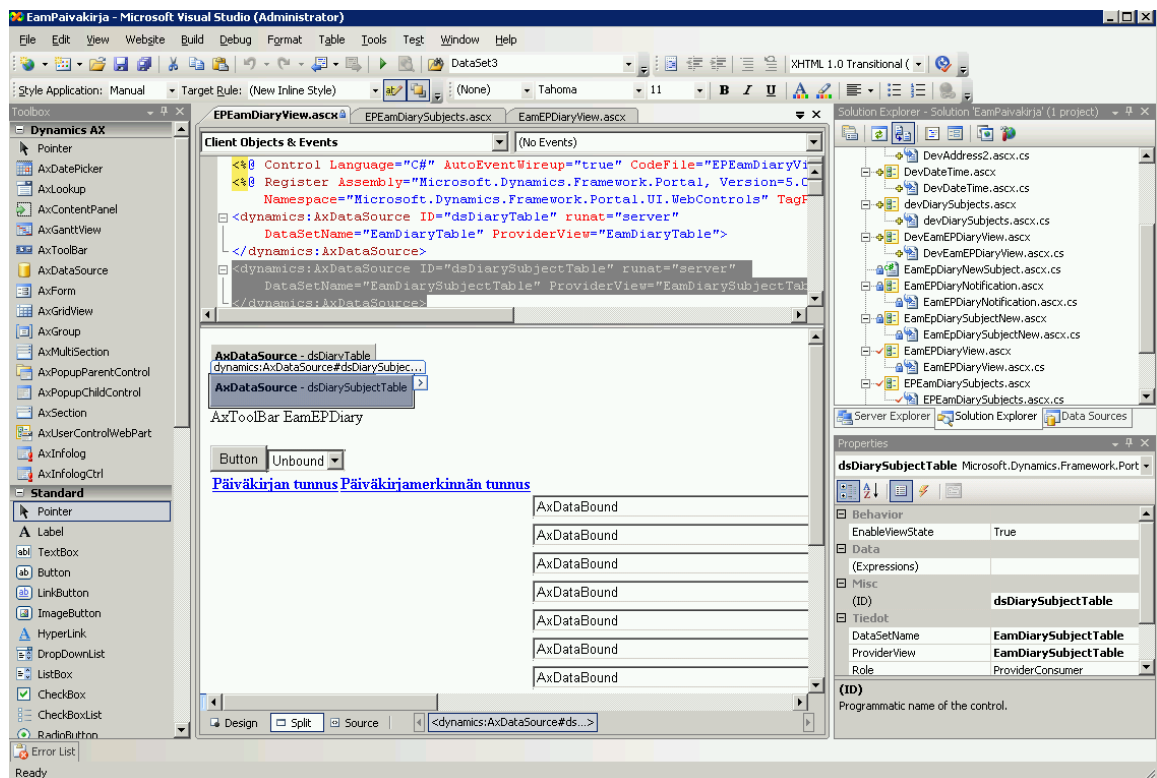
Visual Studio on yhteydessä AX-palvelimelle .NET Business Connector yhteydellä. Mikäli Visual Studio ei löydä yhteyttä AX-palvelimelle, eivät myöskään Enterprise Portal Tools -paketin komponentit ole käytettävissä Visual Studiossa. Yhteyttä AX-palvelimeen hyödynnetään lisättäessä AX-komponentteja, jolloin Visual Studio kykenee hakemaan AX-palvelimelta listan kenttään soveltuvista arvoista, kuten Data Setin nimen tai taulun käyttämät kentät.

AX-yhteyttä voidaan käyttää myös olemassa olevien käyttäjäohjainten hakemiseen AOT:sta, sekä luodun käyttäjäohjaimen lisäämiseen AOT:een. Mikäli käyttäjäohjain on lisätty AOT-palvelimeen, Visual Studio tallentaa käyttäjäohjaimen AOT:een aina, kun ohjain tallennetaan Visual Studiossa. Tältä toiminnolta voidaan välttyä lisäämällä käyttäjäohjainten koodi, AOT:ssa AX:n omaan versionhallintaan, jolloin Visual Studio

kykenee tekemään tallennuksia vain kyseisen käyttäjäohjaimen ollessa uloskuitattuna AX:n versionhallinnassa.

## 5.2. Kehitysympäristö

Enterprise Portal -käyttäjäohjainten kehitys tapahtuu Visual Studio -ympäristössä, johon on asennettu Enterprise Portal Tools -paketti. Enterprise Portal kehityksessä voidaan käyttää kaikkia Visual Studion ominaisuuksia hyväksi. Enterprise Portal kehityksessä Visual Studion tulee olla yhteydessä AX-palvelimelle, jotta Enterprise Portal Tools -paketin ominaisuudet olisivat käytettävissä.



*Kuva 5.1: Visual Studio Enterprise Portal kehityksessä*

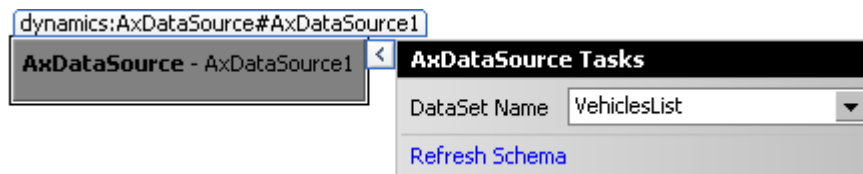
Kuvassa 5.1 on kuvakaappaus Visual Studio kehitysympäristöstä Enterprise Portal -tilassa. Kuvan vasemmassa laidassa on työkaluvalikko, josta voidaan valita komponentteja lisättäväksi käyttäjäohjaimen. Valikossa ensimmäisenä on Dynamics AX -komponentti ryhmä, ja tämän lisäksi valikossa on kokoelmia HTML- ja ASP.NET-komponentteja.

Kuvan keskiosassa on käyttäjäohjaimen käyttöliittymän kehitysnäkymä. Kuvassa se on jaetussa tilassa, jossa näkyvät samanaikaisesti sekä käyttöliittymän koodi (Source) että ulkoasunäkymä (Design). Keskiosan näkymäksi voidaan valita jaetun näkymän lisäksi myös pelkkä ulkoasunäkymä tai koodinäkymä. Komponenttien lisääminen käyttäjäohjaimen onnistuu helposti raahaamalla haluttu komponentti työkaluvalikosta haluttuun kohtaan käyttäjäohjaimella. Raahaamisen voi suorittaa sekä ulkoasunäkymään että koodinäkymään. Valitun komponentin oletuskoodi lisätään siihen kohtaan koodia,

mihin komponentti on pudotettu. Myös jo sijoitettujen komponenttien paikkaa voidaan vaihtaa raahaamalla se ulkoasunäkymässä toiseen kohtaan, tai koodissa leikka/liimaa-menetelmällä.

Kuvan oikeassa laidassa ylempänä on projektinäkömä. Siinä näkyvät kaikki projektiin liitetyt käyttäjäohjaimien koodit, sekä muut tiedostot. Käyttäjäohjain näytetään listassa hierarkkisesti avautuvana solmuna, jossa juuren toimii käyttäjäohjaimen ulkoasun koodi, ja sen alle aukeaa käyttäjäohjaimen ASP.NET -ohjelmakoodi.

Kuvan oikeassa alakulmassa on komponentin ominaisuudet esittävät lomake. Ominaisuudet on mahdollista esittää joko ryhmiteltynä aihealueittain tai aakkosjärjestyksessä. Ominaisuuksien muuttaminen onnistuu helposti klikkaamalla valittua ominaisuutta lomakkeella ja syöttämällä sille haluttu arvo. Kun ominaisuus on syötetty tällä lomakkeella, lisää Visual Studio kyseisen ominaisuuden komponentin koodiin parametrina.



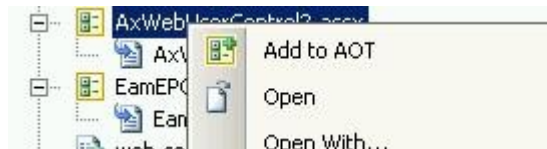
**Kuva 5.2:** Pikavalikko Dynamics AX komponenteille

Kuvan 5.1 alemmassa AxDataSourcessa on nähtävillä oikeassa yläkulmassa pieni oikealle osoittava nuoli. Kun tätä nuolta painaa, aukeaa pikavalikko (englanniksi Context menu), joka on esitetty kuvassa 5.2 AxDataSourcen tapauksessa. Pikavalikkoon aukeaa komponentin tyyppiin liittyvää tietoa ja pudotusvalikko, johon haetaan mahdolliset arvot .NET Business Connectorilla AX:n tietokannasta. Kuvan 5.2 tapauksessa haetaan AOT:sta kaikkien Data Settien nimet, jotka voidaan liittää kyseiseen AxDataSource-komponenttiin. Pikavalikkossa on myös linkki ”Refresh Schema”. Klikkaamalla tätä linkkiä Visual Studio päivittää Data Set:n metatiedot AOT:sta. Samalla se myös päivittää kaikkien siihen liitettyjen komponenttien sarakekentät ja tietoavaimet sekä poistaa komponentteihin aiemmin liitetyt kentät.

### 5.2.1. Kehityksen työnkulku

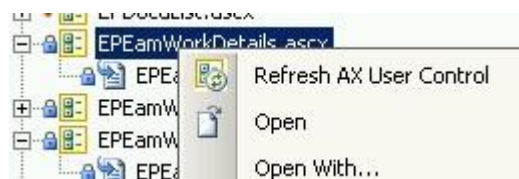
Luotaessa Visual Studiolla Enterprise Portalin käyttäjäohjainta, tulee ohjainta varten luoda ensiksi asianmukainen Web Site -projekti. Oikea projekti löytyy ”File->New Web Site...”-valikosta. Enterprise Portal -projektiä varten tulee käyttää .NET Framework 2.0:a ja Visual C#-kieltä. Käytettävä projekti on nimeltään ”Dynamics AX Web Project”, joka on valittavissa ”My Templates” -osiosta.

Seuraavaksi luodaan käyttäjäohjaimen pohja ”File->New File...”-valikosta ja valitaan ”My Templates” -osiosta Dynamics AX User Control -malli, sekä annetaan sille nimi. Tämä luo kaksi nimen mukaista tiedostoa: ”Nimi.ascx” XML-muotoisen ulkoasutiedoston ja ”Nimi.ascx.cs” C#-koodiluokan.



**Kuva 5.3:** Käyttäjöhjaimen vienti AOT:n

Luotu käyttäjäohjain täytyy lisätä AOT:n, jotta sitä voidaan käyttää Enterprise Portal sivulla. Lisäys tapahtuu hiiren oikealla klikkauksella projektinäkömässä käyttäjäohjaimen kohdalla ja valitsemalla ”Add to AOT”-toiminto. Tämä valikko on esitetty kuvassa 5.3 Tällä toiminnolla Visual Studio vie käyttäjäohjaimen koodit AOT:n Web->Web Files->Web Controls -solmun alle, sekä luo Web->Web Content->Managed -solmun alle SharePointissa käytettävät käyttäjäohjaimen metatiedot.



**Kuva 5.4:** Käyttäjöhjaimen päivittäminen AOT:sta

Kun käyttäjäohjain on kerran viety AOT:n, muistaa Visual Studio tämän. Tällöin kun käyttäjä tallentaa muokatun käyttäjäohjaimen Visual Studiossa, vie Visual Studio muutokset myös automaattisesti AOT:n. Kuvasta 5.4 nähdään, kun käyttäjä klikkaa AOT:n vietyä käyttäjäohjainta hiiren oikealla napilla projektinäkömässä, on ”Add to AOT”-toiminto muuttunut ”Refresh AX User Control”-toiminnoksi. Tämä toiminto palauttaa käyttäjäohjaimen koodit AOT:sta, eikä sitä siis käytetä koodin päivittämiseen AOT:n, joka toteutetaan automaattisesti tallennuksen yhteydessä.



**Kuva 5.5:** käyttäjöhjaimen hakeminen AOT:sta

Kun halutaan muokata olemassa olevaa käyttäjäohjainta, voidaan se myös tuoda suoraan AOT:sta. Kuvassa 5.5 on esitetty, kuinka tämä onnistuu oikealla klikkauksella projektivalikkossa ja valitsemalla ”Add User Control from AOT”, joka avaa listan kaikista AOT:n käyttäjäohjaimista, ja listasta voidaan valita halutut käyttäjäohjaimet muokattavaksi. Dynamics AX-tyylisesti myös kaikki käyttäjäohjaimien koodit ovat



luettavissa ja muokattavissa, kuten Dynamics AX:n lomakkeidenkin koodit ovat kaikkien kehittäjien käytettävissä.

Kun käyttäjäohjaimen pohja on luotu, lisätään sille AxDataSource-tietolähde, joka on yhteydessä valittuun AOT:n Data Set-olioon. AxDataSource-komponentti mahdollistaa käyttäjäohjaimen muille komponenteille AX:n ennalta määrätyn datan käyttämisen. Käyttäjäohjaimelle voi lisätä useampia AxDataSource-tyyppisiä komponentteja, mutta vain yksi niistä voi olla Provider-roolissa oleva tietolähde.

Seuraavaksi käyttäjäohjaimen lisätään halutut elementit. Yleensä joko AxGridView-komponentti tai AxForm-komponentti ja siihen liittyvät alue- ja ryhmätyyppiset komponentit. Kun halutut komponentit on lisätty, ohjelmoidaan niihin liittyvä tapahtumalogiikka .cs-tiedostoon.

## 5.2.2. Komponenttien ominaisuudet

Käyttäjäohjaimen komponenttien toimintaa määritellään komponenttiin liitetyillä ominaisuuksilla. Komponenttien ominaisuudet määritetään käyttäjäohjaimen XML-koodissa komponentin attribuuteilla. Jokaisella käyttäjäohjaimella on useita ominaisuuksia, joita voidaan määrittää, ja mikäli jotain ominaisuutta ei ole määritelty, käytetään ominaisuuden oletusarvoa käyttäjäohjainta ajettaessa. Ominaisuuksien määrittäminen voidaan tehdä joko määrittelemällä se suoraan XML-koodiin tai Visual Studion ominaisuudet-välilehdellä, joka lisää määrittämisen yhteydessä ominaisuuden XML-koodiin.

Jokaiselle komponentille annetaan ID-ominaisuus, jolla se voidaan tunnistaa ja siihen voidaan viitata C#-koodissa. Samassa käyttäjäohjaimessa ei saa olla samaa ID-ominaisuutta useammalla komponentilla, ja koodin luettavuuden ja selvyyden vuoksi tulisi komponentit nimetä loogisesti ja selväkielisesti.

Useat komponentit käsittelevät tai näyttävät dataa ja tällöin ne ovat yhteydessä AX:n tietokantaan AxDataSource-komponentin kautta. Käytetty AxDataSource määritellään komponenttiin DataSourceID-ominaisuudella. Tämän lisäksi määritetään käytetty näkymä DataMember-ominaisuuteen, jonka tulee olla saatavilla AxDataSource-komponentin kautta. Näiden lisäksi on tarpeen määrittää DataKeyNames-arvot, mikäli käyttäjäohjain toimii lähettäjän tai vastaanottajan roolissa. Tähän arvoon määritetään kenttä tai kentät, joilla voidaan yksilöidä välitettävä tai vastaanotettava tietue.

Näiden lisäksi kaikkien näkyvien komponenttien ulkoasua voidaan vaihtaa useammallakin ominaisuudella. Näitä ominaisuuksia voidaan vaihtaa muun muassa Appearance ja Layout-ominaisuusryhmien alta. Appearance-ryhmässä on ainakin CssClass-ominaisuus, jolla voidaan määrittää komponentissa käytetty CSS-luokka, sekä muita ulkoasuun liittyviä parametreja. Layout-ryhmässä taas on ohjaimen asetteluun ja kokoon liittyviä ominaisuuksia, kuten komponentin korkeus ja leveysäädöt.

Komponenttien käytettävyyttä voidaan parantaa Accesibility-ryhmästä, mistä voidaan määrittää ohjaimelle XHTML-standardin mukaiset TabIndex- ja AccessKey-ominaisuudet. TabIndex määrittää missä järjestyksessä komponentteihin kohdistetaan

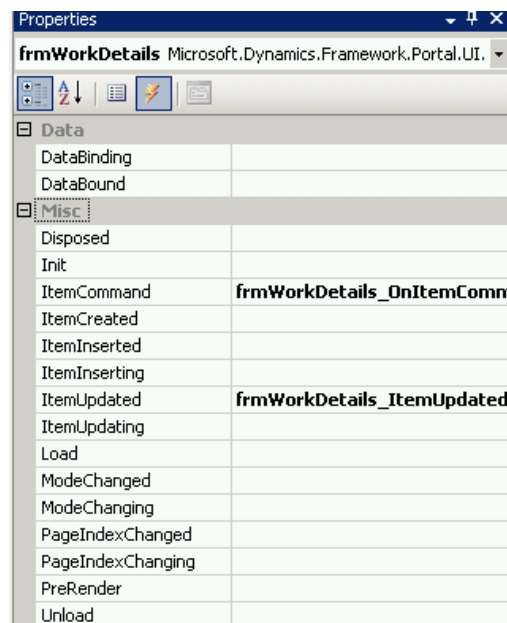
käytettäessä sivulla sarkainnäppäintä. AccessKey-ominaisuudella taas voidaan määrittää näppäin, jolla kyseinen komponentti aktivoituu. [9, TabIndex, AccessKey]

### 5.3. Yleisiä toimintoja käyttäjäohjaimissa

Käyttäjäohjainta ohjelmoitaessa on tiettyjä toimintoja, joita tehdään usein. Tällaisia toimintoja ovat muun muassa tapahtumien yhteydessä ajettavien ohjelmien kirjoittaminen tai käyttäjän ohjaaminen toiselle sivulle.

#### 5.3.1. Eventit eli tapahtumat

Käyttäjäohjaimen ehkä tärkein ohjelmitava ominaisuus on tapahtumien (englanniksi events) yhteydessä suoritettavat koodit. Tapahtumissa on koodia, joka aktivoituu kyseisen tapahtuman tapahtuessa käyttäjäohjaimessa. Jokaisella komponentilla sekä sivulla on omat tapahtumafunktionsa. Kaikilla Enterprise Portal komponenteilla on käytettävissä ASP.NET:n standardit tapahtumat: DataBinding tietoa ladattaessa, Disposed päätettäessä ohjaimen olemassaoloa, Init alustettaessa ohjainta, Load ladatessa ohjainta, PreRender juuri ennen kuin ohjain esitetään ja Unload kun ohjainta puretaan. Näiden lisäksi komponenteilla on myös omia Enterprise Portal -kohtaisia tapahtumia kuten esimerkiksi Activate, ItemInserted tai SectionCollapsed.



**Kuva 5.6:** AxForm-komponentin mahdolliset tapahtumat.

Kuvassa 5.6 on esitetty AxForm-komponentin kaikki tapahtumat. Lista näistä tapahtumista sijaitsee komponentin ominaisuudet ikkunan ”Events”-valinnan (ikoninaan salama) takaa. Listassa on havaittavissa komponentti kohtaisia tapahtumia kuten ItemInserted ja ItemInserting. Huomattavaa näissä tapahtumissa on, että ItemInserting ajetaan ennen kuin tietue on lisätty ja ItemInserted ajetaan lisäämisen jälkeen. Uusi tapahtuma voidaan lisätä joko tuplaklikkaamalla tapahtumakenttää, jolloin Visual

Studio luo uuden tapahtuman rungon, tai valitsemalla pudotusvalikosta funktio ajettavaksi toiminnon tapahtuessa. Ohjelmassa 5.1 on esitetty OnItemCommand-funktio, joka aktivoituu, kun ”OK”- tai ”Peruuta”-nappia on painettu.

**Ohjelma 5.1: OnItemCommand-tapahtuman koodi.**

```
protected void frmWorkDetails_OnItemCommand(object sender,
                                           DetailsViewCommandEventArgs e)
{
    if (e.CommandName == "OK")
    {
        this.DataSourceView.EndEdit();
    }
    if (e.CommandName == "Cancel")
    {
        this.DataSourceView.CancelEdit();
        this.RedirectToPreviousPage();
    }
}
```

Ohjelma 5.1 esittää esimerkin tapahtumakoodista, joka tapahtuu OnItemCommand-tapahtuman yhteydessä. Funktio saa parametreinaan sitä kutsuneen olion sekä argumenttikokoelman. Argumenttikokoelma sisältää käytetyn ominaisuuden nimen CommandName-parametrissä. Tätä nimeä verrataan komponentin painikkeisiin ”OK” ja ”Cancel”. Mikäli käyttäjä on painanut ”OK”-painiketta, päätetään rivin muokkaus. Jos käyttäjä taas painaa ”Cancel”-painiketta, peruutetaan rivin muokkaus ja palataan takaisin edelliselle sivulle.

### 5.3.2. Käyttäjäohjaimen tilan määrittäminen

Käyttäjäohjaimen tila voidaan määrittellä sivulle välitettävän parametrin avulla. Parametri voidaan välittää ohjaimelle SharePoint sivun ja http-protokollan [8] avulla, esimerkiksi post-metodilla [8, 9.5 POST] sivun URI:ssa, jota Enterprise Portalin yhteydessä yleisesti käytetään.

AX:ssa on määritettyä laajennettu tietotyyppi (Extended data type) nimeltään EPFormAction, mitä on suositeltava käytettäväksi sivun moodia määriteltäessä. EPFormAction-laajennettu tietotyyppi sisältää arvot sivun luomis-, poisto-, muokkaus-, info- ja listaustiloille. Käyttämällä EPFormAction tietotyyppiä, säilyy koodi luettavana ja tiedetään helposti mitä tilaa tarkoitetaan.

**Ohjelma 5.2: Ohjaimen tilan määrittäminen EPFormAction tietotyyppiä käyttäen**

```

private void SetupMode()
{
    EPFormAction FormMode =
    (EPFormAction)
    Convert.ToInt16(this.Page.Request.QueryString.Get("mode"));

    switch (this.FormMode)
    {
        // Muokkaus
        case EPFormAction.EditMode:
            this.AxForm1.DefaultMode = DetailsViewMode.Edit;
            this.AxForm1.AutoGenerateEditButton = true;
            break;
        // Katselu
        case EPFormAction.ViewMode:
            this.AxForm1.DefaultMode = DetailsViewMode.ReadOnly;
            this.AxForm1.AutoGenerateEditButton = false;
            break;
        // Luonti
        case EPFormAction.CreateMode:
            this.AxForm1.DefaultMode = DetailsViewMode.Insert;
            this.AxForm1.AutoGenerateInsertButton = true;
            this.AxForm2.Visible = false;
            break;
        // Oletus
        default:
            this.AxForm1.DefaultMode = DetailsViewMode.Edit;
            this.AxForm1.AutoGenerateEditButton = true;
            break;
    }
}

```

Ohjelmassa 5.2 on esitetty kuinka käyttäjäohjaimen tilan määrittäminen voidaan toteuttaa ohjaimelle välitetyn parametrin perusteella. Tilan määrittäminen tehdään omassa SetupMode()-funktiossa, jota voitaisiin kutsua esimerkiksi sivun init-metodissa. SetupMode-funktiossa luetaan ensimmäisenä HTTP:n GET-metodilla välitetyn ”mode”-parametrin arvo. Tämä arvo luetaan EPFormAction-tyyppiseen muuttujaan, joka on enumlista, ja sisältää sivun mahdolliset tilat. Tämän muuttujan perusteella valitaan switch-lauseella, mihin tilaan käyttäjäohjaimen AxForm1-lomake asetetaan ja näytetäänkö lisätietolomake AxForm2:ta. Mikäli parametria ei saada, tai se on virheellinen, käytetään oletusvalintaa, jossa käyttäjäohjain asetetaan muokkaustilaan.

### 5.3.3. Data Set-kutsut

Käyttäjöohjaimia ohjelmoitaessa tulee vastaan usein tilanne, jossa halutaan käyttää Data Setin metodeja, luoda uusia rajauksia tietoon tai muuten vaikuttaa tietueisiin Data Setin kautta. Kutsuja voidaan tehdä suoraan Data Setin metodeihin, tai niille voidaan välittää ajoparametreissa muuttujia, joilla voidaan vaikuttaa koodin toimintaan.

#### *Ohjelma 5.3: MyTrips metodin kutsuminen Data Set:stä*

```
DataSetView dsv =
    this.FMTrips.GetDataSet().DataSetViews["FMTrips"];
dsv.DataSet.DataSetRun.AxaptaObjectAdapter.Call("MyTrips",
                                                this.emplid);
```

Ohjelmassa 5.3 on tehty kutsu Data Setin MyTrips-metodiin. Ensimmäisessä lauseessa luodaan dsv DataSetView-muuttuja, johon haetaan FMTrips-tietolähteestä sen ”FMTrips” näkymä. Toisessa lauseessa suoritetaan dsv-muuttujalle DataSet-metodikutsu MyTrips-metodille, jossa parametrinä on this.emplid-muuttuja, mihin on tallennettu käyttäjän id.

#### *Ohjelma 5.4: Enum-parametring välitys Data Set:lle*

```
void AxDataSource1_CreatingDataSetRun(object sender,
                                     CreatingDataSetRunEventArgs e)
{
    e.DataSetRunArgs.parmEnumType =
        EnumMetadata.EnumNum(this.AxSession, "EPViewType");
    e.DataSetRunArgs.parmEnum(1);
}
```

Ohjelmassa 5.4 on esitetty ennen Data Set- ajoa luotaessa ajettava ohjelma. Tässä ohjelmassa asetetaan Data Setin ajoparametrin enumeraatiotyyppi parmEnumType, EnumMetaData-luokalla ”EPViewType” enumeraatioksi. Tämän jälkeen tämän enumeraation arvoksi asetetaan arvo indeksistä yksi.

### 5.3.4. Käännöstekstit

Microsoft Dynamics AX 2009 on useita eri kieliä tukeva ohjelma ja se sisältää oman sisäisen sanakirjansa. Tämä sanakirja on myös käytettävissä käyttäjäohjaimissa, mutta sen käyttö ei ole aivan yhtä vaivatonta kuin Windows-asiakasohjelmassa. AX:n käännöstekstit (englanniksi label) saadaan käyttöön sekä käyttöliittymänäkymässä, että ohjelmakoodissa.

Käyttöliittymänäkymässä käännöstekstit haetaan <%\$ axlabel:@TUNNUS %> -komennolla, josta @TUNNUS-osa korvataan AX-labelin tunnuksella. Ohjelmakoodissa taas labeli haetaan seuraavasti:

```
Microsoft.Dynamics.Framework.Portal.UI.Labels.GetLabel("@TUNNUS");
```

Käytettäessä AX:n käännostekstejä, hakee .NET Business Connector käyttäjän asetusten perusteella tunnuksen mukaisen käännoksenn. Mikäli käyttäjän oletuskielillä kyseistä koodia ei ole määritelty, palautetaan käyttäjän toissijaisen kielen mukainen käännos. Mikäli tunnusta ei löydy, näkyy käyttäjäohjaimessa tekstinä vain kyseinen käännostunnus, eli esimerkiksi "@TUNNUS".

### 5.3.5. Uudelleen ohjaaminen

Käyttäjäohjaimen suoritettua toiminnon tulee usein vastaan tilanne, että käyttäjä halutaan ohjata toiselle sivulle. Tämä uudelleenohjaus voidaan toteuttaa ASP.NET:n HttpResponseMessage-luokan Redirect-metodilla. Tällöin Redirect-funktiolle välitetään URL-osoite String-muuttujassa. Koska Enterprise Portalin linkit ovat tallessa AOT:n Web Menu Items-olioissa, täytyy URL-osoite hakea AOT:sta. Tämä tapahtuu luomalla uusi AxUrlMenuItem-muuttuja new-metodilla, jossa rakentajaluokka saa Web Menu Item:n nimen parametrinaan. Tästä muuttujasta voidaan lukea URL-osoite sen URL-kentästä ja URL-kenttä OriginalString muuttujasta. Näin saatua arvoa voidaan käyttää HttpResponseMessage-luokan Redirect-funktion parametrina. Ohjelmassa 5.5 on esitetty esimerkki uudelleen ohjauksen toteuttamisesta.

#### *Ohjelma 5.5: Esimerkki uudelleen ohjauksesta.*

```
private void RedirectToPage(bool Previous)
{
    if (Previous) {
        Response.Redirect(AxWebSession.GetPreviousURL(this.Page));
    } else {
        AxUrlMenuItem listpage = new
            AxUrlMenuItem("EamEPObjectDetails");
        Response.Redirect(listpage.Url.OriginalString "&mode=" +
            EPFormAction.EditMode);
    }
}
```

Ohjelmassa 5.5 on esitetty esimerkifunktio uudelleen ohjauksesta toiselle sivulle. Funktio saa parametrinaan boolean-tyyppisen muuttujan, joka kertoo ohjataanko käyttäjä selaimen edelliselle sivulle, vai ennalta määritellylle Enterprise Portal -sivulle. Funktiota voitaisiin käyttää esimerkiksi "OK"/"Cancel"-nappien yhteydessä, jolloin "OK"-nappi ohjaisi ennalta määritelly sivulle ja "Cancel"-nappi takaisin.

Uudelleenohjaus edelliselle sivulle tapahtuu AxWebSession-luokan GetPreviousURL-funktiolla, joka saa parametrinaan kyseisen sivun. Tämä sivu haetaan selaimen välimuistista. Ohjelmassa 5.5 on esitetty myös ohjaus ennalta määriteltyyn osoitteeseen. Osoite saadaan AOT:n Web Menu Item-oliolta, mutta Redirect-funktion kutsussa siihen on lisätty HTTP:n GET-parametriin mode-muuttuja, jolla asetetaan uudelleen ohjattu sivu muokkaustilaan.

## 6. AX-SIDONTA

Visual Studioon on tehty Enterprise Portalia varten useita erilaisia omia komponentteja, joilla voidaan esittää tai käsitellä AX:ssa olevaa tietoa. Osa komponenteista on tarkoitettu vain tiedon esittämiseen ruudulla, osa tiedon hakemiseen AX:n kannasta tai poimimiseen ja osa tiedon ryhmittelyyn ohjaimessa.

### 6.1. AxDataSource

AxDataSource-komponentin tarkoituksena on tuoda AOT:ssa luodut Data Set-oliot käytettäväksi ASP.NET-koodissa. AxDataSource:n kautta voidaan tuoda käyttäjäohjaimelle, Data Set-olioon liitetty data sekä metodit käytettäväksi käyttäjäohjaimen koodissa. AxDataSource:n kautta AX:n tieto on käytettävissä myös ei AX-spesifeissä ASP.NET-ohjaimissa ohjainten DataSourceID-ominaisuuden kautta. [1, s.253]

AxDataSource on yksinkertainen säiliö, joka koostuu yhdestä tai useammasta nimetystä näkymästä AxDataSourceView. AxDataSourceView-näkymä sisältää itsessään funktiot datan lukemiseen ja kirjoittamiseen. AxDataSourceView viittaa AxDataSource-komponenttiin linkitetyn Data Setin näkymään. Tällainen näkymä voi olla joko yksittäinen taulu tai liitostaulu. AxDataSourceView hakee automaattisesti käytettävissä olevat kentät, kenttätypit sekä kenttäryhmät. Näytettävät kentät rajataan AX-palvelimella perustuen kentille määritettyihin avaimiin, sekä käyttäjän oikeuksiin käsitellä kyseisiä kenttiä. [1, s.253]

Useampien AxDataSource-komponenttien on mahdollista toimia yhdessä siten, että yksi AxDataSource toimii lähettäjänä (provider) ja muut ovat vastaanottajana (consumer). Tällöin lähettäjän roolissa oleva komponentti välittää sen aktiivisena olevan rivin tiedot Record Contextissa, ja vastaanottaja roolissa olevat komponentit ottavat tiedon vastaan Record Contextista ja suorittavat tiedonrajaamiseen sen perusteella. Record Context lähetetään aina uudelleen, kun aktiivinen rivi vaihtuu tarjoajan näkymässä. [1, s.253]

Tiedonrajaamista toisen AxDataSourceen avulla voidaan hyödyntää esimerkiksi tilanteessa, jossa käyttäjälle tarjotaan mahdollisuus valita luettelonäkymän avulla lähemmin tarkasteltava päätason taulun tietue, esimerkiksi lasku. Tähän luettelonäkymään on liitetty AxDataSource-komponentti, joka toimii provider -tilassa. Toisessa luettelonäkymässä taas voi olla alitaulu, josta on vierasavain viite päätauluun, ja jonka AxDataSource-komponentti toimii customer-tilassa. Esimerkin tapauksessa kyseessä voisi olla laskun rivit, jotka näytetään valittuun tietueeseen liittyen.

Tiedonrajaamista on mahdollista tehdä myös käyttäjäohjaimen ASP.NET -koodilla. Tällöin AxDataSource-objektin läpi välitetään rajausparametrit filterObject -oliolla AOT:n DataSet-olioon. Ohjelmassa 6.1 on tehty esimerkki tiedon rajaamisesta ASP.NET koodissa.

**Ohjelma 6.1:** *Esimerkki ohjelmakoodi tiedon rajaamisen käyttäjäohjaimen koodissa.*

```
protected void btnEmployeeTrips_Click(object sender, EventArgs e)
{
    // Haetaan nykyinen DataSetView
    DataSetView dsv = this.axDataSource1.GetDataSet().DataSetViews[0];
    // Tyhjennetään avoinna olevat rajaukset.
    Dsv.UserFilter.ClearOpenFilter();
    // Luodaan rajauksessa käytetty filterObject
    Microsoft.Dynamics.Framework.Data.Ax.filterObject flt =
        new filterObject();

    flt.name = "EmployeesTripsFilter";

    // Luodaan hakuehto conditionType-luokkaan,
    // joka lisätään filterObjectiin
    Microsoft.Dynamics.Framework.Data.Ax.conditionType typ =
        new conditionType();

    typ.__operator =
        Microsoft.Dynamics.Framework.Data.Ax.operatorType.eq;
    typ.status =
        Microsoft.Dynamics.Framework.Data.Ax.conditionStatus.open;
    typ.value = this.EmployeeId.text;
    typ.attribute = "EmpId";
    // Lisätään ehto filterObjectiin
    flt.conditionCollection.Add(typ);
    // Lisätään filterObject DataSetView:n
    dsv.UserFilter.Add(flt);
}
```

Ohjelman 6.1 rajauksesta on nähtävissä, kuinka tiedon rajaaminen onnistuu ASP.NET koodissa. Tiedon rajaukseen haetaan ensimmäisenä nykyinen DataSetView ja tyhjennetään siinä mahdollisesti valmiina olevat rajausehdot ClearOpenFilter()-komennolla. Tämän jälkeen luodaan rajausolio filterObject ja nimetään se. Seuraavaksi luodaan rajausehdon sisältävä conditionType-olio, joka myöhemmin liitetään rajausolioon. ConditionType-oliolle määritellään millainen rajaus on kyseessä operatorType-enumeratiolla. Status-muuttuja määrittää, onko käyttäjällä mahdollisuus käsitellä rajausta, value-muuttujaan kerrotaan käytettävän rajauksen arvo, joka ohjelman 6.1 tapauksessa haetaan käyttäjäohjaimen kentästä. Attribute-muuttuja kertoo, mihin tietueen kenttään rajauksen ehtoa verrataan. Lopuksi conditionType lisätään filterObject:n ehtokokoelmaan ja rajausehto lisätään tietokantanäkymään, jolloin seuraavan tietokantanäkymän haun yhteydessä tiedot on rajattu ehdon mukaisesti.



## 6.2. AxGridView

AxGridView-komponentti on tarkoitettu useamman tietorivin esittämiseen kerralla näytöllä. Tiedot esitetään riveittäin taulukonmuotoisessa ruudukossa. Siksi AxGridView soveltuu erityisesti listanäkymien ja listausten näyttämiseen käyttäjälle. AxGridView-komponentti laajentaa ASP.NET:n GridView-komponentin [3, MSDN .NET GridView] toimimaan Dynamics AX:n kanssa. Se tarjoaa mahdollisuudet muun muassa näytetyn tiedon rivikohtaiseen käsittelyyn, valitsemiseen, ryhmittelyyn ja rajaamiseen yhdessä AxDataSet-komponentin kanssa. [1]



Päiväkirjan tunnus	Päiväkirja merkinnän tunnus	Kuvaus	Kohteen tunniste	Tila	Työntekijä	Tapahtuman aika	Ilmoitukset	Luotu
INT00000001	00000300_183	Luotu EP v v	10UUSI		3400	15.10.2010 13:05:40	<input type="checkbox"/>	15.10.2010 13:05:53
INT00000001	00000296_183	nn	API00000117	tila2	6666	15.10.2010 12:51:17	<input type="checkbox"/>	15.10.2010 12:51:37
INT00000001	00000293_183		API00000117		3102	8.10.2010 15:00:35	<input checked="" type="checkbox"/>	8.10.2010 15:00:45
INT00000001	00000279_183	gg	0		3000	17.9.2010 14:47:28	<input checked="" type="checkbox"/>	17.9.2010 14:47:28
INT00000001	00000049_183		10		3102	16.9.2010 14:55:34	<input checked="" type="checkbox"/>	16.9.2010 14:55:08
INT00000001	00000276_183	Descriptionf fff ggg s aaa	6	tila1	3400	3.9.2010 12:46:46	<input type="checkbox"/>	3.9.2010 12:46:56
INT00000001	00000270_183	Uusi	6	tila2	3400	31.8.2010 13:26:47	<input type="checkbox"/>	31.8.2010 13:26:59
INT00000001	00000268_183	10 10	10		3400	31.8.2010 10:23:44	<input type="checkbox"/>	31.8.2010 10:23:59
INT00000001	00000254_183	gg uu	10UUSI		3400	30.8.2010 11:58:43	<input type="checkbox"/>	30.8.2010 11:58:53
INT00000001	00000252_183	s	PIKopio		3400	30.8.2010 11:43:36	<input type="checkbox"/>	30.8.2010 11:43:50

**Kuva 6.1:** AxGridView-komponentti

Kuvassa 6.1 on normaali AxGridView:n näkymä. Komponentin yläosassa on nähtävillä ”Edit filter”-linkki, jota klikkaamalla taulukkoon aukeaa suodatusehtojen ikkuna, jossa näkyvät mahdollisesti käytössä olevat suodatuksia, ja jossa käyttäjä voi halutessaan, lisätä tai poistaa suodatuksia. Suodatuksen alapuolella on näkyvillä sarakkeiden nimet. Sarakenimeä klikkaamalla voidaan taulukko järjestää kyseisen sarakkeen perusteella nousevaan tai laskevaan järjestykseen. Näiden alla on ennalta määrätty määrä tietueita, tai vähemmän, mikäli enempää tietueita ei voida näyttää. Yksi taulukon riveistä on niin sanottu aktiivinen rivi, joka kuvan tapauksessa on merkitty vaaleansinisellä värillä.. Sivun alaosassa on sivunumerot, joilla voidaan hakea näkyviin seuraavat sivut näytettävistä tiedoista.

AxGridView-taulukossa näkyvät sarakkeet määritellään ASP.NET:n .ascx -tiedostossa. Sarakkeet tulevat <AxGridView>-komponentin sisään osaksi <columns> -komponenttiä ja ne ovat boundfield-tyyppisiä elementtejä. Sarakkeen mahdolliset tyypit ovat: tavallinen AxBoundField-sarake [2], AxHyperlinkBoundField [2], joka on linkkityyppinen kenttä, AxBoundFieldGroup, mikä on AOT:ssa ennalta määritelty ryhmä kenttiä [1], AxCheckBoxBoundField, joka on valintaruutu-tyyppinen kenttä, AxDropDownBoundField, mikä on pudotusvalikko, AxRadioButtonBoundField, mikä on valintanappi tyyppinen kenttä ja TemplateField, jolle voidaan koodissa määrittää tarkemmin ulkoasu. [2]

Kaikille BoundField-tyyppisille kentille tulee kertoa mitä kenttää ja mistä DataSetin näkymästä ne edustavat DataField, DataSet, DataSetView -ominaisuuksilla. Lisäksi

kentälle voidaan antaa muita ominaisuuksia AX:n SDK:ssa [2] määriteltyjen ominaisuuksien mukaan. AxHyperLinkBoundField-komponentille tulee määrittää DataField:n lisäksi myös MenuItem ominaisuuteen jokin AOT:n WebMenuItem:n olio, minkä perusteella kyseiseen sarakkeeseen luodaan linkki WebMenuItem:n osoittamaan osoitteeseen. AxHyperLinkBoundField-komponentille voidaan kertoa myös avattavan linkin kohde tai määrittää toisen kentän perusteella mitä AOT:n WebMenuItem:a käytetään linkkinä [2].

Kun AxGridView-komponentille määritetään kenttiä Visual Studiassa, osaa Visual Studio automaattisesti määrittää Business Connectorilla saamansa kentän metadatan perusteella, minkä tyyppisestä BoundField-komponentista on kyse, ja ohjelmoija joutuu harvoin itse puuttumaan kooditasolla näihin määrityksiin. Poislukien AxHyperLinkBoundField-komponentti, jota Visual Studio ei automaattisesti tunnista. Vaan ohjelmoija joutuu aina itse kertomaan koodissa, mikäli hän haluaa käyttää kyseistä saraketyyppiä.

AxGridView:llä näkyvä vaaleansininen rivi on niin sanottu aktiivinen rivi eli rivi johon DataSetin kursori osoittaa. Tämä rivi on se, jonka tiedot välitetään Provider-roolissa olevasta DataSet:stä Consumer-roolin ohjaimiin.

AxGridView-taulukossa näytetään enintään ennalta määritetty määrä rivejä, tai vähemmän, kerrallaan. Mikäli rivejä on enemmän kuin ennalta määritetty määrä, suoritetaan rivien sivutus (englanniksi paging) useammalle näkymälle. Näitä sivuja kuvaamaan luodaan taulukon alareunaan numerointi alkaen yhdestä. Klikkaamalla jotain näistä numeroista siirtyy taulukko näyttämään kyseisen sivun tietueet. Sivun koko määritetään PageSize-attribuutilla ja aktiivinen sivu pageIndex-attribuutilla. Sivuttaminen voidaan ottaa myös kokonaan pois päältä asettamalla AllowPaging-attribuutin arvoksi false [2].

AxGridView-taulukolle on lisäksi mahdollisuus määrittellä kolme erilaista lisäominaisuutta. Nämä lisäominaisuudet ovat laajennettava rivi, ryhmitelty taulukko sekä hiiren oikealla napilla aukeava linkkivalikko. Lisäominaisuuksista laajennettavaa riviä sekä taulukon ryhmittely eivät toimi kunnolla yhdessä, ja siksi on suositeltavaa, ettei niitä käytettäisi samaan aikaan.

CustomerList				
ID	Name	Segment	Subsegment	Main contact
4000	Light and Design	Retail	Large	LJJ
<b>Telephone</b> 213 555 0100				

**Kuva 6.2:** Laajennettava rivi [2, AxGridView]

Kuvassa 6.2 on esitetty rivi, jolle on määritelty laajennusosa. Kun rivin laajennus otetaan käyttöön, ilmestyy taulukossa jokaisen rivin loppuun nuoli, josta rivin voi laajentaa. Laajennettaessa riviä, aukeaa rivin alle näkyviin ennalta määritellyt kentät. Rivin laajennus saadaan käyttöön määrittelemällä ShowExpansion-parametrin arvoksi

true sekä antamalla ExpansionColumnIndexesHidden-parametriin laajennososasta piilotettavien kenttien indeksinumero ja GridColumnIndexesHidden-parametriin taulukko-osasta piilotettavien kenttien indeksinumero. Kenttien indeksinumero määräytyvät siinä järjestyksessä, kuin ne on Grid-komponentille määritelty numerosta 1 alkaen. [2 ,GridView]

Työnumero	Nimi	Kohteen tunniste	Työn tila	Projekti
ProjId :				
ProjId : 10001				
ProjId : 10001-1				
ProjId : 10001-11				
ProjId : 10001-2				
APP003943		11808	Työn alla	10001-2
APP006544			Tilattu	10001-2
APP007233			Valmis	10001-2

**Kuva 6.3:** Rivien ryhmittelynäkömä

Rivit voidaan ryhmitellä valmiiksi jonkin kentän mukaan. Tällöin AxGridView- taulukon rivit esiintyvät kuvan 6.3 mukaisesti jaoteltuna. Ryhmittelyyn valitun kentän arvo näkyy ryhmän otsikkona, ja klikattaessa ryhmän vieressä olevaa plussa-kuvaketta aukeaa ryhmän rivit normaalin taulukon tapaan. Ryhmittely voidaan ottaa käyttöön määrittelemällä AllowGrouping-parametrin arvoksi ”true” sekä GroupField-parametriin ryhmittely kentän DataField-parametrin arvo. Lisäksi ryhmittelyyn käytetyn kentän SortExpression-arvo täytyy olla määriteltynä. Ryhmittelyn alkuarvo avattuna vai suljettuna voidaan määritellä DefaultCollapsed-arvolla. Lisäksi ryhmittelyn toimintaa voidaan ohjata AllowGroupCollapse-arvolla, jolla voidaan estää ryhmien sulkeminen tai avaaminen, ja DisplayGroupFieldName-arvolla, jolla voidaan määrittää näytetäänkö ryhmittelyyn käytetyn kentän otsikko. [2 ,GridView]

Vehicle Number	Description	Employee
A101	Van 1	CJC
A102	Van 2	AFE
B100	Box Truck	UFF
C100	Flatbed Truck	ASR

**Kuva 6.4:** Konteksti lista, joka aukeaa hiiren oikealla napilla [2, AxGridView]

Riville on myös mahdollista lisätä kuvan 6.4 mukainen valikko, joka aukeaa klikattaessa hiiren toisella napilla aktiivista riviä. Mikäli avatun valikon linkkiä klikataan, aukaisee se linkin lähettäen aktiivisen rivin linkille kontekstina. Valikko voidaan ottaa käyttöön määrittelemällä ShowContextMenu-parametrin arvoksi ”true” ja antamalla ContextMenuName-parametrin arvoksi halutun AOT:n Web Menu-olion nimi. [2, AxGridView]

### 6.3. AxForm - lomake

AxForm on tapa esittää ja käsitellä yksittäisen tietueen tai tietojoukon tietoja. AxForm-lomake koostuu kolmesta osasta AxMultiSection, AxSection ja AxGroup komponenteista, joiden tehtävänä on ryhmitellä lomakkeella näytettävä tieto. Lomakkeella haettu tietue on sille määritetyn DataSource:n aktiivinen rivi.

AxForm-lomakkeella käytetty tietue määritellään kuten muissakin tietolähdettä käyttävissä elementeissä DataSourceID, DataMember ja DataKeyNames ominaisuuksilla. Huomattavaa tietolähteen määrittelyssä kuitenkin on, että DataMember-ominaisuutena voidaan käyttää vain *\_Current* näkymää, joka sisältää ainoastaan yhden tietueen.

Lomakkeelle tulee määritellä sen lähtötila *DefaultMode*-attribuuttiin. Oletuksena lomake on lukutilassa. Muita mahdollisia tiloja ovat muokkaus- ja lisäystila. Luku- ja muokkaustilassa lomakkeelle haetaan tietolähteen nykyisen rivin tiedot, kun taas lisäystilassa tietue on uusi ja siinä on vain tietolähteessä *initValue*:ssa määritetyt arvot. Muokkaus- ja lisäystilassa oltaessa lomakkeelle voidaan luoda valmiiksi muokkauksen tai lisäyksen tekevät napit asettamalla *AutoGenerateEditButton* tai *AutoGenerateInsertButton*-ominaisuuksiin *True*-arvo. Lomake voidaan määrittää myös päivittymään automaattisesti *UpdateOnPostBack*-ominaisuudella aina kun tehdään PostBack-toiminto, eli esimerkiksi kun näytöllä on muokattu jonkin kentän arvoa. Automaattinen päivitys on hyödyllinen toiminto, mikäli jonkin kentän arvon perusteella haetaan muihin kenttiin arvoja joko AOT:n tai käyttäjäohjaimen koodissa.

AxForm-lomake rakentuu AxMultiSection, AxSection, AxGroup ja sen sisältämistä AxBoundfield-ryhmän elementeistä. Yksinkertainen lomake voi koostua vain AxGroup-ryhmän elementeistä. Monimutkaisemmat lomakkeet voivat sisältää myös AxMultiSection- ja sen sisältämiä AxSection-elementtejä.

The screenshot displays an AxForm interface with the following structure:

- AxForm** (Root)
  - AxMultiSection**
    - Asetukset** (AxSection)
    - Myyntitilaus** (AxSection)
    - Maksu** (AxSection) - Expanded
      - Maksu** (AxGroup)
        - Maksuehto:
        - Maksutapa:
        - Maksunääritys:
        - Maksusuunnitelma:
    - Yhteystiedot** (AxSection) - Expanded
      - Dimensio** (AxSection) - Expanded
        - Dimensiot** (AxGroup)
          - Osasto:
          - Kustannuspaikka:
          - Tarkoitus:

At the bottom right, there are two buttons: **OK** and **Peruuta**.

**Kuva 6.5:** AxForm lomakenäkymä

Kuvassa 6.5 on esitetty AxForm-elementti sekä sen sisältämät elementit. Kuvasta näkee kuinka AxMultiSection-elementillä kootaan useampia AxSection-elementtejä yhteen ja kuinka AxSection-elementtien sisällä on AxGroup-elementit. Lisäksi kuvasta näkee miltä AxSection-elementti näyttää suljettuna ja laajennettuna.

**Listaus 6.2:** Lomakkeen rakenne XML-koodina ulkoasunmäärittelyssä

```

<AxForm>
  <AxMultiSection>
    <AxSection>
      <AxGroup>
        <Fields>
          <AxBoundField />
          ...
        </Fields>
      </AxGroup>
    </AxSection>
    <AxSection>
      <AxGroup>
        <Fields>
          <AxBoundField />
          ...
        </Fields>
      </AxGroup>
    </AxSection>
    ...
  </AxMultiSection>
</AxForm>

```

Listauksessa 6.2 on esitetty lomakkeen rakenne XML-koodina. Kyseessä on pelkistetty listaus, josta puuttuvat kaikki elementtien vaatimat attribuutit. Rakenteessa esitetyt kolme pistettä koodissa kuvaavat, että yllä olevaa rakennetta, eli BoundField-tyyppisiä kenttiä tai AxSection-kokonaisuuksia voi olla useampia elementin sisällä.

AxMultiSection-monialue mahdollistaa AxSection-elementtien näyttämisen ja niistä koostuvan kokoelman luomisen. Oletusarvoisesti AxSection-elementit ovat laajennettuina ohjainta ladatessa, mutta AxMultiSection mahdollistaa myös toiminnon, jossa vain yksi siihen kuuluva AxSection voi kerrallaan olla laajennettuna. Tällöin *ActiveMode*-attribuutti on asetettava päälle. Kun *ActiveMode*-ominaisuus on käytössä, voidaan aktiivisena oleva alue valita koodissa käyttämällä elementin *ActiveSectionIndex*-attribuuttia. AxMultiSection-elementtiin voi sisältää vain AxSection-elementtejä.

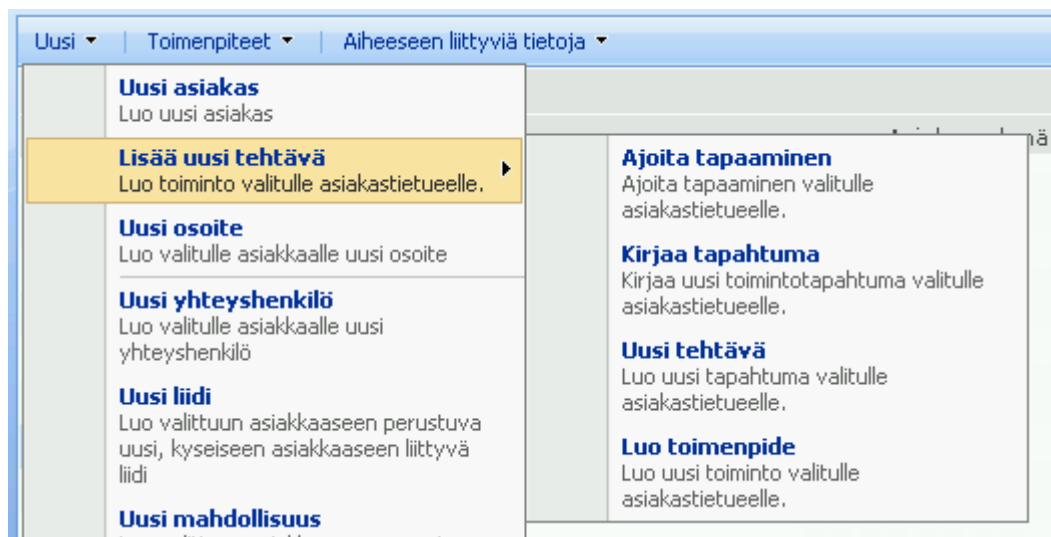
AxSection-alue on geneerinen kontrollien säiliö. Se voi sisältää minkä tahansa tyyppisiä elementtejä lapsielementteinään. AxSection tyyppisen kentän tarkoitus on kerätä sisälleen looginen kokonaisuus, joka on otsikoitu ja joka voidaan tarvittaessa laajentaa tai sulkea pois näkyvistä. AxSection-elementin yhteydessä voidaan käyttää myös *SecurityKey*-avainta määrittämään vaaditut käyttöoikeudet alueen näkymiseen.

AxGroup-elementtiä käytetään esittämään AxDataSourcen kenttiä lomakkeella. Kentät listataan <Fields>-elementin sisälle ja ne esitetään käyttämällä sopivaa AxBoundField-ryhmän elementtejä. Visual Studio osaa määrittää automaattisesti valitun elementin tyyppin kentän metadatan perusteella.

## 6.4. AxToolbar - työkalurivi

AxToolbar on komponentti, jolla voidaan esittää linkkejä ja toimintoja työkalurivinä käyttäjäohjaimessa. Se laajentaa Dynamics Toolbar komponentin, joka on mahdollista liittää itsenäisenä elementtinään Enterprise Portal –sivulle. AxToolbar tarjoaa kaikki Dynamics Toolbarin toiminnot käyttäjäohjaimessa sekä mahdollisuuden muokata valikkoriviä käyttäjäohjaimen koodissa. [1]

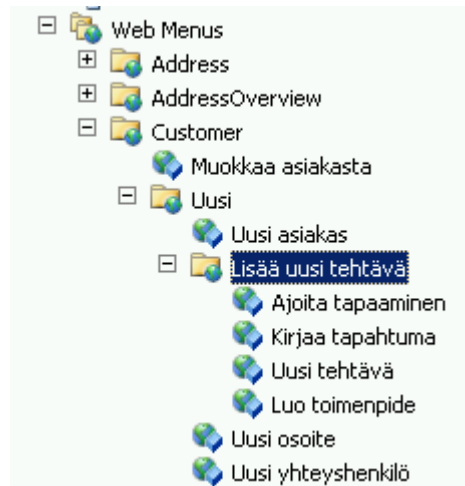
AxToolbar:lla on mahdollista esittää kokoelma linkkejä, joko painikkeina tai alavetovalikkoina, joita on vielä mahdollista laajentaa sivullepäin. AxToolbar:n ulkoasu ja toiminta on havainnollistettu 6.6.



*Kuva 6.6: AxToolbar-työkalurivi*

Kuvassa 6.6 on esitetty AxToolbarin toiminta. Alasvetovalikot ja sivuvalikot aukeavat, kun hiiren kursori viedään niiden päälle ja paljastaa valikon seuraavan tason. Alas- ja sivuvetovalikoiden teksteihin voidaan haluttaessa lisätä linkkejä, jotka aktivoituvat niitä klikattaessa.

Helpoin tapa luoda työkalurivin valikkoja on tehdä AOT:ssa WebMenu-tyyppinen linkkikokoelma, joka voi sisältää hierarkkisen kokonaisuuden WebMenuItem:n Action- tai URL-tyyppisiä linkkejä. Näistä Action-tyyppisiä linkkejä käytetään Ax:n sisäisten toimintojen suorittamiseen ja URL-tyyppiset linkit voivat viitata toiseen Enterprise Portal -sivuun tai mihin tahansa muuhun URL-osoitteeseen.



**Kuva 6.7:** Web Menus solmu AOT:ssa, jossa kuvan 6.6 navigointinäkymä

Käytettävä AOT:n WebMenu voidaan määrittää AxToolbar-komponentille sen *Dynamics AX Web Menu*-ominaisuudella. AOT:n WebMenu:sta muodostuu työkalurivin valikko siten, että ensimmäiset SubMenu-tyyppiset solmut muodostavat työkaluriviin alavetovalikot. SubMenun alle liitetyt WebMenuItem:t muodostavat valikon vaihtoehdot, ja sivulle aukeava valinta luodaan uudella sisäkkäisellä SubMenu:lla, johon voidaan liittää vielä WebMenuItem:t valinnoiksi.

AxToolbar-komponentti on mahdollista sitoa AxDataSource-tietolähteeseen tai se voidaan jättää sitomatta. Mikäli työkalurivi on sidottu tietolähteeseen, saa se automaattisesti tietolähteen aktiivisesta rivistä contextin elementeilleen, joka välitetään kohteelle käytettäessä valikkorivin kohteita. Mikäli työkalurivi on sitomattomana, joudutaan sen välittämä contexti määrittelemään koodissa. [1]

Koska SharePointin mukana toimitettava Dynamics Toolbar on tarkoitettu ensisijaisesti käytettäväksi työkaluriviksi, on sitä syytä käyttää mikäli työkalurivin ei tarvitse olla interaktiivinen käyttäjäohjaimen kanssa. Jos työkalurivin kuitenkin pitää olla interaktiivinen, on syytä käydä lävitse myös muutamia tärkeitä koodinpätkiä, joilla voidaan manipuloida työkalurivillä näytettäviä linkkejä, tai välitettävää tietoa.

Käyttäjäohjaimen ja työkalurivin välillä on pääsääntöisesti kaksi tapaa interaktioon. Toinen tapa on käyttäjäohjaimen toimintoihin ja aktiiviseen riviin liittyvä linkkien näyttäminen tai käyttöönotto/ käytöstä poisto. Toinen tapa on taas työkalurivin linkkien klikkaamiseen liittyvät tapahtumat käyttäjäohjaimessa.

Työkalurivin klikkauksia voidaan seurata ActionMenuItemClicking- ja ActionMenuItemClicked-tapahtumilla. ActionMenuItemClicking-tapahtuma tapahtuu, kun käyttäjä on klikannut työkalurivin linkkiä, mutta ennen kuin linkkiin liittyvä toiminto on tehty. Tämän toiminnon yhteydessä voidaan määrittää esimerkiksi mitä välitetään linkin kohteelle.

**Ohjelma 6.3:** Linkin ajamisen estäminen, mikäli poistaminen ei ole sallittu [1]

```

void webpart_ActionMenuItemClicking(object sender,
                                   ActionMenuItemClickingEventArgs e)
{
    if ((e.MenuItem.MenuItemAOTName.ToLower() ==
        "epitemdelete") && !this.allowDelete)
    {
        e.RunMenuItem = false;
    }
}

```

Ohjelmassa 6.3 on esitetty linkin ajamisen estäminen, mikäli poistaminen ei ole sallittua. ActionMenuItemClicking-toiminto ajaa funktion, joka saa parametreinaan lähetettävän objectin, sekä ActionMenuItemClickingEventArgs-olion, joka sisältää tapahtumaan liittyvät ajoparametrit (Args). Funktiossa tarkistetaan mitä linkkiä on painettu, lukemalla ajoparametreista AOT:n WebMenuItem:n nimi, sekä onko ohjaimessa poistaminen luvallista. Mikäli poistoon ei ole lupaa, mutta silti yritetään poistaa riviä, asetetaan ajoparametrien RunMenuItem-parametri false-arvoon, joka estää MenuItem:n toiminnon ajamisen.

Mikäli halutaan vaikuttaa työkalurivillä näkyviin linkkeihin tai vaihtaa niitä, tulee kutsua SetMenuItemProperties-toimintoa, jolla voidaan muuttaa työkalurivin toimintaa, kuten linkkien tai alasettovalikkojen näkyvyyttä tai saatavuutta, tai määrittää linkin yhteydessä välitettävää tietuetta.

**Ohjelma 6.4:** Linkin klikkaamisen estäminen, mikäli poistaminen ei ole sallittu. [1]

```

void webpart_ActionMenuItemClicking(object sender,
                                   SetMenuItemPropertiesEventArgs e)
{
    if ((e.MenuItem.MenuItemAOTName.ToLower()
        == "epitemdelete") && !this.allowDelete)
    {
        ((AxUrlMenuItem)e.MenuItem).Disabled = true;
    }
}

```

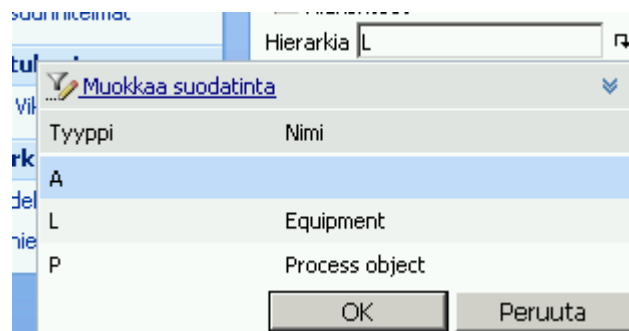
Ohjelmassa 6.4 on esitetty poistolinkin estäminen samoilla ehdoilla kuin listauksessa 6.3 estettiin toiminnon ajo. Listauksessa 6.4 estetään työkalurivin linkin käyttäminen, mikäli poistaminen ei ole mahdollista. Tällöin linkki näkyy harmaana työkalurivillä, mutta se ei ole käytettävissä. Esto tehdään castaamalla ajoparametrien MenuItem AxUrlMenuItem tyyppiseksi oliksi ja määrittämällä sen Disabled-muuttujan arvoksi *true*.



## 6.5. AxLookup - poimintaikkuna

AxLookup-komponentin käyttötarkoitus on tarjota käyttäjälle apukeino oikean arvon syöttämiseen tiedonsyöttökenttään. AxLookup-komponentilla käyttäjälle tarjotaan tiedon poimintalista, jolta käyttäjä voi valita laillisen arvon kenttäänsä. Tämä poimintalista luodaan AX:n tietokannan tietojen mukaan, ja se voi viitata joko jonkun taulun sisältöön tai laajennetun tietotyypin arvovaihtoehtoihin.[1, s260]

Sen lisäksi, että poimintaikkuna on mahdollista valita AxGridView tai AxForm -komponentteihin määriteltyihin kenttiin, voidaan poimintaikkuna luoda mihin tahansa kenttään AxLookup-komponentin avulla. [1, s260]



*Kuva 6.8: AxLookup-komponentilla luotu poimintaikkuna*

AxLookup komponentti luo kuvan 6.8 mukaisen poimintaikkunan, joka aukeaa sivun päälle. Poimintaikkuna on luotu HTML:n DIV-elementtejä ja CSS-määrittämiä apunaan käyttäen siten, että se näyttäisi aukeavan omana ikkunanaan sivun päälle. Poimintaikkunassa tietueet näytetään AxGridView-näkymässä. Näkymä sisältää yleensä korkeintaan kymmenen tietuetta, ja sivutusta käytetään useamman kentän näyttämiseen. Lisäksi poimintanäkymä voi olla järjesteltävissä kenttien mukaan, ja sille on mahdollista käyttää suodatusta.

AxLookup-komponentille määritetään mihin kenttään tietoa ollaan hakemassa, ja mistä kentästä tieto haetaan. Tietoa voidaan hakea minkä tahansa dataset:n näkymästä. Lisäksi valintalistasta voidaan määrittellä mitä muuta tietoa kuin haettava tieto näytetään valintalistalla. Lisätiedon saaminen näkyviin on erityisen hyödyllistä haettaessa arvoa, joka ei itsessään kerro mitä sillä tarkoitetaan, kuten tunnus. Tällöin voidaan valintalistalla näyttää esimerkiksi tunnuksen lisäksi tietueen nimitiedot tai muita merkityksellisiä arvoja.

Kentille, joille AX:ssa on määriteltynä oletus poimintanäkymä, voidaan määrittää Visual Studioissa ohjaimen koodilla, näytetäänkö mahdollisuus käyttää poimintanäkymää. Oletuksena poimintanäkymää käytetään, mikäli se on käytettävissä. Poimintanäkymä voidaan poistaa kentiltä määrittämällä kentät LookupButtonDisplaySettings-arvoksi "Never" tai se voidaan pakottaa määräämällä arvoksi "Always" [1, s260]

Oletusarvoisesti poimintanäyttö voidaan luoda kenttiin, jotka ovat laajennettua tietotyyppiä, mikä sisältää relaation johonkin AX:n tauluun, jolle on määritetty

AutoLookup-kenttäryhmä. Tällöin näytetään AutoLookup-kenttäryhmään määritellyt kentät valitussa järjestyksessä. [2, Lookups]

Mikäli automaattinen poimintalista ei ole riittävä, tai esimerkiksi listalle valittavien arvojen joukkoa halutaan karsia, on mahdollista luoda muokattuja valintalistoja. Muokatun valintalistan voi luoda, joko luomalla X++:lla AOT:n DataSet:n oman dataSetLookup-metodin Visual Studioissa käyttäjäohjaimen määrittämiseen tai käyttäjäohjaimen koodiin OnLookup-tapahtumasta ajettavaan funktioon. [2, Lookups]

### *Listaus 6.5: AxLookup-painikkeen XML-määrittely*

```
<dynamics:AxLookup ID="VehiclesLookup" runat="server"
  TargetControlId="Vehicle" ExtendedDataType="VehicleNum"
  LookupType="EDT" SelectField="VehicleNum">
  <Fields>
    <dynamics:AxBoundField DataField="VehicleNum"
      SortExpression="VehicleNum" />
    <dynamics:AxBoundField DataField="Description"
      SortExpression="Description" />
    <dynamics:AxBoundField DataField="EmplId"
      SortExpression="EmplId" />
  </Fields>
</dynamics:AxLookup>
```

Listauksessa 6.5 on kuvattu poimintaikkunan luominen käyttäjäohjaimen määrittämisen avulla. Poimintaikkuna käyttää AxLookup-komponenttia, jolle määritetään ID:n lisäksi parametrit: LookupType, joka määrittää minkä tyyppisestä tietojoukosta tietoa haetaan, ExtendedDataType, määrittää mistä laajennetusta datatyyppistä tietoa haetaan, SelectField kuvaa minkä kentän arvo valitaan, ja TargetControlId määrittää, mihin komponenttiin tieto tuodaan. LookupType-parametrillä kuvataan minkä tyyppisestä datasetistä tietoa haetaan. AxLookup-komponentille määritetään Fields-elementti, johon lisätään kaikki poimintaikkunassa käytettävät kentät. Listauksessa 6.5 poimintaikkunaan on määritetty näkymään kolme kenttää.

### *Ohjelma 6.6: AxLookup-ikkunan luominen ohjelmakoodissa*

```
protected void Vehicles_Lookup(object sender, AxLookupEventArgs e)
{
    AxLookup lookup = (AxLookup)sender;
    // Luodaan lookup Data Set Vehicles taululle.
    Proxy.SysDataSetBuilder sysDataSetBuilder;
    sysDataSetBuilder =
        Proxy.SysDataSetBuilder.constructLookupDataSet(
            AxSession.AxaptaAdapter,
            TableMetadata.TableNum(AxSession, "Vehicles"));
    // Asetetaan luotu Data Set Lookup Data Setiksi.
    lookup.LookupDataSet = new DataSet(AxSession,
        sysDataSetBuilder.toDataSet());
    // Määritetään Lookup kentät
```

```
lookup.Fields.Add(AxBoundFieldFactory.Create(AxSession,
    lookup.LookupDataSetViewMetadata.ViewFields["VehicleNum"]));

lookup.Fields.Add(AxBoundFieldFactory.Create(AxSession,
    lookup.LookupDataSetViewMetadata.ViewFields["Description"]));

lookup.Fields.Add(AxBoundFieldFactory.Create(AxSession,
    lookup.LookupDataSetViewMetadata.ViewFields["EmpId"]));
// Määritetään poimittava arvo
lookup.SelectField = "VehicleNum";
}
```

Ohjelmassa 6.6 on kuvattu poimintaikkunan luomista käyttäjäohjaimen koodissa. Tämän kaltaisen poimintaikkunaa voidaan käyttää, kun määrittelyssä on lisätty poimintaikkunan kohdekenttään OnLookup-tapahtumaan poimintaikkunan koodin funktio. Muokattu valintalista luodaan käyttäjäohjaimen koodiin ohjelman 6.6 mukaisella funktiolla. Funktiossa luodaan uusi AxLookup-tyyppinen olio, joka alustetaan sisääntulevan AxLookupEventArgs:n avulla vanhan poimintaikkunan päälle. Poimintaikkunalle kerrotaan sen käyttämä dataset, lookup-ikkunassa näkyvät kentät, sekä minkä kentän arvo haetaan poimintaikkunalta. Olion DataSetille on mahdollista tehdä samat toiminnot, kuten tiedon rajaamisen, kuin tavallisellekin DataSetille. Siksi muokattu valintalista sopii tapauksiin, joissa näytettävää tietoa on hyvä rajata annettujen ehtojen perusteella. [8]

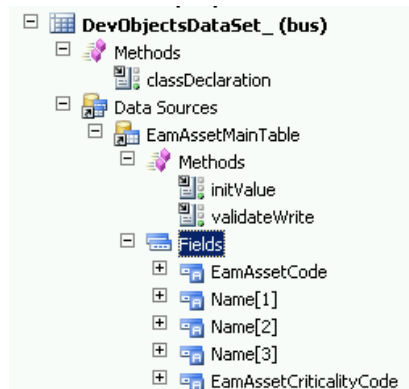
## 7. ESIMERKKISOVELLUS

Esimerkkisovelluksena toteutetaan yksinkertainen kohteidenhallintajärjestelmä. Esimerkkisovellus koostuu kahdesta käyttäjäohjaimesta sekä kahdesta Enterprise Portal -sivusta. Esimerkkisovelluksen tarkoituksena on esitellä kohteita kokonaisuutena, joka koostuu kaikki kohteet esittävästä listasta, kohteiden sekä kohdetietueiden katselun, lisäyksen ja muokkaamisen käsittävistä näytöistä.

Ennen Enterprise Portal -osuuden luomista on Dynamics AX:ssa täytynyt luoda ainakin ohjelmassa käytetyt taulut, niihin liittyvät kentät ja kenttien osalta laajennetut tietotyypit. Koska taulujen luominen ei kuulu työn aihepiiriin, ei luomisprosessia esitellä tarkemmin, vaan taulujen sisältö käydään lävitse vain ohjelman kannalta merkityksellisimpien kohtien osalta.

### 7.1. Kohteiden hallinta

Esimerkkisovelluksen tekeminen aloitetaan luomalla Visual Studioon Enterprise Portal -projekti. Minkä jälkeen aloitetaan kohteiden käsittelyyn tarkoitettun käyttäjäohjaimen ohjelmointi. Tämä käyttäjäohjain tulee kohteiden hallintaan ja sillä voidaan näyttää, muokata sekä lisätä kohdetietueita järjestelmään.



*Kuva 7.1: Kohdetiedoissa käytetty DataSet*

Ennen kuin käyttäjäohjaimen ohjelmointi voidaan aloittaa Visual Studiossa, tulee ohjaimelle luoda sen vaatima Data Set-olio AOT:n. Luotu Data Set-olio on esitetty kuvassa 7.1. Data Set-olio on nimetty DevObjectsDataSet\_-nimiseksi, ja siihen liitetään EamAssetMainTable niminen taulu sen Data Source -osaan, ja tämä osa nimetään EamAssetMainTable nimiseksi taulun mukaan. Tässä vaiheessa ei vielä kuormiteta Data Set:n metodeja tai luoda uusia metodeja, vaan ne lisätään Data Set-oliolle myöhemmässä vaiheessa.

## 7.2. Käyttäjöohjaimen katselutila ja ulkoasu

Kun Data Set -runko on luotu, voidaan Visual Studiossa aloittaa käyttäjöohjaimen luominen. Aloitetaan käyttäjöohjaimen luominen lisäämällä projektiin Dynamics AX User Control ohjelmapohja, ja nimeämällä se ”DevEPObjectControl”.ascx-nimiseksi. Kun pohja on luotu, lisätään se heti AOT:n klikkaamalla ”Add to AOT” valintaa käyttäjöohjaimen kontekstivalikosta.

Käyttäjöohjaimen ohjelmointi aloitetaan lisäämällä siihen AxDataSource-komponentti, ja nimeämällä se ”dsDevObject”-nimiseksi ja liittämällä siihen DevObjectsDataSet -Data Set. AxDataSource:n ProviderView on EamAssetMainTable, koska se on Data Setin ainut tietolähde ja rooli saa olla ProviderConsumer, koska käyttäjöohjaimen käyttö halutaan mahdollistaa sekä tarjoajana että vastaanottajana.

Seuraavaksi lisätään käyttäjöohjaimen yksittäisen tietueen käsittelyyn käytetty AxForm-lomake, sekä kenttien näyttämiseen käytetty AxGroup-ryhmä. Sivun asetteluun ja käytettävyyteen käytetään AxMultiSection- ja AxSection-komponentit. AxForm-lomake tulee ylimmän tason komponentiksi ja sen alle lisätään yksi AxMultiSection-komponentti, jonka sisälle luodaan useita AxSection-komponentteja ja jokaisen AxSection-komponentin sisälle lisätään AxGroup-ryhmä.

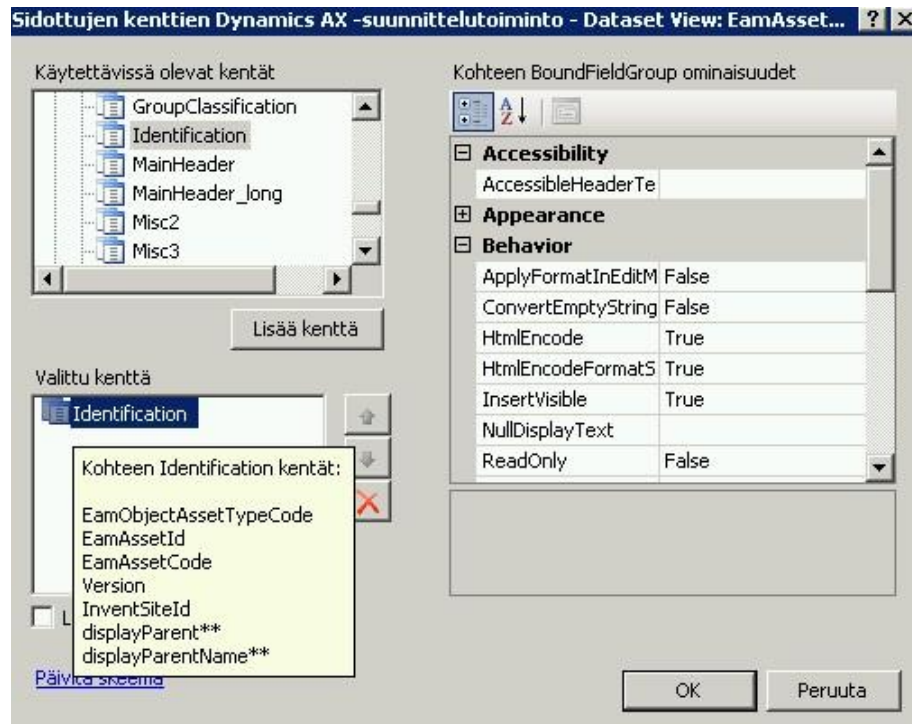
AxForm-lomakkeelle määritetään nimeksi frmDevAsset ja sen Data Sourceksi valitaan dsDevObject-tietolähde. Lisäksi lomakkeen DataMember arvoksi määritellään EamAssetMainTable\_Current-näkymä käytetystä Data Setistä. Oletuksena käyttäjöohjain on katselutilassa, joten lomakkeen DefaultMode on ReadOnly ja AutoGenerateEditButton- sekä AutoGenerateInsertButton-arvot ovat *false*.

Seuraavaksi luodaan lomakkeen ulkoasu lisäämällä AxMultiSection-komponentti ja tämän sisään AxSection-komponentteja. AxSection-komponenteilla luodaan lomakkeelle avattavia ja suljettavia alueita. Oletuksena nämä alueet tulevat olemaan avonaisia, niille voidaan määrittää otsikko ja vielä erillinen SecurityKey, joka vaaditaan kyseisten tietojen näyttämiseen. AxSection-komponenttien sisälle lisätään vielä AxGroup-ryhmät.

Ensimmäiseen AxSection-komponenttiin lisätään tietueen yksilöivät tiedot. AxSection-komponentin otsikoksi Caption-kenttään annetaan AX:n label-kenttä, mikä kuvaa ”Tunnus” tekstiä. Tämä tapahtuu kooditilassa syöttämällä AxSection-komponentille Caption-ominaisuudeksi Caption=”<%\$ axlabel:@SYS36186>”. Nyt AxSection teksti haetaan AX:n käännoستaulusta, ja se on suomenkielisillä käyttäjillä ”Tunnus”.

Ensimmäisen AxSection-komponentin sisälle lisätään nyt AxGroup-ryhmä ”gpTunnus” ID:llä. Tämän jälkeen ryhmään liitetään näytettävät kentät Fields-ominaisuuden liitosominaisuudella. Kentäksi liitetään Identification- BoundFieldGroup-ryhmä, joka on taulutasolla määritelty ryhmä, joka sisältää taulun tunnistiedot. Saman AxSection-komponentin sisälle lisätään myös toinen AxGroup-ryhmä. Tämän ryhmän tarkoituksena on esittää kohteen nimi ja kuvaus, joten ryhmään lisätään Names-

BoundFieldGroup ryhmä sekä Description-kenttä ja annetaan otsikoksi AxLabel, joka on suomeksi ”Pääotsikko”.



*Kuva 7.2: Kenttien valintaikkuna*

Kuvassa 7.2 on esitetty kenttien lisääminen AxGroup-ryhmään. Ryhmään on nyt valittuna Identification-niminen ryhmä, jonka sisältämät kentät näkyvät aukioleavassa työkaluvihjeessä. Oikealle puolella on näkyvillä ryhmän ominaisuudet, josta niitä voidaan halutessa myös muokata. Käytettäessä BoundFieldGroup-tyyppistä ryhmää, päivittyvät ohjaimen kentät automaattisesti, mikäli ryhmää päivitetään AOT:ssa. Toisaalta kyseistä ryhmää käytettäessä yksittäisen kentän tasolle tehtävät muokkaukset ohjelmakoodissa ovat vaikeita.

Seuraavaksi lomakkeelle lisätään muita ryhmiä samaan tapaan kuin tunnus-ryhmä lisättiin, eli luodaan AxGroup-ryhmät AxSection-alueiden sisään ja annetaan näille otsikkotiedot. Ryhmät lisätään omiin AxSectioneihin siten, että Sijainti ja Luokittelutekijät ryhmä on yhdellä alueella, Vastuualueet, Dimensiot ja Tuotetiedot ovat omalla alueellaan.

Kun ryhmät on lisätty, on yksittäisen kohteen katseluun soveltuva käyttäjäohjain valmiina. Katselutilaa varten käyttäjäohjaimen tai Data Settiin ei tarvitse luoda ohjelmakoodia, vaan ne toimivat oletusarvoisesti oikein. Lisäksi oikeustarkistukset ja muut AX:n rajoitteet toimivat .NET Business Connectorin lävitse kuin käyttäjä käyttäisi AX-asiakasta.

### 7.3. Lisäys ja muokkaus ominaisuuksien lisääminen

Kun ulkoasu ja katselutila ovat valmiina, voidaan seuraavaksi käyttäjäohjain laajentaa toimimaan myös lisäys ja muokkaustilassa. Tämän tapahtuu välittämällä käyttäjäohjaimelle POST-menetelmällä käyttäjäohjaimen tila ja tunnistamalla käyttäjäohjaimessa välitetyn viestin perusteella, missä tilassa käyttäjäohjaimen tulee olla. Lisäys ja muokkaus tilaa varten myös Data Settiin lisätään validointimenetely

Tilantunnistus tapahtuu käyttäjäohjaimen ajaessa omaa init-funktiotaan, jota laajennetaan siten, että tila tunnistetaan ja asetetaan AxForm joko katselu-, muokkaus- tai luomistilaan. Tilanvalinnan koodi on esitetty ohjelmassa 7.1.

*Ohjelma 7.1: Tilan valinta GET-metodissa saadun parametrin perusteella.*

```
protected void Page_Init(object sender, EventArgs e)
{
    // Luetaan EPFormAction tila muuttujaan, HTTP:n GET-metodilla
    // lähetetystä mode-muuttujasta.
    EPFormAction FormMode =
        (EPFormAction)
        Convert.ToInt16(this.Page.Request.QueryString.Get("mode"));
    // Valitaan tila parametrin perusteella
    switch (FormMode)
    {
        // Muokkaustila
        case EPFormAction.EditMode:
            this.AxForm1.DefaultMode = DetailsViewMode.Edit;
            this.AxForm1.AutoGenerateEditButton = true;
            break;
        // Luomistila
        case EPFormAction.CreateMode:
            this.AxForm1.DefaultMode = DetailsViewMode.Insert;
            this.AxForm1.AutoGenerateInsertButton = true;
            break;
        // Oletustila on lukutila
        default:
            this.AxForm1.DefaultMode = DetailsViewMode.ReadOnly;
            break;
    }
}
```

Ohjelmassa 7.1 on käyttäjäohjaimen init-metodia laajennettu. Ensimmäisenä luetaan EPFormAction-tyyppiseen muuttujaan luetaan sivun tila HTTP:n GET-metodilla välitetystä mode parametrinä. Tämän jälkeen Switch-käsittelyllä tarkistetaan onko tila muokkaus-, luku- vai lisäystila. Muokkaustilassa AxForm asetetaan Edit-tilaan, ja sille generoidaan muokkaustilan painikkeet. Lisäystilassa asetetaan AxForm lisäystilaan ja generoidaan lisäyspainikkeet. Muissa tapauksissa käytetään oletustilaa, jossa AxForm asetetaan lukutilaan.

Halutaan että Enterprise Portalista voi luoda kohteita, joiden kriittisyys luokka on kolme. Tätä varten kuormitetaan Data Set-olioon tarkastus validateWrite-metodi, jolla tarkastetaan arvon oikeellisuus ennen tietojen tallennusta. Tämä tapahtuu luomalla kyseiseen metodiin vertailu, jolla verrataan onko kyseessä epäkelvo arvo, ja epäkelvon arvon kohdalla asetetaan paluuarvoksi false.

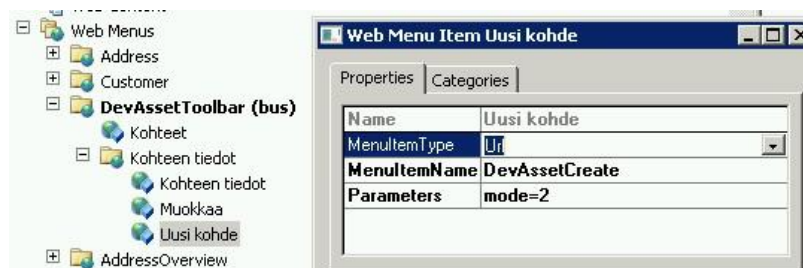
Lisäksi alustetaan jo oletuksena kriittisyysluokan arvoksi kolme, kuormittamalla initValue-metodia siten, että EamAssetMainTable.EamAssetCriticalityCode-kenttään sijoitetaan arvoksi 3. Tämä tapahtuu klikkaamalla AOT:ssa hiiren oikealla napilla Data Setin datasourcen metodit-solmua ja valitsemalla haluttu metodi. Tällöin AX generoi kyseisen metodin kuormituksen rungon, johon voidaan kirjoittaa oma koodi.

## 7.4. Linkit käyttäjäohjaimiin

Ennen kuin voidaan luoda Grid-näkymä, täytyy luoda Web Menu Item, jolla linkitetään luettelonäkymästä yksittäisen tietueen näytölle. Samalla luodaan myös muut esimerkkikohdehallinnan linkit ja työkaluvalikon valikkorakenne. Linkkejä luodessa tulee tietää sivujen osoitteet, joten päätetään etukäteen, että luettelosivu luodaan osoitteeseen ”EAM/Enterprise%20Portal/devEamAssetList.aspx” ja kohdenäkymä osoitteeseen ”EAM/Enterprise%20Portal/devEamAssetView.aspx”.

Luodaan linkit Web Menu Items URL-solmun alle ”Uusi URL”-valinnalla ja määritetään linkkien tunnukseksi DevAssetList luettelolle ja DevAssetView näkymälle. Tämän jälkeen haetaan linkeille sopivat label-tekstit: ”Kohteet” ja ”Kohteen tiedot”. Sekä HelpText arvoon kuvaukset: ”Näytä kohdeluettelo” ja ”Katsele kohteen tietoja”. Tämän jälkeen määritellään vielä URL-arvot aiemmin päätettyjen osoitteiden mukaisiksi. Avainarvoja ei tarvitse esimerkkisovelluksen tapauksessa määrittää, joten linkit ovat valmiit. Tämän jälkeen otetaan vielä kopiot ”Kohteen tiedot” näytön linkeistä, ja luodaan samaan sivuun osoittavat linkit eri otsikoilla kohteen muokkaamista ja lisäämistä varten.

Seuraavaksi luodaan uusi Web Menu, jota tullaan käyttämään kohteentiedot näytöillä. Tämä tapahtuu luomalla uuden Web Menun Web Menus -solmun alle ja nimeämällä se DevAssetToolbar-nimiseksi. Valikon rakenteen toivotaan olevan sellainen, että ensimmäisenä on linkki kohdeluetteloon, ja seuraavana alavasetovalikko kohteen tietoihin, josta voi valita katselun, muokkaamisen tai uuden luomisen.



*Kuva 7.3: Valmiin valikon rakenne, sekä uuden kohteen luomisen linkin ominaisuudet*



Kuvassa 7.3 on esitetty kohdevalikon valikkorakenne AOT:ssa. Kuvasta näkee kuinka linkit on sijoitettu hierarkkisesti. ”Kohteet”, ”Kohteen tiedot”, ”Muokkaa” ja ”Uusi kohde” ovat valikon vaihtoehtoja ja ”Kohteen tiedot”-solmu on tyypiltään Alavalikko. Kuvassa 7.3 on esitetty myös valikon vaihtoehdon ominaisuudet. Valikon vaihtoehtoon on mahdollista valita sen tyyppi, mikä Web Menu Item siinä on, ja mitä parametrejä lisätään linkkiin. Esimerkin tapauksessa on lisätty lisäystilan parametri mode=2.

## 7.5. Grid-näkymä

Seuraavaksi luodaan kohdeluettelon ohjain. Käyttäjäohjaimen lisätään AxDataSource-komponentti, joka viittaa samaan Data Set-olioon kuin kohteen tiedot viittasivat. Tämän jälkeen lisätään käyttäjäohjaimen AxGridView-komponentti, missä näytetään luettelo kohteista. Kohdeluetteloon lisätään näkyville perustiedot kohteesta kuten tunnus, nimi ja kuvaus. Tämän lisäksi tunnuksesta luodaan linkki kohteen tietojen katselunäytölle.

Kenttien lisääminen AxGridView-komponentille tapahtuu kuin kenttien valinta AxGroup-komponenttiin. Avataan kenttävalikko ja valitaan sopivat kentät. Koska AxGridView-näytölle lisätään myös linkki kohteen tietojen katseluun, täytyy tämä kenttä lisätä yksittäin luettelolle, jotta sen muokkaaminen onnistuisi helposti.

### *Listaus 7.2: Linkkikentän XML-koodi*

```
<ynamics:AxHyperLinkBoundField DataField="EamAssetCode"
  DataSet="DevObjectsDataSet_" DataSetView="EamAssetMainTable"
  SortExpression="EamAssetCode" MenuItem="DevAssetView">
</ynamics:AxHyperLinkBoundField>
```

Listauksessa 7.2 on esitetty linkkikentäksi muutetun kentän XML-koodi. Ohjelmasta selviää, että dynamics:AxBoundField-tyyppinen kenttä on muutettu dynamics:AxHyperLinkBoundField-tyyppiseksi kentäksi. Tämä on täytynyt kirjoittaa käsin koodiin, sillä automaattinen kenttien lisäys ei osaa liittää linkkikenttiä. Lisäksi kentälle on lisätty MenuItem="DevAssetView" ominaisuus, joka kertoo mitä Web Menu Itemiä käytetään linkkinä.

## 7.6. SharePoint-sivut

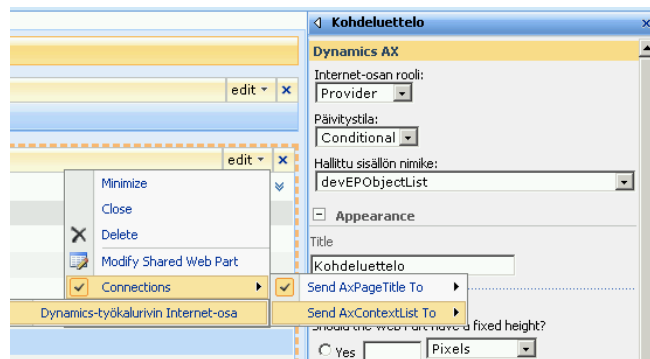
Lopuksi luodaan Enterprise Portal -sivut SharePointiin. Sekä kohdeluettelo- etteä kohdenäkymäsivulle on tarkoitus tuoda kaksi komponenttia: käyttäjäohjain ja työkaluvalikko. Lisäksi käyttäjäohjaimen ja työkaluvalikon välille luodaan yhteys, jotta työkaluvalikko käyttäisi käyttäjäohjaimen aktiivista tietuetta kontekstinaan. Koska molemmat sivut ovat käytännössä samanlaiset, riittää että käymme vain kohdeluettelosivun luomisen lävitse.



**Kuva 7.4:** SharePoint-sivun SiteActions-valikko

Kuvassa 7.4 on SharePoint sivun Site Actions-valikko, joka sijaitsee oletuksena sivun oikeassa laidassa. Enterprise Portal -sivu luodaan klikkaamalla Create-linkkiä ja valitsemalla avautuvasta valikosta Web Pages-valikon alta Web Part Page. Tämän jälkeen aukeaa New Web Part Page-sivu. Nimetään kohdeluettelo ennalta sovitun mukaisesti devEamAssetList-nimiseksi ja valitaan ulkoasuksi ”Full Page, Vertical”, mihin sivulle ei lisätä yleistä pikalinkkivalikkoa. Varmistetaan vielä, että sivu käyttää Enterprise Portal -kirjastoa.

Kun sivu on luotu, aukeaa se muokkaustilassa. Sivulle lisätään tässä tilassa ensimmäiseksi Dynamics User Control Web Partin ja valitaan se näyttämään kohdeluettelon käyttäjäohjainta. Käyttäjäohjaimen yläpuolelle lisätään vielä Dynamics Toolbar Web Part- työkaluvalikko. Tämä valikko asetetaan näyttämään aiemmin luotua kohdevalikkoa.



**Kuva 7.5:** Kohdeluettelon asetukset SharePointissa

Kuvassa 7.5 on esitelty kohdeluettelon web partin ominaisuudet SharePoint ympäristössä. Web part on asetettu Provider-rooliin, sillä sen konteksti välitetään sekä sivun otsikkopalkkiin että työkaluvalikolle. Yhteyden luominen tehdään valitsemalla Connections-luettelosta Send AxContextList To -valinnan alta työkalurivin web part. Tämän jälkeen työkalurivin linkit välittävät kohdeluettelon aktiivisen rivin kontekstina linkkiä klikattaessa.

## 8. ARVIOINTI

Enterprise Portal -käyttäjäohjaimien ohjelmointi osoittautui varsin helpoksi etenkin yksinkertaisissa tapauksissa, kuten esimerkisovellus osoittaa. Yksinkertaisimmillaan tiedon katselu-, muokkaus- ja lisäysnäytöt syntyvät ilman, että ohjelmoijan tarvitsee kirjoittaa riviäkään koodia. Toki todellisuudessa ohjaimet jäävät harvoin näin yksinkertaiselle tasolle joitain luettelonäyttöjä lukuun ottamatta, ja myös taustalle tulee toteuttaa ohjelmakoodia

Ohjelmoinnin vaikeusaste on Enterprise Portalin osalta vaihtelevaa. Itse käyttäjäohjaimia on helppo ohjelmoida Visual Studiassa. Kehittynyt ohjelmointiympäristö ja valmiit työkalut tarjoavat kaiken tarvittavan ulkoasun ja perustoiminnallisuuksien luomiseen, ja ohjelmoija voi keskittyä lähes täysin ohjelmalogiikan suunnitteluun ja toteutukseen.

AOT-palvelimelle tapahtuva ohjelmointi täytyy sen sijaan toteuttaa Microsoft Dynamics AX 2009:n sisällä sen omassa ohjelmointiympäristössä. X++-ohjelmointikielenä ei eroa juurikaan perinteisistä C++ tai Java-ohjelmointikielistä, mutta ohjelmointiympäristö on hankala. Ympäristö on melko alkeellinen, eikä sisällä kuin osittain ennakoivaa tekstinsyöttöä ja taulun kenttien tai funktioiden hakua. Lisäksi saman AOT-solmun sisältämät funktiot aukeavat samalle ohjelmointinäytölle, eikä siten voi samanaikaisesti pitää näkyvissä kahta saman olion eri metodia.

SharePoint-ympäristössä sivujen luominen on helppoa. Muutamalla klikkauksella saadaan aikaan yhtenäiset ja hyvännäköiset sivut valmiita sivupohjia ja web parteja käyttäen. Sen sijaan SharePoint ympäristöön liittyy yllättäviä ongelmia. Sivut täytyy aina viedä erikseen AOT-palvelimelle, jotta ne voidaan asentaa helposti uutteen Enterprise Portal asennukseen. Tämän lisäksi ongelmaksi ovat muodostuneet myös niin sanotusti kummittelevat käyttäjäohjaimet. Mikäli SharePoint-sivulta sulkee epähuomiossa web part-osan, se kyllä katoaa näkyvistä sivulta mutta sen koodi jää talteen, minkä seurauksena web part ilmestyy asennuksen yhteydessä sivulle takaisin näkyviin.

Visual Studion ja SharePointin yhteistoiminta AOT-palvelimen kanssa ei toimi aivan moitteettomasti. SharePoint-sivut eivät siirry automaattisesti AOT-palvelimeen, vaan ne täytyy aina siirtää sinne käsin Web Menu Items-linkkien kautta. Visual Studion taas pitäisi toimia automaattisesti yhdessä AOT-palvelimen kanssa. Tallennettaessa koodi Visual Studiassa sen pitäisi tallentua myös AOT-palvelimelle. Valitettavasti Visual Studio ei muista joka kerta avauksessaan, että käyttäjäohjain on jo viety AOT:een. Eikä näin ollen myöskään osaa tallentaa automaattisesti koodia, ennen kuin sen käsketään lisätä käyttäjäohjain AOT:een. Lisäksi mikäli .NET Business Connector-

yhteydessä tapahtuu virhe, täytyy Visual Studio käynnistää aina uudelleen. Tällainen virhetilanne saattaa johtua monesta syystä. Ongelma saattaa esiintyä esimerkiksi yhteydessä AOT-palvelimelle, virheellisissä .NET Business Connector asetuksissa, tai virheellisessä koodissa AOT-palvelimella esimerkiksi Data Set-oliiossa. Virheilmoitus on aina sama, joten virheen syyhyn on vaikea päästä käsiksi ja aikaa kuluu hukkaan ongelmaa selvitetessä.

Kun halutaan toteuttaa Microsoft Dynamics AX 2009 järjestelmälle web-käyttöliittymää, on Enterprise Portal oikea vaihtoehto. Oman web-käyttöliittymän toteuttaminenkin olisi toki mahdollista .NET Business Connector-rajapintaa käytettäessä, mutta muutamaa erikoistapausta lukuun ottamatta tämä ei vaikuta järkevältä. Erikoistapauksessa joissa oman rajapinnan kautta web-käyttöliittymän toteutusta voisi harkita, on tavoite saada mahdollisimman kevyt ja yksinkertainen käyttöliittymä SharePoint-alustaa tukemattomille päätelaitteille tai palvelimille, tai halu saada vain yksittäinen ominaisuus käytettäväksi webistä käsin.

Palvelimen osalta tätä voidaan pitää lähinnä teoreettisena vaihtoehtona. Microsoft Dynamics AX 2009 -toiminnanohjausjärjestelmän valittuaan voidaan asiakkaan olettaa olevan jo niin vahvasti mukana Microsoftin toimintamalleissa, ettei soveltuvan web-palvelimen ja ilmaisen Windows SharePoint Services-julkaisujärjestelmän asentamiseen pitäisi olla esteitä.

Haluttaessa perus -Enterprise Portal -ympäristöä kevyempää tai yhteensopivampaa ympäristöä. Vaihtoehdoksi jää pyrkiä keventämään SharePoint-ympäristöä teemoilla ja parametrisoinneilla, tai toteuttamalla täysin omat sivut .NET Business Connectorin päälle.

Verrattaessa Enterprise Portalin käyttöä Windows-sovelluksen käyttöön on Enterprise Portalilla monia hyviä puolia, jotka puoltavat sen käyttöä joillain käyttäjillä ensisijaisena ja jopa ainoana alustana. Tärkeimpänä näistä ominaisuuksista on mahdollisuus käyttää Enterprise Portalia ilman erillisen ohjelman asennusta. Tällöin Enterprise Portalia voidaan käyttää miltä tahansa koneelta, jolla vain on nettiyhteys palvelimelle ja vaatimukset täyttävä Internet selain. Toinen Enterprise Portalin käyttöä puoltava tekijä on Microsoftin lisensointimalli, missä yhtäaikaisten Enterprise Portalin käyttäjien hinta on selvästi halvempi kuin yhtäaikaisten Windows-sovelluksen käyttäjien hinta. Enterprise Portalin tullessa halvemmaksi jo kolmella yhtäaikaisella käyttäjällä, ja kolmen yhtäaikaisen Client-käyttäjän hinnalla on mahdollista hankkia 25 yhtäaikaisen Enterprise Portalin käyttäjän pakettin.

Halvemman lisensointinsa ja kevyemmän asennuksensa takia Enterprise Portal on varteenotettava korvaaja myös Windows-sovellukselle. Enterprise Portalia voidaan jopa suositella ainoaksi käyttöjärjestelmäksi niin sanottujen perusduunareiden kohdalla, jotka hakevat töitään töidenhallintajärjestelmästä, kirjaavat työtunteja tai käyttävät AX muihin pieniin toimintoihin. Se on myös hyvä työkalu pääasiallisesti AX:n parissa työskentelevälle käyttäjälle, joskin kaikkia valmiiden moduulien toiminnallisuuksia ei ole viety Enterprise Portal -ympäristöön, eikä näin ollen sovellu ainoaksi työkaluksi esimerkiksi taloushallintoon.

## 9. YHTEENVETO

Diplomityössä on läpikäyty Enterprise Portal käyttäjäohjaimen ohjelmointi, Enterprise Portalin ohjelmakomponentit AOT:ssa, sekä Microsoft Dynamics AX 2009-järjestelmän ja Enterprise Portalin toimintaa ja taustoja yleisesti. Työssä selvisi, että Enterprise Portal -ohjelmointi koostuu kolmesta erilaisesta ohjelmoinnista: käyttäjäohjaimen ohjelmoinnista Visual Studiossa, tietokantalinkin ja taustalogiikan ohjelmoinnista AX:ssa, sekä Enterprise Portal -sivujen ohjelmoinnista SharePoint-ympäristössä.

Visual Studio on ensisijainen ohjelmointiympäristö ohjelmoitaessa Enterprise Portal -käyttäjäohjaimia. Käyttäjäohjaimia on pääsääntöisesti kahtalaatua: yksittäisen tietueen näyttäviä ja tietuekokonaisuuksia näyttäviä ohjaimia. Tietuekokonaisuuksia näyttävät ohjaimet koostuvat pääsääntöisesti pelkästä AxGridView-komponentista, kun taas yksittäisen tietueen tiedot esitetään AxForm-komponentissa, jossa tieto muotoillaan ja jäsenellään AxMultisection-, AxSection- ja AxGroup-komponenteilla.

Käyttäjäohjaimen ulkoasun määrittämisen lisäksi ohjaimen toimintaan voidaan vaikuttaa C#.NET-koodilla. Koodia voidaan kirjoittaa aktivoitumaan käyttäjäohjaimen komponenttien tapahtumien yhteydessä. Tämä koodi voi sisältää käytännössä mitä vain toimenpiteitä, mutta yleensä ne liittyvät tiedon tarkistamiseen, uudelleen rajaamiseen, tai oletusarvojen syöttämiseen.

AX:n ohjelmointiympäristöön luodaan käyttäjäohjaimia Data Set-oliot, joilla kerrotaan mitkä taulut ovat käytettävissä Enterprise Portal -käyttäjäohjaimessa. Tämän lisäksi Data Set-olioon voidaan kirjoittaa tiedon käsittelyyn liittyviä toimenpiteitä ja rajauksia. Data Set-olioita voidaan käyttää helposti useammassa kuin yhdessä käyttäjäohjaimessa, ja ne tulisikin pyrkiä suunnittelemaan mahdollisimman uudelleen käytettäviksi.

AX:n AOT-palvelimelle luodaan myös Enterprise Portalin navigointijärjestelmä, linkit ja linkkilistat. Näiden lisäksi AOT-palvelinta käytetään Enterprise Portalin metatietojen tallentamiseen, kuten myös käyttäjäohjainten ja SharePoint-sivujen tallennuspaikkana, mikä mahdollistaa Enterprise Portalin helpon asentamisen uuden asennuksen yhteydessä.

Enterprise Portal käyttää AX:n AOT-palvelinta .NET Business Connectorin lävitse. Tämän yhteyden kautta kaikki AX-palvelimen rajaukset, turvallisuusasetukset sekä validointimetodit tiedon syötön kannalta ovat automaattisesti Enterprise Portalin käyttäjäohjaimissa myös voimassa. Tämä tarkoittaa siis sitä, että mikäli tarkistukset ja validoinnit ovat kirjoitetut taulutasolle AOT:ssa, ei näitä tarkistuksia tarvitse välttämättä enää kirjoittaa uudelleen käyttäjäohjainten koodiin.

SharePoint-alustalla luodaan Enterprise Portal -websivut. Se ei vaadi erillistä ohjelmointiympäristöään, vaan kaikki tarvittavat säädöt voidaan tehdä SharePoint-sivujen muokkaustilan kautta. Enterprise Portal -ohjelmoinnin kannalta SharePoint onkin vain käyttäjäohjainten julkaisualusta.

## LÄHTEET

- [1] The Microsoft Dynamics AX Team 2009. Inside Microsoft Dynamics AX 2009, USA, Microsoft Press. 689 s.
- [2] Microsoft, Microsoft Dynamics AX 2009 SDK [WWW]. [Viitattu 4.11.2009] Saatavissa [http://msdn.microsoft.com/fi-fi/library/aa496071\(en-us\).aspx?ppud=4](http://msdn.microsoft.com/fi-fi/library/aa496071(en-us).aspx?ppud=4)
- [3] Microsoft, E-learning Introduction, Microsoft Dynamics AX 2009: Introduction - Introduction to Microsoft Dynamics AX 2009. [WWW]. [Viitattu 4.11.2009]. Saatavissa <https://dynamics.microsoftlearning.com/eLearning/courseDetail.aspx?courseId=93996>
- [4] Microsoft PartnerSource, Development I in Microsoft Dynamics AX 2009 [WWW]. [Viitattu 4.11.2009] Saatavissa <https://mbs.microsoft.com/partnersource/communities/training/trainingmaterials/student/course80011.htm?printpage=false>
- [5] Microsoft Dynamics AX 2009 system requirements [WWW]. [Viitattu 6.5.2011] Saatavissa <http://www.microsoft.com/en-us/dynamics/using/ax-system-requirements.aspx>
- [6] Microsoft, Windows SharePoint Services 3.0 Overview [WWW], [Viitattu 17.11.2009] Saatavissa <http://technet.microsoft.com/fi-fi/windowsserver/sharepoint/bb684453%28en-us%29.aspx>
- [7] Holliday, Alexander, Julian, Robillard, Schwartz, Ranlett, Attis, Buenz, Risso 2007, Professional SharePoint 2007 Development, USA, Wiley Publishing, Inc., 716 s.
- [8] Microsoft, AX 2009 EP Dev Cookbook [WWW] [Viitattu 10.9.2010] Saatavissa [http://blogs.msdn.com/cfs-file.ashx/\\_\\_key/CommunityServer-Components-PostAttachments/00-09-94-40-64/AX2009EPDevCookbook.docx](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Components-PostAttachments/00-09-94-40-64/AX2009EPDevCookbook.docx)
- [9] W3C, W3C Schools [WWW] [Viitattu 14.2.2011] Saatavissa <http://www.w3schools.com/>
- [10] Microsoft, Technet [WWW] [Viitattu 14.2.2011] Saatavissa <http://technet.microsoft.com/en-us/library/dd309664.aspx>