LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

Toiminnanohjausjärjestelmän kehittäminen Access 2010 avulla

LAHDEN
AMMATTIKORKEAKOULU
Liiketalouden ala
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Kevät 2014
Päivi Leppänen

Lahden ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

LEPPÄNEN, PÄIVI:

Toiminnanohjausjärjestelmän kehittäminen Access 2010 avulla

Tietojenkäsittelyn opinnäytetyö, 50 sivua, 9 liitesivua

Kevät 2014

TIIVISTELMÄ

Tämä opinnäytetyö tutkii Microsoft Access 2010-tietokantaohjelman hyödyntämistä Microsoft Dynamics AX-toiminnanohjausjärjestelmän aputyökaluna. Tutkimus perustuu suunnittelutieteeseen ja se toteutetaan toimeksiantona kansainväliselle tuotantoalan organisaatiolle. Tutkimustehtävänä on luoda ERP-työkalu, jonka avulla hinnastot saadaan päivitettyä helpommin järjestelmään.

Työkalun tärkeimpänä tehtävänä on parantaa ERP-järjestelmän käytettävyyttä. Työkalun rakentamiseen käytetään Accessia ja Visual Basic -ohjelmointia. Tutkimuskysymyksenä on: miten Access 2010-tietokantaohjelmalla voidaan toteuttaa tuotehintoja ERP-järjestelmään päivittävä työkalu? Työssä on pyritty selvittämään, mitä ominaisuuksia Accessista tarvitaan käyttöön työkalun toteuttamiseksi sekä arvioidaan uuden työkalun tuomia hyötyjä.

Aiheesta ei juuri löydy kirjallista materiaalia. Tutkimuksen tueksi selvitetään aluksi yleisesti toiminnanohjausjärjestelmissä esiintyviä käytettävyysongelmia, jonka perusteella voidaan havaita niiden olevan yleisiä myös muissa yrityksissä.

Tutkimuksen teoriaosuudessa keskitytään kuvailemaan työkalun käyttöympäristöä, kuten käyttäjiä ja teknistä ympäristöä ohjelmistoineen, jonka avulla saadaan tarvittava yleiskuva siitä ympäristöstä, johon työkalu tehdään. Työkalu määritettiin haastattelujen avulla.

Vastauksia tutkimuskysymykseen etsittiin työkalun rakentamisvaiheen tuottaman datan avulla sekä tutkimalla valmiin työkalun tuomaa kehitystä sekä hyötyjä. Tulokset osoittivat, että Access 2010 soveltuu hyvin kyseisen ERP-järjestelmän räätälöintiin. Sen avulla pystytään luomaan käyttäjäystävällisempi ja toiminnaltaan varma työkalu, johon Accessilla on tarjota monia eri ominaisuuksia. Tulosten perusteella voidaan todeta tällaisen työkalun parhaimmillaan hyödyttävän pitkällä aikajänteellä niin järjestelmän käyttäjiä, kuin koko yritystä.

Asiasanat: Access, sovellukset, toiminnanohjaus, kehittäminen

Lahti University of Applied Sciences
Degree Programme in Information Technology

LEPPÄNEN, PÄIVI:

ERP -System Development
Using Access 2010

Bachelor's Thesis in Information Technology 50 pages, 9 pages of appendices

Spring 2014

ABSTRACT

This thesis deals with the Microsoft Access 2010 -database program and how it can be exploited as a tool for the Microsoft Dynamics AX -enterprise resource system. The study is based on design science and it is implemented commissioned by an international organization of the manufacturing industry. The research task is to create an ERP-tool that allows the product price list to be updated more easily to the system.

The aim of the tool is to improve the usability of the ERP-system. Access and Visual Basic Programming are used for building the tool. The research question is how Access 2010 can be used to implement an update tool for the ERP-system. An attempt is made to discover what Access features are needed to implement the tool and to estimate the benefits.

There is not much literature available on the subject. Therefore, a general review about the occurrence of ERP -usability problems is presented and on this basis, it can be seen that similar problems are also common in other companies.

The theoretical part of the thesis focuses on describing the environment of the tool, such as the users and technical environment. This description provides the necessary overview for creating the tool. The tool itself was defined by interviews.

Answers to the research question were sought examining the data of the construction phase of the tool and its benefits. The research results indicated that Access 2010 is well suited to ERP -system customization. It helps to create a more user-friendly and reliable tool, for which Access has many different features to offer. Based on the results it can be concluded that at its best, such a tool can be beneficial to the users of the system and the company in the long term.

Key words: Access, applications, enterprise resource planning, development

SISÄLLYS

1 JOHDANTO

Toiminnanohjausjärjestelmät (ERP – Enterprise Resource Planning) ovat nykypäivänä menestyvän yrityksen peruspilari.
Parhaimmillaan yritysten on katsottu hyötyvän valtavasti ERP-järjestelmästä sen tukiessa johdon päätösten tekoa, helpottaen varainhoitoa, parantaen asiakaspalvelua ja tehden liiketoiminnan tarkemmaksi. Lisäksi sen avulla voidaan saada parannuksia logistiikkaan, inventaarioon sekä vähentää fyysisiä resursseja. (Foster, Hawking & Stein 2004.)

Aiemmin toiminnanohjausjärjestelmiä tunnettiin monimutkaisuudesta sekä heikosta käytettävyydestä, mutta viimeisen kymmenen vuoden aikana kehitystä on tapahtunut huimasti ja uusia versioita kehitetään järjestelmien parantamiseksi. Nykypäivän järjestelmissä onkin saatu parannuksia esimerkiksi eri toimintoihin ja käytettävyyteen, mutta edelleen käyttäjäystävällisyydessä löytyy kehitettävää. Osintsev (2012) on löytänyt käytettävyysongelmiin useita syitä tutkiessaan järjestelmien haasteita, ja näistä yksi esimerkki on järjestelmien logiikan monimutkaisuus. Yksi suhteellisen yksinkertainen prosessi voi vaatia kahlaamaan läpi jopa neljä tai viisi eri lomaketta järjestelmässä, että haluttu toimenpide onnistuu. Myös navigointityökaluissa ja linkeissä esiintyy epäloogisuutta tai näyttö voi olla sullottu liian täyteen tietoa, liian pienellä fontilla. (Osintsev 2012.)
Samansuuntaisia tuloksia on saanut myös Marketvisio (2011) ERP-käyttäjille tekemässään tutkimuksessaan, jossa käyttäjät ovat arvioineet järjestelmän heikommaksi puoleksi juuri järjestelmän huonon käytettävyyden (Rajamäki & Ahtola 2011).

Aladwani (2001) on perustanut tutkimuksensa muihin ERP-järjestelmän haasteisiin ja löytänyt käytettävyysongelmiin syitä myös käyttäjistä itsestään. Koska ERP-projektit aiheuttavat usein työntekijöissä muutosvastarintaa, samalla ne aiheuttavat ongelmia ERP:n käyttöönotossa, kun järjestelmän käyttämistä työn apuvälineenä arastellaan. (Aladwani 2001.)
Käyttäjät siis omalla vastarinnallaan vaikuttavat negatiivisesti uuden järjestelmän sisäistämiseen ja oppimiseen.

Aiempiin tutkimuksiin viitaten voidaan siis sanoa, että käyttäjät eivät usein osaa käyttää järjestelmää tarpeeksi hyvin käyttäjäystävällisyysongelmien tai muutosvastarinnan ilmentymien takia. Myös Järvelä (2012) toteaa tutkimuksessaan, että juuri nämä asiat voivat aiheuttaa ERP-käyttäjille epämiellyttäviä käyttökokemuksia, joista taas pahimmillaan seuraa motivaation puutetta käyttää järjestelmää, ja tämä taas aiheuttaa koko järjestelmän tehokkuuden menetystä (Järvelä 2012).

Monista yleisistä ongelmista huolimatta, ilman kunnon toiminnanohjausta olisi mahdotonta saada irti kaikkea toiminnallista tehokkuutta yrityksessä. Toiminnanohjausjärjestelmät ovat osittain kilpailukeino, mutta myös liiketoiminnan tarpeet muuttuvat jatkuvasti ja näihin organisaatiot pyrkivät vastaamaan toiminnanohjausjärjestelmillä. (Marketvisio 2011).

Toiminnanohjausjärjestelmien tuomat etuudet ovat siis yrityksille niin suuret, että niistä ei haluta luopua, vaan niitä ennemminkin pyritään kehittämään. Että käyttäjille saataisiin motivaatiota käyttää järjestelmää, olisi tärkeää, että ne olisivat käyttäjäystävällisiä ja toisivat käyttäjille positiivisia kokemuksia työskentelystä järjestelmän parissa.

ERP-järjestelmän hankaluus ja sen tuoma motivaation puute on huomattu myös eräässä kansainvälisessä, eteläsuomalaisessa tuotantoyrityksessä. Ongelmaksi tässä yrityksessä on koitunut konsernin yhteisessä toiminnanohjausjärjestelmässä oleva tuotehintojen päivitystyökalu, joka on osoittautunut niin vaikeaksi käyttää, ettei sitä monessa yrityksen myyntiyhtiössä käytetä ollenkaan.

Aiemmin tässä kappaleessa viitattiin havaintoihin järjestelmän hankaluuksista, joissa yksinkertainen prosessi voi vaatia useamman lomakkeen läpikäymisen. Hintojen päivittämisprosessissa on havaittavissa samankaltaista ongelmaa. Oletetaan, että käyttäjä haluaa viedä ensin järjestelmästä hinnastotietokantataulun sisällön Exceliin ja palauttaa sen takaisin järjestelmään. Tämän helpolta kuulostavan prosessin tekemiseen tarvitaan järjestelmätyökalu, jossa lomakkeita avautuu käyttäjälle yhteensä yli kymmenen kappaletta. Tämän lisäksi käyttäjällä on oltava tietoa, minkä nimisestä taulusta, niiden satojen tietokantataulujen joukosta, juuri tämä haluttu taulu löytyy.

Järjestelmän hankaluus on johtanut siihen, että kaikki konsernin yhtiöt eivät syötä ollenkaan tuotehintoja yhteiseen ERP-järjestelmään. Ongelma näkyy esimerkiksi laskutuksessa, kun hintoja kirjataan asiakkaille lähetettäviin laskuihin käsin sen sijaan, että laskut tulostettaisiin suoraan järjestelmästä. Tämä aiheuttaa laskutuksessa virhesyöttäjä, joita joudutaan korjaamaan hyvityslaskuilla. Se aiheuttaa yritykselle lisäkustannuksia, kuten henkilötyövuosien menetyksiä, kun käsin kirjaaminen ja hyvityslaskujen tekeminen vie myyjien aikaa. (Kontio 2013.)
Lisäksi järjestelmästä ei pystytä saamaan irti täyttä tehoa, koska siellä olevat tiedot ovat puutteellisia ja se vaikuttaa siten koko konsernin toimintaan.

Yritys antoi tehtäväksi toteuttaa työkalu, jolla tuotehinnat saataisiin päivitettyä helpommin järjestelmään. Uuden työkalun avulla halutaan innoittaa loputkin konsernin yhtiöt järjestelmän käyttöön ja sitä kautta saada järjestelmä ehyemmäksi ja tehokkaammaksi.

Työkalu toteutetaan Microsoft Access 2010 -tietokantaohjelmalla ja Visual Basic-ohjelmoinnilla tutkimalla samalla Accessin soveltavuutta tällaisen sovelluksen tekemiseen. Tutkimus perustuu suunnittelutieteeseen, mutta pitää sisällään joitain piirteitä myös kvalitatiivisesta tutkimuksesta.

2 TUTKIMUSTEHTÄVÄ

Tämä opinnäytetyö toteuttaa konsernin käyttöön uuden, ERP-järjestelmän käyttöä helpottavan työkalun, jonka avulla konsernin myyntiyhtiöt pystyvät helpommin päivittämään tuotehinnastonsa yhteiseen järjestelmään. Aloite työkalun tekemiseen tuli suoraan yritykseltä.

Konsernissa on tällä hetkellä käytössä Microsoft Dynamics AX 4.0 – toiminnanohjausjärjestelmä, mutta samaan aikaan on menossa järjestelmäversion vaihdosprojekti, jossa AX 4.0:n tilalle tulee uudempi versio Microsoft Dynamics AX 2012. Uusi versio oli tämän opinnäytetyön tekemisen aikaan jo testikäytössä yrityksessä ja sen perusteella osattiin sanoa, että parannusta ongelmakohtaan ei uudessakaan versiossa ole järjestelmäkehittäjältä tullut. Niinpä työkalu tulee käyttöön myös uudessa ERP-versiossa.

Uuden työkalun suurimpana tehtävänä on paikata järjestelmässä esiintyvää vajaakäyttöä ja sitä kautta luoda sen toimintaan samalla tehokkuutta. Työkalun tekemiseen käytetään Microsoft Access 2010 – tietokantaohjelmaa sekä VBA-ohjelmointia. Ohjelman valintaa uuden työkalun tekoon määritti vahvasti yrityksessä oleva Microsoftin toiminnanohjausjärjestelmä, joka on suoraan yhteen sopiva Microsoftin Accessin kanssa. Yrityksen ei tarvitse näin myöskään sijoittaa uuteen ohjelmaan, kun Microsoft Office on jo valmiiksi käytössä koko konsernissa ja samalla tuttu lähes jokaiselle työntekijälle. Tutkimuksen tavoitteena on testata Access 2010 ohjelmiston soveltuvuutta ERP-järjestelmää täydentävänä komponenttina.

Uusi työkalu on hyödyllinen pitkällä aikajänteellä. Tulevaisuudessa voi tulla ajankohtaiseksi asiaksi ottaa käyttöön extranet, jolloin asiakkaat itse syöttävät tilauksensa Internetin kautta ja tilaukset päivittyvät suoraan ERP-järjestelmään. Jotta tilausten syöttäminen ja laskutus tätä kautta onnistuu, on kaikissa yhtiöissä oltava tuotehinnastot päivitettyinä ERP-järjestelmään. Uuden työkalun avulla luodaan pohjaa extrnetin käyttöönotolle, kun yhtiöt saadaan päivittämään hinnastonsa järjestelmään. (Kontio 2013.)

Microsoft Dynamics AX – toiminnanohjausjärjestelmä oli vuonna 2013 Suomen suosituin järjestelmä keskisuurten yritysten keskuudessa eri toimialoilla (Mäntysaari 2013). Opinnäytetyön tuloksena tuleva työkalu on sovellettavissa käytettäväksi myös muualla ja muiden AX:ssa olevien tietojen päivitykseen, ja siksi tämä voi hyödyttää monia muita suomalaisia yrityksiä, joissa kyseinen järjestelmä on käytössä.

2.1 Tutkimuskysymys ja -menetelmät

Tämän opinnäytetyön tarkoituksena on luoda uusi, yrityksen vaatimusten mukainen ERP-työkalu eli sovellus sekä toteuttaa tutkimus, jossa tutkimuskysymyksenä on: miten Access 2010-tietokantaohjelmalla voidaan toteuttaa tuotehintoja ERP-järjestelmään päivittävä helppokäyttötyökalu? Tutkimuskysymys on luonteeltaan kvalitatiivinen eli laadullinen. Sen ratkaisemiseksi etsitään induktiivisen päättelyn avulla vastausta tutkimusongelmaan: mitä ominaisuuksia ohjelmasta tarvitaan käyttöön, että saadaan tehdyksi tavoitteen mukainen ERP-järjestelmän lisätyökalu? Vastausta tähän haetaan tutkimuksen tuottamasta datasta sisältöanalyysin avulla. Tarkastelun kohteena on myös tutkimuksen lopputuote, eli valmis työkalu.

Lopuksi valmista tuotosta arvioidaan viimeisen tutkimusongelman avulla: onko työkalu toimiva ja hyödyllinen? Hyödyllisyydellä tässä tarkoitetaan työkalun yksinkertaisuutta ja helppoutta käyttäjän näkökulmasta katsottuna. Sen todistamiseksi tarkastellaan ensin hintojen päivitystapahtumaa vanhan menetelmän avulla, jonka jälkeen sitä verrataan uuden työkalun avulla tapahtuvaan päivitykseen ja etsitään niiden välisiä eroavaisuuksia ja muita huomioita.

Tämä opinnäytetyö perustuu suunnittelutieteeseen (Design Science). Suunnittelutiedettä käytetään tyypillisesti IT-alan tutkimuksissa. Hevner, March ja Park (2004) määrittelevät suunnittelutieteen olevan ongelmanratkaisua, jolla pyritään luomaan innovaatioita ja sen avulla uusia ajatuksia, käytäntöjä ja teknisiä valmiuksia. Suunnittelutieteellisten tutkimusten tuloksena tulevat IT-artefaktit parantavat organisaation valmiuksia tarjoamalla erilaisia työkaluja ja teorioita seuraten näiden kehitystä ja käyttöä. Artefaktien on tarkoitettu olevan ratkaisuja yksilöityihin organisaatiollisiin ongelmiin, joilla järjestelmän analyysiä, suunnittelua,

toteutusta ja käyttöä voidaan laadukkaasti toteuttaa. Suunnittelututkimukseen sanotaan kuuluvan kaksi lähestymistapaa; rakentaminen ja arviointi. (Hevner, March & Park 2004, 1-5.)

Järviset (2011) toteavat suunnittelutieteen koskevan kolmea suunnittelua: kohteen (artefaktin suunnittelu), toteutuksen (suunnitelman laatiminen artefaktin toteuttamista varten) ja prosessin suunnittelua (metodin kehittelyä suunnitelman ratkaisemiseksi).
Heidän mukaan suunnittelututkimuksen tulosta kutsutaan teknologiseksi säännöksi, joka määritellään yleisen tietämyksen tihentymäksi ja se liittää artefaktin haluttuun tulokseen. Testaamista pidetään tärkeimpänä seikkana tutkimuksen tuloksessa. (Järvinen & Järvinen 2011, 104.)

Hevnerin (2004) mukaan tulokset voidaan jakaa seuraavasti: käsitteistöt, mallit, menetelmät ja toteutukset.
Käsitteistöt tarjoavat sanaston millä ongelmat ja ratkaisut määritellään, kun taas mallit käyttävät käsitteistöjä reaalimaailmassa olevaa suunnitteluongelmaa ja sen ratkaisua varten, eli toisin sanoen ne ilmaisevat käsitteiden väliset suhteet. Menetelmät määrittelevät prosesseja ja tarjoavat ohjeita tehtävän suorittamiseen. Toteutuksilla halutaan osoittaa, että käsitteistöillä, malleilla tai menetelmillä voidaan toteuttaa toimiva järjestelmä, eli artefakti toteutetaan tällöin ympäristössään. (Hevner ym. 2004, 6-7.)

Hevnerin ja Järvisten muotoilemat suunnittelutieteen määritykset tulevat hyvin esille tutkimuksen painottuessa Access 2010 – ohjelman testaukseen uudessa käyttötarkoituksessa sekä samalla uuden artefaktin luomiseen. Tässä toteutuvat suunnittelututkimuksen molemmat lähestymistavat: rakentaminen ja arviointi. Kokonaisuudessaan opinnäytetyö perustuu artefaktin luomiseen ja on luonteeltaan konstruktiivinen, eli haluttu tulos on ennalta tiedossa. Tulokseksi halutaan yrityksen määrittämä toteutus, joka testataan sille luonnollisessa ympäristössä.

Työkalu rakennetaan valmiiksi ja sen toimivuus testataan, mutta käyttöönoton vaiheita tai käyttöönottovaiheen jälkeisiä tuloksia ei tässä työssä tutkita.

2.1.1 Tietojen kerääminen

Työkalun rakentamiseen tarvitaan hyvät lähtötiedot, määritykset sekä sovelluksen rakentamisen aikaiset tarkennukset, jotta sovelluksesta tulisi sopiva yrityksen käyttöön. Nämä tiedot saadaan haastattelemalla yrityksen ERP-päällikköä, joka vastaa yrityksen ERP-järjestelmästä kokonaisuudessaan ja on siksi pätevin henkilö haastateltavaksi tähän tarkoitukseen. Haastattelut toteutetaan avoimina haastatteluina, jotta vastauksista saataisiin mahdollisimman paljon tietoa ja niihin voitaisiin pyytää tarkennuksia.
Saaranen-Kauppinen & Puusniekka (2006) kuvaavat avoimen haastattelun olevan haastattelu, joissa tarkoituksena on keskustella tietystä aihepiiristä haastateltavan ehdoilla. Haastattelut ovat mahdollisimman luonnollisia ja ennaltasuunnittelemattomia eikä niitä sidota tietyn muotoiseksi. Haastateltavan tehtävänä on luoda ilmapiiristä mahdollisimman luonteva ja esittää tarpeen mukaan tarkennuksia tai kysymyksiä haastateltavaa mukaillen. (Saaranen-Kauppinen & Puusniekka 2006.)

Koska kyseessä on tietoteknisen sovelluksen rakentaminen, kaikkien haastattelujen tarpeellisuuden ajankohtaa on ennalta vaikea määrittää, ja siksi ne toteutetaankin tilanteen mukaan tarvittaessa ja teemat muotoutuvat sen hetkisestä tilanteesta.

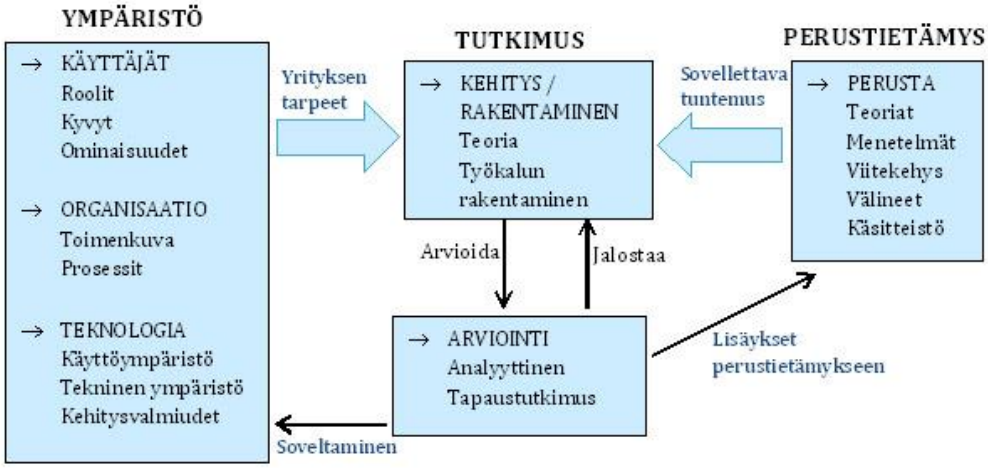
ERP-järjestelmän lisätyökalu toteutetaan toimeksiantona yritykselle, joten tehtävä Access-sovellus tehdään juuri tämän yrityksen tarpeisiin. Tutkimuksesta syntyvä data perustuu tutkijan yrityksessä puolen vuoden työskentelyn aikana keräämiin havaintoihin sekä muistiinpanoihin, jotka on kirjattu ylös työkalua tehdessä. Muistiinpanot on kerätty järjestelmällisesti ja perusteellisesti työkalun suunnitteluvaiheesta lopputestaukseen.

Omien havaintojen lisäksi tutkimuksen tietopohjana käytetään ERP-päällikön haastatteluja, joista ensimmäinen suoritetaan ennen kuin työkalua aletaan rakentaa. Tässä haastattelussa teemana on työkalun tarpeeseen johtavat tekijät ja määritykset työkalulle. Toisen haastattelun teemana on työkalun tietoturva ja se toteutetaan työkalun rakentamisen alkuvaiheessa. Haastattelun avulla määritetään työkalulle käyttöoikeuksiin liittyvät asiat ja niiden toteutustekniikat. Näiden lisäksi käytetään tarvittaessa suppeampia haastatteluja työkalun rakentamisen aikana, että saadaan täydentäviä vastauksia vastaan tulleisiin kysymyksiin.

2.2 Viitekehys

Opinnäytetyö rakentuu Hevnerin (2004) kuvaileman tutkimustieteen käsitteistön mukaan ja siitä muodostuu viitekehys.

Aluksi työssä keskitytään luomaan perustaa tutkimukselle, etsimällä teoriaa ja taustatietoa, valitsemalla tutkimukseen menetelmät sekä tutustumalla tutkimustieteen käsitteistöihin.
Toiseksi kuvaillaan yrityksen sisäisiä valmiuksia, kuten organisaatio- ja teknologiaympäristöä. Teknologiaympäristön kuvauksessa saadaan esitettyä tarvittavat tekniset apuvälineet, joiden avulla hahmotetaan työkalun vaatimia teknisiä valmiuksia. Näiden ja haastattelujen perusteella muodostetaan kokonaiskuva yrityksen tarpeista ja vaatimuksista, jotka vaikuttavat työkalun rakentamiseen.
Kolmannessa vaiheessa paneudutaan itse työkalun rakentamiseen ja samalla etsitään tutkimuskysymykseen vastauksia. Työkalua rakennetaan sille luonnollisessa ympäristössä jatkuvasti arvioiden ja jalostaen työkalua paremmaksi. Seuraavassa kuviossa nähdään Hevnerin käsitteistön pohjalta tehty viitekehys tähän opinnäytteeseen sekä yhteys näiden käsitteiden välillä (KUVIO 1).



KUVIO 1. Opinnäytetyön viitekehys

2.3 Organisaatio

Opinnäytetyön toimeksiantajana on suuri rakennusteollisuusalan yritys Peikko Group Oy, joka valmistaa erilaisia teräsosia valmistuviin rakennuksiin. Yrityksen kotipaikka on Etelä-Suomessa, jossa sijaitsee konsernin pääkonttori. Toiminnaltaan tämä yritys on hyvin kansainvälinen – myyntiyhtiöitä sillä on noin 30 maassa ja tuotteita valmistetaan yli kymmenessä tehtaassa ympäri maapallon. Jokaisella tytäryhtiöllä on oma toimitusjohtaja, mutta konsernin omistajat ovat suomalaisia. Yhtiöt ovat tiiviissä yhteistyössä keskenään.

Yrityksen suurin markkina-alue on Suomi ja muu Eurooppa, mutta laajan kansainvälisen yhtiöverkoston mahdollistamana liiketoimintavolyymi on kasvussa myös muissa maanosissa. Työntekijöitä yrityksessä on yhteensä yli 1000, joista työskentelee Suomessa noin 250 henkilöä. (Peikko 2014.)

Yhtiön strategisena tavoitteena on olla alalla johtavassa markkina-asemassa ja kasvaa kannattavasti. Toiminnan perusajatuksena on innovatiivisten ratkaisujen tarjoaminen asiakkaille, jotta he saavat kehitettyä rakennusprosessejaan paremmiksi. Toimintaa ohjaaviksi arvoiksi yritys määrittelee muun muassa jatkuvan parantamisen, tuloshakuisuuden sekä asiakastyytyväisyyden. (Peikko 2014.) Tämän opinnäytteen taustalta voidaankin löytää nämä kolme arvoa, koska uuden työkalun avulla haetaan parannusta järjestelmään ja järjestelmän toimivuus on yhteydessä koko liiketoimintaan ja asiakastyytyväisyyteen.

2.3.1 Organisaatioprosessit

Tämä teollisuusalan yritys haluaa pitää toimitusketjunsa erityisen hyvin hallinnassaan. Yritys ostaa tuotteisiin käytettävät teräkset suurimmaksi osaksi suoraan terästehtaista. Tuotteensa se valmistaa omissa tehtaissaan, jotka ovat toiminnaltaan tehokkaita ja nykyaikaisia. Tehtailta valmiit tuotteet siirtyvät lähempänä asiakasta oleviin varastoihin. (Peikko 2014.)

ERP-järjestelmän tehtävänä on tukea näitä prosesseja ja tätä kutsutaan tuotannonohjaukseksi. Järjestelmään syötetään materiaalin ostot ja sen kulkeminen tehtaassa materiaalista valmiiksi tuotteeksi. Järjestelmästä löytyvät tuotantoon liitty-

en esimerkiksi materiaaliluettelo, nimikkeet, niiden varastosijainnit sekä osto- ja myyntihinnat. Kaikki nämä johtavat järjestelmässä asiakkaan luokse, joka on tilannut tuotteen.

Käytännössä konsernin myyntiyhtiöt ostavat tuotteensa jostakin konsernin tehtaasta ja myyvät niitä eteenpäin omalle markkina-alueelle. Myyntihinnat määritetään oman yhtiön sisällä, joten joka maan yhtiöllä on omat hinnastonsa, joiden kaikkien haluttaisiin löytyvän konsernin yhteisestä toiminnanohjausjärjestelmästä.

3 KÄYTTÖYMPÄRISTÖ

Ympäristö, johon sovellus tulee käyttöön, määrittää pitkälle työkalun toiminnallisuuksia ja mitä ominaisuuksia Accessista tarvitaan käyttöön sen tekemiseen. Tämän lisäksi on tiedettävä, ketkä sovellusta käyttää, millaisessa tarkoituksessa ja mitä yritys toivoo työkalulta. Tässä luvussa käydään ensin läpi työkalun käyttötarkoitusta sekä käyttäjiä ja lopuksi keskitytään kuvailemaan työkalun tekemiseen vaadittua teknistä ympäristöä eri käsitteineen.

3.1 Työkalun käyttötarkoitus ja käyttäjät

Aiemmassa luvussa kuvailtiin tuotantoprosessia sekä ERP:ssä olevan hinnaston muodostumista pääpiirteittäin. Vähintään kerran vuodessa organisaation yhtiöissä tapahtuu näiden hinnastojen päivitystapahtuma, jossa päivitetään tuotehinnastot ja kukin yhtiö vie uudet, päivitetyt myyntihintansa konsernin yhteiseen ERP-järjestelmään.

Työkalun tarkoitus on, että eri yhtiössä myyjät pystyvät lisäämään helpommin omia myyntihintojaan järjestelmään ja tarkistamaan konsernin yhteisistä hintatiedoista sen hetkiset suositellut myyntihinnat. Työkalun tulee olla jokaisen työntekijän saatavilla yhteisellä palvelimella ja helppo käyttää, mutta samalla sen on oltava suojattu siten, että sitä pääsevät käyttämään vain he, ketkä näitä hintoja järjestelmään syöttävät. (Kontio 2013.)

Työkalu hakee käyttäjälle ostohinnat siitä yrityksestä, mistä yritys ostaa, joka on siis saman konsernin yhtiö. Tämän hetkiset myyntihinnat työkalun on haettava omasta kotiyrityksestä, ja tuoda ne käyttäjän näytölle katseltavaksi ja päivitettäväksi. Ostohintoja käyttäjä voi käyttää apuna määritettäessä uutta myyntihintaa. Lopuksi käyttäjä saa työkalun avulla palautettua päivitetyt hinnat takaisin järjestelmään helposti ja virheettömästi.

Uuden työkalun käyttäjät tulevat olemaan pääasiassa eri maiden yhtiöiden myyjiä. Tämä määrittää sovellukselle kielen, joka on englanti. Myyjät ovat tottuneita Microsoftin Excelin käyttäjiä ja käsittelevät hinnastojaan sen avulla. Päivitetyt hinnat kuuluisi olla joka yhtiöllä myös ERP-järjestelmässä. Kuten on aiemmin mainittu,

kaikki yhtiöt eivät ole tähän ryhtyneet, koska ovat kokeneet järjestelmän importointityökalun liian hankalaksi ja ovat päätyneet hallinnoimaan hinnastojaan pelkästään Excelin avulla.

3.2 Tekninen ympäristö

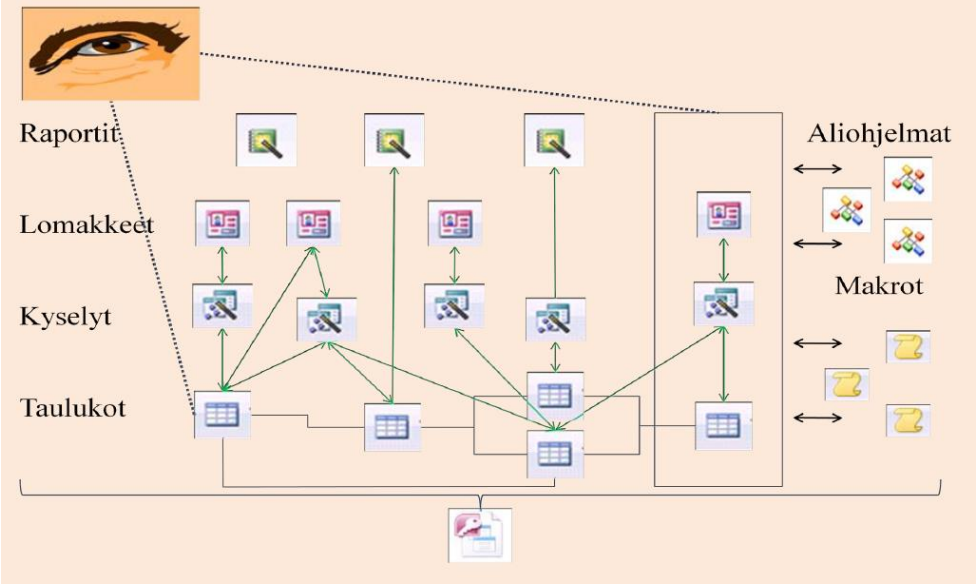
Kuten edellisessä kappaleessa mainittiin, yrityksellä on käytössä koko konsernin yhteinen ERP-järjestelmä, jonne tietokantoihin syötetään lähes kaikki tieto liittyen yrityksen toimintaan. Yrityksen ERP-järjestelmän toimivuudesta vastaa Suomen toimipisteessä työskentelevä ERP-tiimi, joten järjestelmän ylläpito sekä kehitys tapahtuvat tätä kautta.

Järjestelmä sijaitsee yrityksen omalla palvelimella, jonne on pääsy jokaisesta tytäryhtiöstä etätyöpöydän kautta, käyttäjätunnuksen ja salasanan avulla. Jokaiselle toimistotyöntekijälle luodaan oma käyttäjätunnus siinä vaiheessa, kun henkilö tulee yritykseen töihin. Palvelimet on jaettu turvallisuussyistä siten, että osa toiminnoista vaatii vielä erillisen salasanan ja tunnuksen, jotta sisään kirjautuminen onnistuu ja joita ei usein ole tiedossa kuin IT-asioista vastaavilla. Tällaisia ovat esimerkiksi järjestelmien ylläpitoon liittyvät asiat sekä SQL-tietokannat.

3.2.1 Tietokannat

Nykypäivän yrityksissä tietokannat ovat suuri osa menestyvän yrityksen toimintaa. Ilman toimivia tietokantoja ajantasaista tietoa on vaikea pitää yllä. Tietokantoja käytetään hyväksi tietojen jäsentämisessä sekä yhteenvedoissa, joissa halutaan tietoa ulos esimerkiksi raporttien muodossa. Lisäksi tietokantoja käytetään tietojen päivittämiseen, kuten vanhentuneen tiedon poistamiseen tai muuttamiseen ja uuden lisäämiseen. Tietokantoihin voidaan määritellä myös käyttäjäoikeuksia, joita eri sovelluksista voidaan hakea ja näin antaa sisään kirjautumisoikeuksia määritetyille tahoille.

Kun tietokanta on oikein rakennettu, se on tarkka, tieto on täsmällistä ja se on loppukäyttäjän näkökulmasta helposti käsiteltävissä (KUVIO 2).



KUVIO 2. Tietokanta loppukäyttäjän näkökulmasta (Keinonen 2010)

Tässä tutkimuksessa tietokanta on jo valmiiksi olemassa toiminnanohjausjärjestelmässä ja sieltä haetaan määrättyä tietoa Access-sovellukseen käsiteltäväksi. Työn edetessä kiinnitetään erityistä huomiota järjestelmän tietokannan päivitykseen, että tietokantaan takaisin menevä tieto on oikeassa muodossa ja siirrettävissä päivitettynä takaisin yrityksen yhteiseen järjestelmään.

Opinnäytetyössä käytettävä Microsoft Access pohjautuu SQL-kieleen (Structured Query Language). SQL:n avulla saadaan luotua toimivia tietokantoja, ylläpidettyä tietokannoissa olevia tietoja ja käsittelemään oikeuksia (Huotari 1999–2013, 11).

Erilaisten SQL-komentojen avulla suoritetaan tehtäviä näissä tietokannoissa. Siksi Työkalua tehdessä käytetään Microsoft Server Management Studio -ohjelmaa, jonka avulla saadaan hallittua SQL-palvelimen tietokantoja graafisesti. Näihin tietokantoihin tallennetaan tietoja, joita tarvitaan tietoverkkojen hallinnoimisessa, kuten oikeuksia. SQL-palvelimen käyttäjäryhmä on hyvin pieni ja sinne vaaditaan erillinen salasana ja tunnus. Siksi työkalun rakentamisessa käytetään SQL-tietokantoja sovelluksen käyttöoikeuksia määritettäessä, koska se on huomattavasti luotettavampi tapa, kuin oikeuksien määrittely suoraan Accessissa.

Jotta Access pystyisi hakemaan tietoa SQL-palvelimelta mahdollisimman tehokkaasti ja varmasti, on hyvä käyttää serverille tallennettua proseduuria. Proseduuriin tallennetaan SQL-lause, jonka avulla haetaan haluttuja tietoja serverin tieto-

kantataulusta. Proseduurien käytöllä on sanottu olevan useita etuja, joista tärkeimpänä tietoturvan lisääntyminen, kun serverille yhteydessä oleva sovellus tai sen käyttäjä pääsee käyttämään ainoastaan tallennettua proseduuria ja näin muut tietokantataulut ovat turvassa. Proseduurien avulla saadaan irti suurempaa tehoa sekä verkkoliikenteen vähenemistä. (Moilanen 2002.)

Työssä kiinnitetään huomiota SQL:n niihin asioihin, joita tarvitaan kun työkalun käyttäjämääritykset haetaan erillisestä yrityksen tietokannasta. Tietokantojen selaukseen käytettävää tietokantojen hallintaohjelmaa ei sen sijaan työssä laajemmin esitellä. Tietokantahallintaohjelman avulla määritetään tehtävälle sovellukselle käyttäjäoikeuksia, joten työssä käydään läpi, minkälainen tietokantataulu vaaditaan oikeuksien määrittämiseksi ja kuinka ne otetaan käyttöön Access-sovelluksessa. Tämän lisäksi kieltä tarvitaan jäsentämään ERP-järjestelmästä haettua dataa sovelluksessa.

3.2.2 Microsoft Dynamics AX

Yritys jolle työkalu toteutetaan, käyttää siis toiminnanohjaukseensa Microsoftin Dynamics AX-toiminnanohjausjärjestelmää. Järjestelmän tarkoitus on ohjata yrityksen toimintaa helposti ja luotettavasti. Järjestelmää voidaan kutsua suureksi tietopankiksi, minne varastoidaan tai sieltä tulostetaan tietoa. Tuotantoyrityksessä järjestelmän tietokantoihin syötetään tietoja esimerkiksi toimittajista, asiakkaista, tilauksista, työntekijöistä, tuotteista, materiaaleista ja tuotantoketjuista. Järjestelmä on koko konsernin yhteinen, mikä tarkoittaa sitä, että jokaisen konsernissa olevan yrityksen tiedot tallentuu samaan järjestelmään etätyöpöytäyhteyden avulla. Järjestelmä on kuitenkin jaettu maittain, eli sen perusteella, missä konsernilla on yhtiö. Se tekee yhteisen järjestelmän käytön selkeämmäksi, koska silloin käyttäjä näkee ainoastaan oman yrityksensä tiedot. Tietokannoissa nämä yritykset erotellaan yhtiötunnisteen (dataarea id) perusteella.

MS Dynamics AX toiminnanohjausjärjestelmä soveltuu suurille ja keskisuurille yrityksille sen monipuolisten toiminnallisuuksien takia, jotka pitävät sisällään esimerkiksi talouden hallinnan ja yrityksen eri toimintojen johtamisen. Dynamics AX rakentuu eri moduuleista ja työkaluista. Järjestelmä räätälöidäänkin

aina jokaisen yrityksen tarpeisiin ja eri ominaisuuksia voidaan lisätä tarpeen vaatiessa.
Dynamics AX:n vahvuuksiin kuuluu, että se on perusnäkymältään hyvin samantyyppinen muiden Microsoftin ohjelmistojen kanssa ja siten monelle käyttäjälle helpompi hahmottaa. Lisäksi se mahdollistaa, että Microsoftin ohjelmistoja, kuten Exceliä, Accessia ja Microsoft SQL Serveriä, voidaan käyttää toiminnanohjausjärjestelmän rinnalla. (Microsoft 2012.)

Yritysmaailmassa toiminnanohjausjärjestelmiä on käytössä useilta eri valmistajilta, mutta tässä tutkimuksessa keskitytään Dynamics AX:n ja Accessin yhteensovittamiseen, joten muiden valmistajien järjestelmät jäävät tämän tutkimuksen ulkopuolelle.

3.2.3 Microsoft Access 2010

Microsoft Access on tietokantojen käsittelyohjelma ja sovelluskehitin, joka kuuluu tunnettuun Microsoft Office -tuoteperheeseen. Se on relaatiotietokannan hallintajärjestelmä, jossa tietoja käsitellään tietotauluina, jotka kytketään toisiinsa kenttien avulla. Accessissa tieto on saatavilla muistakin Windows-sovelluksista. Ohjelma toimii käyttöliittymänä SQL-tietokantoihin ja siihen voidaan luoda yhteys muistakin ohjelmista ODBC-rajapinnan avulla, josta kerrotaan enemmän myöhemmin tässä luvussa.
Accessin toimintapääperiaatteisiin kuuluu, että sillä voidaan käsitellä tietoa esimerkiksi kyselyiden ja suodatusten avulla. Tieto on helposti päivitettävissä, poistettavissa ja muokattavissa. (Keinonen 2010, 8.)

Accessilla pystytään tekemään sovelluksia siten, että loppukäyttäjä käsittelee tietoa vain sitä varten rakennetun lomakkeen avulla ja hänen ei tarvitse koskea itse tietokantaan. Näin taataan tietojen päivittyminen oikein ja estetään virheet sovelluksen toiminnassa. Lomakkeet saadaan muokattua helposti juuri tiettyyn tarkoitukseen sopivaksi ulkoasun, tekstilaatikoiden ja painikkeiden avulla ja siitä tallennetaan vain rakenne. Painikkeisiin saadaan ohjelmoitua taustalle proseduureja, jolloin käyttäjän painaessa painiketta, taustalla tapahtuu sarja toimintoja käyttäjän huomaamatta. Näitä huomaamattomia toimintoja ovat esimerkiksi kyselyt, joilla tietokantadataa ohjataan.

Kyselyt ovat Accessissa tärkeimpiä toimintoja. Kyselyiden avulla on helppo hakea juuri oleellinen tieto, koska hakua voidaan rajata merkkijonon avulla ja kyselyihin voidaan valita mukaan vain osa tietotaulun sarakkeista. Kyselyillä pystytään löytämään kaksoiskappaleet tai virheelliset tiedot ja niihin pystytään tekemään laskukaavoja. Access tallentaa kyselyistä vain määritykset, joten samaa kyselyä voidaan käyttää yhä uudelleen. (Keinonen 2010, 17.) Kyselyn tulosjoukko voidaan tallentaa omaksi tauluksi.

Accessissa kyselyitä on erilaisia ja niitä käytetään tilanteesta riippuen. Poistokyselyn avulla tyhjennetään taulukoita tai tiettyjä rivejä niistä antamalla ehtoja. Päivityskyselyjen avulla pystytään poistamaan yksittäisiä arvoja tai muuttamaan niitä halutunlaisiksi. Liittämiskyselyitä käytetään silloin, kun halutaan yhdistää kahden tai useamman taulun tietoja keskenään. Lisäksi on valintakysely, jota käytetään oikeastaan vain tietojen etsimiseen sekä tarkasteluun, kun halutaan löytää suuresta tietomassasta juuri ne tietyt tiedot.

Viidettä kyselytyyppiä tarvitaan, kun Accessilla ollaan yhteydessä SQL-tietokantaan. Silloin käytetään läpivientikyselyitä (Pass-through query), joiden avulla lähetetään käskyjä suoraan ODBC-tietokantapalvelimeen. Nämä kyselyt eroavat aiemmin mainituista siten, että läpivientikysely tehdään suoraan toiselle palvelimelle, eikä Accessin oma tietokantamoduuli käsittele näitä kyselyjä. (Microsoft 2014.)

Lomakkeen tekstilaatikot ovat hyvä apuväline, kun tietoa halutaan hakea tietokannasta tietyn ehdon perusteella. Ohjelmoija määrittää tässäkin tapauksessa taustalle käskyn, joka suoritetaan, kun käyttäjä on antanut tiedon. Haetut tiedot voidaan tulostaa käyttäjälle siten, että ne ovat käyttäjän saatavilla, mutta ei suoraan tietokantaan muokattavissa. Tällä keinolla pystytään estämään tässä kappaleessa aiemmin mainitut virhesyötöt.

3.2.4 Visual Basic for Application

Accessista löytyy valmiina ohjelmointikieli, Visual Basic for Application (VBA) ja sen koodin kirjoittamiseen editori. Visual Basic on olio-ohjelmointikieli, jonka

avulla pystytään luomaan proseduureja, eli toimintosarjoja ja pystytään tätä kautta hallitsemaan sovelluksen toimintaa. Toimintosarjat voivat olla sarja aliohjelmia tai funktioita, joita kutsutaan Accessissa moduuleiksi (Keinonen 2010, 18). Aliohjelmaksi sanotaan toimintosarjaa, joka suoritetaan Access-lomakkeen ohjausobjektiin liitetyn tapahtuman yhteydessä, kuten käyttäjän klikatessa tiettyä painiketta (Keinonen 2010, 189). Yhteen moduuliin voidaan liittää useamman lomakkeen ohjausobjektin toimintasarjat.
Käytännössä toimintosarja voi olla esimerkiksi tietokantakyselyitä, tiedon tarkistuksia, yhteyksien luomista muihin ohjelmiin, erilaisten automaattisten ohjeiden antamista käyttäjälle sanomaruutujen avulla ja niin edelleen. Toimintosarjoihin pystytään upottamaan myös SQL-komentoja tietokantojen käsittelyyn.

Työkalua tehdessä käytetään Data Access Object (DAO) -objektimallia, jolla voi manipuloida tietokannan rakennetta ja sisältöä käyttämällä kyseisen mallin objektikirjastoa, joka löytyy valmiiksi Accessista.
On myös olemassa vaihtoehtoinen objektimalli nimeltään ActiveX Data Objects (ADO). ADO:a voidaan pitää joko vaihtoehtoisena tai täydentävänä lähestymistapana manipuloida tietoja.
Ensin DAO oli ainoa tapa yksityiskohtaisten tietojen manipulointiin ohjelman ohjauksessa. Myöhemmin ADO tuli saataville ja oli joidenkin ihmisten mielestä parempi vaihtoehto, koska se tarjosi hieman paremman suorituskyvyn. Kuitenkin ADO:sta ei tullut hallitseva valinta, koska DAO oli jo hyvin vakiintunut ja nykyisin DAO:a pidetäänkin avainteknologiana Access-tietokantaan. (Couch 2011,161–162.)

VBA-ohjelmointia tehdessä täytyy siis tietää minkä kirjaston objekteja käyttää. Yksi paljon käytetty DAO-objekti on nimeltään Recordset, jota tässäkin työkalussa tarvitaan. Recordsettiin tallennetaan tietoa ja sen eri metodien avulla, kuten ”MoveNext” tai ”MoveFirst”, pystytään liikkumaan näissä tietueissa tai käsittelemään siellä olevaa tietoa. Lisäksi Recordsetillä on muuttujia, joista yksi tässäkin työkalussa käytetystä on BOF (beginning of the recordset). BOF:in arvo on joko ”true” tai ”false” ja sen avulla pystytään esimerkiksi vertaamaan eri tietoja keskenään.

3.2.5 Open Database Connectivity

Kyselyiden luomiseen eri ohjelmien välille tarvitaan käyttöön ohjelmointirajapinta. Tässä sovelluksessa käytetään Open Database Connectivity (ODBC) -yhteyttä, joka on avoin rajapinta tietokantaohjelmille ja sen avulla luodaan yhteys ERP-järjestelmän tietokantoihin sekä SQL-palvelimelle. Ohjelmistorajapinnan kautta kyselyt saadaan tehtyä siis suoraan sinne, mistä tietoa haetaan. ODBC-tietolähteen nimi (DSN) on oltava määritettynä tietokoneen ohjauspaneelin kautta, että tietokantataulujen hakeminen toiminnanohjausjärjestelmästä ja sovellus toimivat. Sen avulla määritetään, miten yhteys muodostetaan tiettyyn palveluun ja ovat tietokone- ja käyttäjäkohtaisia.

4 TYÖKALUN MÄÄRITYKSET JA RAKENTAMINEN

Tässä kappaleessa syvennytään itse sovelluksen rakentamiseen ja sen tuottamaan tutkimusdataan. Jotta vastaukset molempiin tutkimusongelmiin saataisiin, työkalun rakentaminen jaetaan kolmeen vaiheeseen.

Ensimmäisessä vaiheessa tutustutaan pääpiirteittäin järjestelmän omalla päivitystyökalulla tehtävään päivitysprosessiin, että ymmärrystä uuden työkalun hyödyllisyydelle sekä parannuksia hintojen päivitysprosessiin saadaan.
Toisessa vaiheessa päästään määrittelyvaiheeseen, missä hahmotellaan mahdollisimman tarkasti yrityksen toiveet työkalun toiminnasta. Määritykset on tehty haastattelujen tuottaman datan pohjalta siten, että heti rakentamiseen ryhtyessä oli selkeä kuva siitä, minkälaisen työkalun yritys haluaa.
Viimeisessä vaiheessa kuvaillaan artefaktin toteutuksen pääkohtia ja toteutuskeinoja ERP-työkalun rakentamiseen Access 2010 avulla.

Näiden kolmen vaiheen avulla etsitään tutkimusdatan joukosta seikkoja, joiden avulla pystytään myöhemmin pohtimaan valmiin artefaktin hyödyllisyyttä ja analysoimaan muita esille nousevia asioita.

4.1 Hintojen päivitys vanhan työkalun avulla

Kun ERP:ssä olevia tuotehinnastoja halutaan päivittää järjestelmän omalla työkalulla, prosessi on monivaiheinen (LIITE 1). Näiden vaiheiden avulla AX luo ensin halutusta tietokantataulusta Excel-tiedoston, jossa käyttäjä pääsee muokkaamaan hinnat ja tämän jälkeen se siirretään takaisin järjestelmään.

Ennen kuin hinnat saadaan Exceliin, käyttäjän tarvitsee etsiä järjestelmästä työkalu, jolla luodaan siirtoon tarvittava järjestelmäkansio, jota kutsutaan tässä myös Journal-kansioksi. Järjestelmä vaatii tämän tiedon aina, kun päivitettyä dataa siirretään järjestelmään.
Kun Journal-kansio on luotu, on etsittävä tietojen vientiin tarkoitettu velho, eli ohjattu toiminto, jonka avulla saadaan luotua työkirjaan Excel-taulukko käyttäjän valitsemalle tietokantataululle. Velhon ensimmäinen lomake kertoo lyhyesti työ-

kalun toiminnasta ja toisessa lomakkeessa määritetään uudelle tiedostolle nimi. Sen jälkeen avautuu kolmas lomake, jossa valitaan järjestelmästä taulu, josta tiedot halutaan Exceliin. Valittavana on satoja tauluja, joten käyttäjän täytyy tietää, minkä nimisessä taulussa halutut tiedot sijaitsevat. Kun taulut on valittu, AX prosessoi tiedot neljännen lomakkeen avulla ja viides käskee käyttäjää luomaan laskentataulukon kentät valintaruutujen avulla, joista himmennetty valintaruutu tarkoittaa pakollista kenttää tai yksilöivä indeksiä. Keltaiset riippulukkokuvakkeella merkityt kentät ovat järjestelmäkenttiä eikä niitä ole oletusarvoisesti valittu. Kuudennesta kohtaa ohjeistetaan käyttäjää valitsemaan, luoko velho mallin sisältävälle Excel-taulukolle tuontimääritysryhmän, jota voidaan käyttää tuontivaiheessa.

Tämän jälkeen aukeaa seitsemäs, datan vientilomake kolmella valintapainikkeella. Tässä valittavana ovat kohdat: vie data, luo tukevat taulut Excel-työkirjaan tai luo Excel-projektitaulu.
Kahdeksas, eli viimeinen vaihe ilmoittaa prosessin päättyneen. Nyt käyttäjä voi mennä etsimään Excel-työkirjan sieltä, minne hän määritti sen haluttavan luotavan ja muuttaa hinnaston siellä.

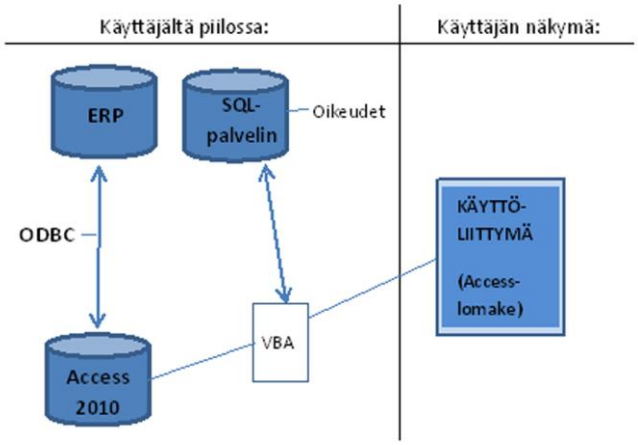
Kun käyttäjä haluaa syöttää uudelleen hinnoitellut tuotteet takaisin järjestelmään, hän etsii toisen, tietojen tuontiin tarkoitetun työkalun. Tuontityökalun ensimmäisessä vaiheessa kysytään määritysryhmää ja etsitään tiedosto, mikä halutaan tuoda. Sen jälkeen hän joutuu etsimään vielä Journal-työkalun, jonka avulla ihan ensimmäiseksi luotiin oma Journal-kansio tietojen siirrolle. Tämän avulla siirretään tieto järjestelmään. Painikkeesta ”lines” tulee yleiskatsaus tuotavasta datasta ja uusi lomake, mistä löytyy painike datan lähetykseen takaisin järjestelmän tietokantatauluun.

Tästä voidaan laskea, että koko hintojen päivitysprosessi järjestelmän omilla valmiuksilla tarvitsee yhteensä kolmen eri työkalun käyttöä. Ensimmäinen työkalu on Journal-kansion luomiseksi ja tietojen siirtämiseksi tietokantatauluun takaisin, toinen tietojen viemiseksi Exceliin ja kolmas tietojen tuomiseksi takaisin järjestelmään. Käyttäjän tarvitsee käydä läpi tässä prosessissa yhteensä 12 lomaketta neljän eri vaiheen aikana.

4.2 Käyttö- ja vaatimusmäärittely uudelle työkalulle

Uuden työkalun tarvitsee olla siis käyttäjälle helpompi käyttää kuin järjestelmän oma päivityskeino. Toisaalta se tarvitsee taustalle paljon toimintoja, joilla dataa saadaan käsiteltyä ja sovellus toimii. Toimintojen on tapahduttava joka kerran sovelluksen avaamisen yhteydessä samalla kaavalla ja virheettömästi. Jotta tämä saadaan toteutettua, työkalua tarvitsee ohjelmoida siten, että suurin osa toiminnoista tapahtuu käyttäjältä piilossa.

Työkalun ohjelmointi toteutetaan Accessin mukana tulevalla Visual Basic for Applications (VBA) ohjelmointikielellä. VBA on looginen vaihtoehto Access-sovelluksen ohjelmointiin, koska se löytyy jokaisesta Access-ohjelmasta käyttövalmiina.



KUVIO 3. Sovelluksen toimintaperiaate

Tietoturvaan halutaan kiinnittää erityistä huomiota, ja siksi sovelluksen käyttäjä tarvitsee työkaluun itselleen käyttöoikeudet. Muiden pääsy sovellukseen estetään.

4.2.1 Määritykset käyttäjänäkökulmasta

ERP-päällikön haastatteluissa tuli useasti esille työkalun tarkoitus helpottaa työtä ja siksi uudesta sovelluksesta halutaan käyttäjälle mahdollisimman helppo ja selkeä. Yritys oli tehnyt jo ennalta pienimuotoista suunnittelua, miettien samalla käyttäjien valmiuksia sovelluksen käyttöön.

Ensimmäisen haastattelun avulla luotiin toimintaperiaate sovelluksen toiminnalle käyttäjänäkökulmasta katsottuna. Sovelluksen toimintaperiaatteena on siis käyttäjän kannalta yksinkertaisuus.

Käyttäjä kirjautuu yhteiselle palvelimelle etätyöpöydän avulla ja avaa sieltä Access-sovelluksen. Sovelluksen käyttöliittymänä toimii Access-lomake, joka aukeaa käyttäjän ruudulle. Lomakkeelta hän pääsee määrittämään nappia painamalla, haluaako käsiteltäväkseen järjestelmästä ostohinnat, myyntihinnat vai molemmat. Data tulee Accessin taulukkoon, josta myyjä pääsee siirtämään tiedot Excel-taulukko-ohjelmaan, jossa hän tottuneesti päivittää hinnastojaan. Päivitettäviä tuotteita voi olla tuhansia, joten hintojen päivitys voi kestää useita päiviä, ennen kuin hinnasto on valmis takaisin järjestelmään siirrettäväksi. (Kontio 2013.)

Myöhemmässä haastattelussa tuli ilmi selkeämmin, kuinka sovelluksen käyttäjä toimii Excelin kanssa. Käyttäjä siirtää tiedot omaan Exceliin ja vaihtaa siellä myyntihintojaan. Ostohintoja käytetään vain apuna määritellessä myyntihintoja. Ostohintoja ei siis tarvita enää tämän jälkeen. Excel-taulukkoon ei tarvita lopulta kuin nimikekoodi, valuutta, myyntihinta ja ehkä joku muu kenttä, eli käyttäjän tehtäväksi jää poistaa taulukosta kaikki muu ylimääräinen.

Kun myyjä on tarvittavat muutokset hinnastoon tehnyt, hän avaa uudelleen Access-sovelluksen ja painiketta painamalla pääsee siirtämään hinnastonsa sovellukseen takaisin. Sovellus tekee käyttäjälle siirtotiedoston valmiiksi päivitetyistä hintatiedoista, jonka käyttäjä pystyy siirtämään takaisin järjestelmään. (Kontio 2013.)

4.2.2 Sovelluksen toiminta

Aiemmin tässä luvussa mainittiin, että tietoturvan takia työkalun käyttö halutaan sallia vain niille työntekijöille, joille tämä hintojenpäivitys tapahtuma kuuluu työnkuvaan. ERP-päällikön haastattelun pohjalta tulee ilmi, että sovelluksen on tultava saataville yhteiselle palvelimelle, joten sovelluksen on tunnistettava käyttäjä käyttäjätunnuksen perusteella. Jotta tunnistaminen onnistuisi, on tehtävä uusi tietokanta, jonne syötetään sovelluksen käyttäjät manuaalisesti. Tähän paras tapa

tietoturvallisuuden kannalta on SQL-tietokanta, jonne myös yhteys sovelluksesta toimii hyvin ODBC:n avulla.

Sovelluksen toimintaan tarvitaan paljon erilaisia toimintoja ja näiden tapahduttava käyttäjältä piilossa. Koska työkalu tulee yhteiselle palvelimelle, se tarkoittaa, että sama sovellusversio on kaikkien käyttäjien saatavilla. Siksi on tärkeää, ettei työkalun toimintataustaan pääse vahingossa käsiksi. Tähän ratkaisuna on Access-asetuksista löytyvä "on-open" asetus, jossa pystytään määrittämään VBA-koodin avulla, että lomake aukeaa suoraan, kun sovellus avataan.
Lomakkeella on käyttäjää helpottavia painikkeita ja tekstiruutuja määrityksiin, joiden avulla käyttäjä hakee järjestelmästä tietoa. Muut, itse sovelluksen toimintaan liittyvät pääsytiet, ovat käyttäjältä piilotettuna. (Kontio 2013.)

Luvussa 2 käytiin läpi yrityksen ERP-järjestelmää ja mainittiin, että järjestelmä on jaettu konsernin yritysten kesken siten, että data jaetaan osiin data area id:n perusteella, eli esimerkiksi saksalaisen myyjän kirjautuessa järjestelmään hän näkee ainoastaan oman yrityksen datan. Tämä on otettava huomioon sovellusta tehtäessä, että järjestelmästä tulee käyttäjälle ainoastaan oman yrityksen hintatiedot muutettavaksi. Tähänkin ratkaisuna on SQL-tietokanta. Käyttäjäoikeuksia SQL-tietokantaan syötettäessä, sinne syötetään myös käyttäjän kotiyrityksen id, jonka sovellus käy lukemassa.

Sovellus rajaa dataa käyttäjän antamilla määrityksillä. Taustalle tarvitaan paljon kyselyitä sekä VBA-ohjelmointia. Yhteys Accessin ja ERP-järjestelmän välillä luodaan ODBC:n avulla. Yhteys mahdollistaa haluttujen tietokantataulujen katselun Accessissa ja niiden kopioimisen, mutta järjestelmässä olevaa tietoa ei pääse tätä kautta muuttamaan. Järjestelmästä valitaan ne taulut tuotavaksi Accessiin, jotka ovat sovelluksen kannalta oleellisia. Tuontivaiheessa dataa ei pystytä suodattamaan, joka tarkoittaa, että tässä vaiheessa linkitetyissä tauluissa on nähtävissä kaikkien yhtiöiden data. Tietokantataulut pystytään suodattamaan vasta seuraavassa vaiheessa luomalla kysely, joka rajaa datan käyttäjän kotiyhtiön id:n avulla.

ODBC on kätevä kun tietoa halutaan Accessiin, mutta sen avulla ei pystytä siirtämään sitä takaisin järjestelmään. Järjestelmä vaatii tiedon siirtoon erillisen tiedoston, jonka suunnitellaan olevan Excel-tiedosto. Ennen datan siirtämistä takai-

sin järjestelmään on oltava varma, että sinne menevä tieto on oikeanlaista, eikä muutoksia ole tehty vääriin kohtiin vahingossa, kuten esimerkiksi nimikkeen tunnisteeseen. Tämä tapahtuu vertaamalla käyttäjän päivittämää dataa järjestelmässä olevaan dataan. Jos esimerkiksi kaikkia nimiketunnisteita ei löydy järjestelmästä, sovelluksen on annettava siitä käyttäjälle virheilmoitus ja sen käyttäjän korjattava virheet.

Viimeisenä tehtävänä käyttäjällä on kirjautua ERP-järjestelmään ja tuoda data järjestelmään. Järjestelmään luodaan sitä varten käyttöä helpottava painike, jota avulla importtaus tapahtuu. Tämän jälkeen tuotehinnasto on päivitettynä järjestelmässä.

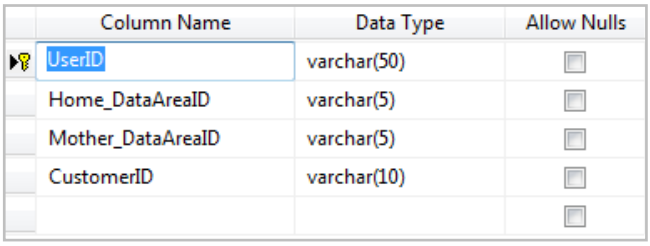
4.3 Käyttäjäoikeudet

Tietoturvan merkitys on tullut aiemmissa kappaleissa monesti esiin ja on muistiinpanoissa ensimmäinen tarkastelun kohde. Oikeuksien määrittämiseen ei tässä sovelluksessa tarvitse käyttöön paljoakaan Accessin ominaisuuksia, koska paremman tietoturvallisuuden vuoksi käytetään SQL-tietokantaa ja Windows-autentisointia. Tämän takia riittää, että Accessin avulla otetaan yhteys SQL-proseduuriin ja tarkastetaan löytyykö kyseiselle käyttäjälle oikeus sovelluksen käyttöön.

Ensin täytyy luoda SQL:ään tarvittavat tiedot, että Accessissa onnistutaan ottamaan käyttöön oikeuksiin viittaavat toiminnot, joka tapahtuu luomalla Microsoft SQL Management-ohjelmistolla uusi tietokanta. Tietokantaan tarvitsee tehdä vain yksi uusi taulu, joka sisältää neljä kenttää.

SQL-tietokantataulun tarkoituksena on tunnistaa työkalun käyttäjä luotettavasti ja turvallisesti. Työkalun toiminnalle on tarpeen, että tauluun syötetään myös muita kenttiä, mistä ilmenee käyttäjän kotiyrityksen maatunnus sekä sen maan tunnus, josta käyttäjän yritys ostaa. Näiden kenttien avulla Access pystyy lajittelemaan ERP-järjestelmästä haetut tiedot siten, että käyttäjälle tulee näkyviin tietoja vain näistä kahdesta maista.

Lisäksi tarvitaan kenttä, mihin syötetään käyttäjän kotiyhtiön asiakasnumero, jonka avulla pystytään hakemaan myyntihinnat yrityksestä, mistä käyttäjän yritys tuotteita ostaa. Seuraavassa kuvassa on nähtävillä tietokantataulun kentät, joihin tulee tiedot jokaisesta uuden työkalun käyttäjästä.



Column Name	Data Type	Allow Nulls
UserID	varchar(50)	
Home_DataAreaID	varchar(5)	
Mother_DataAreaID	varchar(5)	
CustomerID	varchar(10)	

KUVA 1. SQL-tietokantataulun kenttien määrittäminen

Työkalua varten luodaan SQL-proseduuri, jotta tauluun saadaan turvallinen yhteys sovelluksesta. Proseduuriin syötetään kysely, joka etsii kyseisen käyttäjän tunnuksen perusteella tiedot luodusta SQL-taulusta.

Koska oikeuksien määritys tehtiin SQL-palvelimelle, siihen täytyy saada yhteys Accessilla, jotta sovellus saa tietoonsa nämä oikeudet. Tässä vaiheessa tehdään Accessiin uusi tietokanta uutta työkalua varten ja sen jälkeen luodaan yhteys SQL-palvelimelle, jossa käyttäjien tiedot tulevat olemaan ja mistä työkalu saa ne käyttöönsä.

Yhteys muodostetaan ODBC-rajapinnan avulla ja se luodaan joka kerran, kun joku avaa sovelluksen. Samalla kertaa luodaan yhteys sekä SQL-tietokantaan, että järjestelmään. Määritys lisätään käyttöliittymälomakkeen taakse koodiin ja virheen käsittelijän avulla taataan käyttäjälle informointi tilanteesta, jos ODBC-yhteyttä ei jostain syystä pystytäkään luomaan. VBA:ssa virheen käsittelijä on käytännössä koodin pätkä, johon lisätään toiminto, jos Access antaa virheilmoituksen eikä pysty jatkamaan. Tässä tilanteessa se on sanomaikkuna käyttäjälle, joka kertoo, että yhteyttä ei pystytty luomaan ja pyytää ottamaan yhteyttä IT-vastaavaan.

Jotta yhteys järjestelmään ja SQL-kantaan saadaan, tarvitsee luoda yleismoduuli, jonne pysytään viittaamaan mistä vain sovelluksen eri kohdista. Tänne määrite-

tään pääsyt eri tietokantoihin salasanoineen sekä funktioaliohjelma yhteyden luomiselle. Lisäksi siellä on oltava aliohjelmia, josta yksi rekisteröi ODBC-yhteyden ja toinen ajaa kaikki lomakkeiden ja raporttien Pass-through -kyselyt. Aliohjelmaa tarvitaan myös, kun tietoa aletaan hakea ERP-järjestelmästä.

Tähän moduuliin laitetaan siis VBA-koodilla määritys, että joka kerta kun sovellus avataan, se ottaa ensimmäisenä yhteyden SQL-palvelimelle ja siellä olevaan proseduuriin, jonka avulla katsotaan, onko tällä käyttäjällä oikeuksia. Jos käyttäjän Windows-käyttäjätunnus löytyy tietokantataulusta, sovellus luo automaattisesti kyselyn, jolla se etsii kyseistä käyttäjää tietokannasta sen käyttäjätunnuksen perusteella. Jos käyttäjää ei löydy, sovellus ei avaudu ja käyttäjälle tulee ilmoitus, että hänellä ei ole oikeuksia.

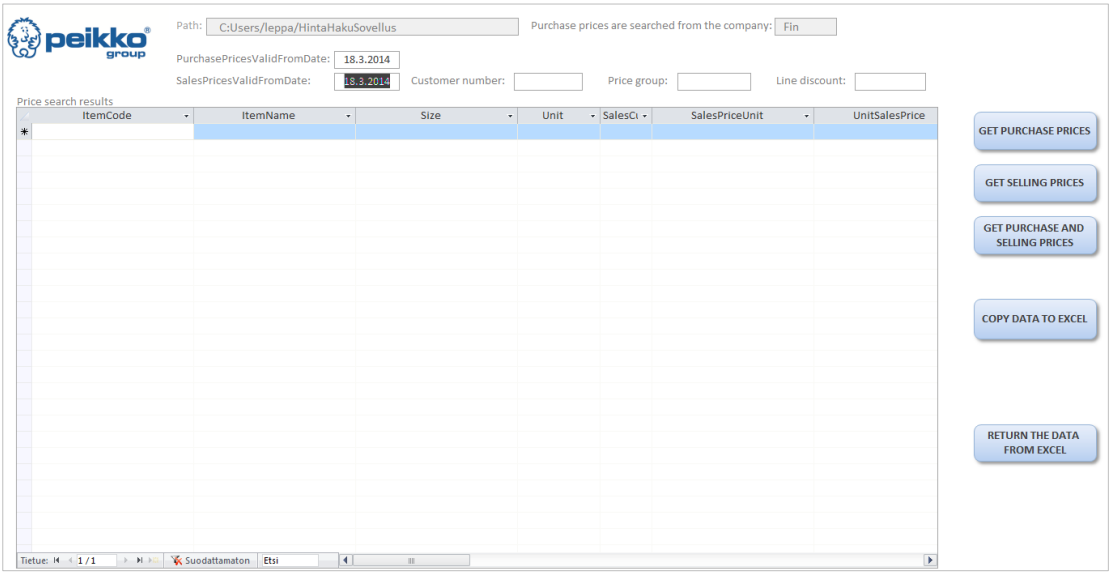
Työkalu tarvitsee käyttöönsä nämä SQL-tietokannasta noudetut käyttäjätiedot, jotta ERP-järjestelmästä haettuja tuotetietoja pystytään lajittelemaan näiden perusteella. Ratkaisuksi tähän lomakkeeseen luodaan jokaiselle tiedolle oma tekstikenttä, joihin tiedot noudettaessa laitetaan ja joista sovellus pystyy lukemaan ja käyttämään niitä toiminnoissaan. Kaikkia tietoja ei tarvitse tuoda käyttäjälle näkyviin, joten kentät piilotetaan tekstikenttien asetusten avulla.

4.4 Access-lomakkeen käyttäminen sovelluksen käyttöliittymänä

Jotta sovelluksen toimintaa päästään ohjelmoimaan, on aluksi tehtävä Accessiin lomake, jonka avulla luodaan käyttäjälle sovellukseen käyttöliittymä. Lomake laitetaan avautumaan koko näytölle heti, kun käyttäjä avaa Access-sovelluksen. Tämä on toimiva tapa estää, että käyttäjä ei pääse käsiksi muihin toimintoihin sovelluksessa ja lisää samalla käyttäjäystävällisyyttä, kun käyttäjällä on mahdollisimman vähän ylimääräistä näkyvillä.

Samalla lomakkeen tarkoituksena on helpottaa käyttäjää siten, että painikkeiden avulla käyttäjä saa helposti järjestelmästä ulos sen, mitä päivitysprosessi vaatii. Painikkeen kautta käyttäjä pystyy helposti myös palauttamaan päivittämänsä datan. Käyttäjän on pysyttävä suoriutumaan hintapäivitysprosessista mahdollisimman helposti, jotta mielenkiinto sovelluksen käyttöön säilyisi ja ERP-järjestelmän hinnasto saataisiin pidettyä eheänä.

Jokaisen vaihtoehtoisen painikkeen taakse syötetään VBA-koodin avulla toiminnot, joiden avulla sovellus nopeasti käsittelee ERP-järjestelmästä tulevat tiedot oikeaan muotoon ja tuo ne lopuksi käyttäjän saataville. Painikkeet nimetään siten, että sovelluksen käyttäjä pystyy helposti hahmottamaan nimien perusteella painikkeesta tapahtuvan toiminnon. Seuraavassa kuvassa on nähtävissä millaiselta työkalu näyttää käyttäjälle, kun hän avaa sovelluksen.



KUVA 2. Access-sovelluksen ensimmäinen käyttöliittymälomake

Käyttöliittymästä on haluttu mahdollisimman selkeä ja helppokäyttöinen. Lomakkeen oikeassa ylänurkassa on edellisessä kappaleessa mainitut, käyttäjältä piilotetut apukentät, joita apuna käyttäen saadaan tehtyä oikeanlainen tulos kyselyllä. Apukentät ovat käytännössä Access-tekstilaatikoita, jotka piilotetaan käyttäjältä laatikoiden asetuksista.

Tummennettuina näkyvät, lukitut tekstikentät ovat luotu käyttäjän avuksi, että hän ymmärtää sovellusta käyttäessään, mistä yhtiöstä ostohinnat haetaan ja hahmottaa tiedostopolun, jonne sovellus lopuksi tekee siirtotiedoston päivitetyistä hinnoista. Tämä on tärkeää, koska käyttäjän tarvitsee löytää ja siirtää tiedostot ERP-järjestelmään viimeisessä vaiheessa.

Lomakkeeseen on laitettu tekstikenttiä, joiden avulla käyttäjä voi määrittää ERP-järjestelmästä tulevaa dataa. Yksi niistä on päivämääräkenttä, johon oletuksena laitetaan nykyinen päivä. Oletuspäivämäärän saa lisäämällä Date() – määrityksen

kyseisen tekstilaatikon oletusarvoksi. Käyttäjä voi syöttää halutessaan muuttaa päivämäärän, jolloin hän saa hinnaston laittamansa päivämäärän mukaan. Tämän lisäksi hän voi suodattaa järjestelmästä tulevaa dataa antamalla hakuehdon, kuten asiakasnumeron (Customer number), hintaryhmän (Price group) ja tarjousryhmän (Line discount), mutta tämä ei ole pakollista. Hakuehtojen määritys on avuksi myös silloin, jos päivitystarve tulee esimerkiksi ainoastaan johonkin hintaryhmään, niin työkalua voidaan käyttää avuksi. Jos hakuehtoja ei syötetä, järjestelmästä tuodaan oletuksena kaikki käyttäjän yhtiön hintatiedot nimikkeistä.

Käyttäjä saa järjestelmästä lomakkeen painiketta painamalla valittua, haluaako hän käyttöönsä tuotteiden myynti- vai ostohinnat, tai molemmat. Käyttäjällä ei ole oikeutta muuttaa muita hintoja ERP- järjestelmässä kuin kotiyrityksensä myyntihintoja, mutta usein myyntihintoja määritettäessä halutaan käyttää apuna myös tuotteiden ostohintoja. Haun tulos tulee alilomakkeelle, joka on liitetty tämän käyttöliittymälomakkeen päälle, jolloin se näyttää yhtenäiseltä lomakkeelta. Alilomake nollataan jokaisen haun aluksi, jolloin hakutulokset tulevat aina tyhjään pohjaan eikä mene sekaisin, vaikka käyttäjä painelisi jokaista painiketta vuorotellen. Lisäksi tulokset näyttävä lomake on lukittu, jonka avulla data pidetään muuttumattomana, eli vältetään käyttäjän vahingot hänen selaillessa tuotua dataa.

Jokainen painike vaatii toimiakseen taakseen paljon Access-kyselyjä, joiden avulla lajittelu ERP-järjestelmän tiedoista tapahtuu. Kun käyttäjä painaa painiketta, työkalu suorittaa kyselyt nopeasti ja käyttäjältä huomaamatta. Apuna kyselyissä käytetään sekä SQL:ssä määritettyjä käyttäjän tietoja, että käyttäjän itse lomakkeelle syöttämiä ehtoja.

4.5 Datan siirto ja palautus Excelistä

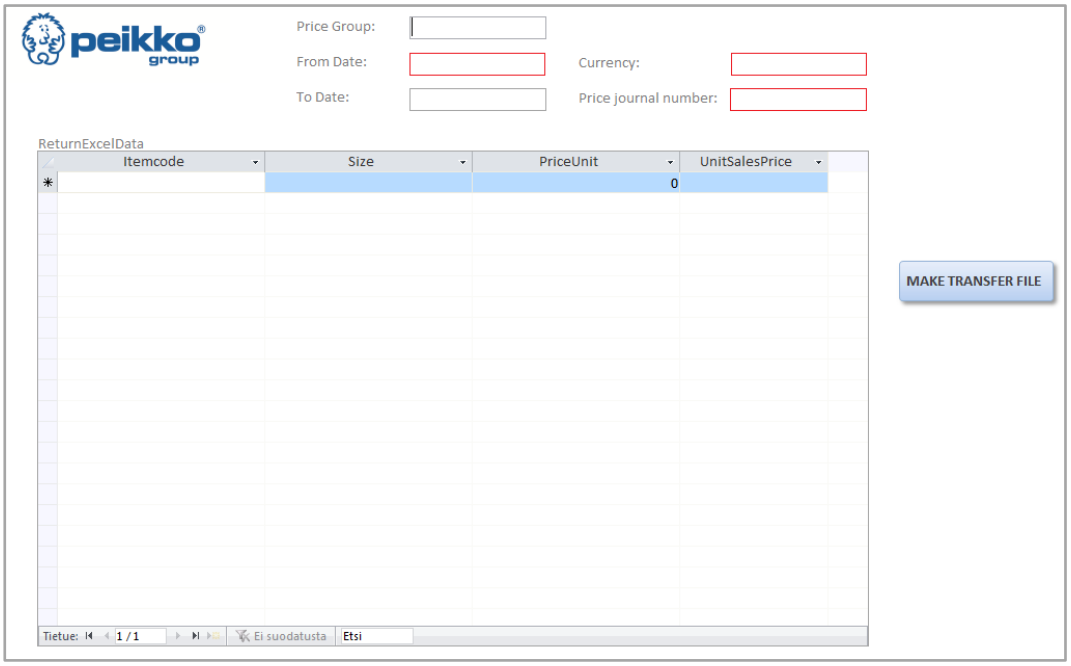
> *Mietintää aiheuttaa, kuinka saada data Exceliin ensin käyttäjän muokattavaksi, siitä muokattuna takaisin Accessiin ja Accessin kautta AX:aan. (Muistiinpano 29.8.2013)*

Käyttäjien tottumusten takia yritys piti Exceliä tärkeänä osana hintojen päivitysprosessia ja siitä ei sovellusta tehtäessä haluttu luopua. Asiaan etsittiin ratkaisua siten, että käyttäjäystävällisyys ei kärsisi paljoa.
Access-lomakkeeseen tehdään tiedon siirtämistä varten painike ”Copy data to

Excel”, jota painamalla sovellus ajaa kyselyn, joka siivoaa järjestelmästä tulleen datan poistaen turhat sarakkeet ja jäljelle jää vain hintojen päivitysprosessin kannalta oleellisemmat. Näiden tietojen näyttämiseen avataan uusi taulukko, jossa tiedot ovat ja näin käyttäjän on helpompi kopioida tieto Exceliin ja käsitellä sitä siellä. Käyttäjälle jää Excelissä tehtäväksi päivittää myyntihinnat ja jättää jäljelle sarakkeet, jotka tarvitaan kun tieto siirretään takaisin järjestelmään.

Päivitetyn hinnaston siirtäminen takaisin sovellukseen vaatii lisätoimintoja, että tiedot saadaan täydelliseksi ja tarkistukset tehtyä. Avuksi tarvitaan toinen lomake, joka aukeaa siinä vaiheessa, kun käyttäjä haluaa siirtää hinnaston takaisin järjestelmään. Lomakkeella on kaksi päätehtävää. Ensimmäinen tehtävä liittyy käyttäjän päivittämän datan tuomiseen takaisin sovellukseen.

Kun käyttäjä on tarvittavan päivityksen tehnyt, hän avaa taas työkalun ja painaa ”Return data from Excel” – painiketta ja tällöin uusi käyttöliittymä aukeaa. Käyttäjä käy kopioimassa tiedot Excelistä ja liittää ne tähän Paste-Append – kuvakkeen avulla, jonka avulla saadaan data liitettyä oikeisiin kenttiin. Paste-Append -kuvake saadaan näkyviin Accessin yleisistä asetuksista, oletuksena se on yleensä piilossa.



KUVA 3. Toisen vaiheen käyttöliittymä sovelluksessa

Lomakkeen toisena tehtävänä on siirtää päivitetty hinnasto takaisin ERP-järjestelmään siinä muodossa kuin järjestelmä vaatii. Hintatietoja siirrettäessä pakollisia kenttiä ovat päivämäärä (From Date), mistä alkaen hinnat ovat voimassa sekä valuutta (Currency). Niin kuin aiemmin kappaleessa 4.1 mainittiin, järjestelmä vaatii tietojen päivittämiseen oman Journal-kansion, jolla on yksilöivä numero. Käyttäjän on luotava tämä kansio järjestelmään ja syötettävä se tämän lomakkeen kolmanteen pakolliseen kenttään nimeltä ”Price journal number”. Journal-kansion luominen on siis pakollinen järjestelmän kannalta ja tätä kohtaa ei pysty uuden sovelluksenkaan avulla kiertämään.

Pakollisten tekstikenttien väri on muutettu punaiseksi, että ne kiinnittäisivät käyttäjän huomion ennen kuin hän painaa painiketta siirtotiedoston tekoon.
Lisäksi hän voi halutessaan syöttää tuotehinnoille kaksi muuta tietoa niitä varten oleviin kenttiin ja sovellus osaa lisätä tiedot siirtotiedostoon kaikille tuotteille.

Viimeinen vaihe on siirtää päivitetyt hinnat takaisin ERP-järjestelmään. Käyttölomakkeessa siihen on ohjelmoitu painike ”Make transfer file”, jota painamalla siirtotiedosto tehdään. Painikkeiden taakse kätkeytyy erilaisia toimintoja, joita lähemmin tarkastellaan kappaleessa 4.5.

4.6 Työkalun VB-ohjelmointi ja ehtolauseet

Käyttövirheet ovat inhimillisiä, mutta pienikin virhe voi sotkea koko sovelluksen toiminnon aiheuttaen käyttäjälle epämiellyttäviä kokemuksia, ja yritelmä kehittää järjestelmää voisi epäonnistua. Sen takia on tärkeää yrittää minimoida virheet. VBA:sta löytyy ohjelmallisia keinoja siihen sekä moneen muuhun tarvittavaan toimintoon, joista tässä ERP-työkalussa käytetyimpiä keinoja ovat ehtolauseet.

Joskus virhe voi aiheutua ohjelmallisestakin ominaisuudesta, jota ei ole ymmärretty sovellusta rakentaessa huomioida. Yksi tällainen on null-arvon ja välilyönnin eron huomioiminen, joka täytyy ottaa huomioon työkalun koodissa. Jos kentissä on null-arvoja, haku ei välttämättä onnistu tai tuloksesta jää pois kenttiä. Tämäkin virhe estetään ehtolauseella, joka syötetään jokaisen hakumäärityskentän taakse. Ehdossa määritetään, että jos hakuarvona on ”null” vaihdetaan tilalle tyhjä merkki ” ”.

Järjestelmän ollessa tarkka tiedoista mitä sinne syötetään, se ei osaa yhdistää sanoja keskenään, jos määrityksen eteen on käyttäjältä vahingossa tullut tyhjä välilyönti. VBA:sta löytyy komento LTRIM, jonka avulla työkalu poistaa turhan tilan sanan edestä. Komentoa käytetään jokaisessa kentässä, mihin käyttäjä pystyy syöttämään tietoa.

VBA:ssa käytettävät if-, eli ehtolauseet toistuvat monessa eri sovelluksen rakentamisen vaiheessa ja niiden avulla on asetettu joko sovelluksen toiminnoille tai käyttäjän valinnoille ehtoja, joiden avulla virheet poistetaan. Ensimmäinen ehtolause löytyy heti käyttäjän avatessa lomakkeen, jolloin tarkastetaan onko yhteys olemassa, jos ei ole, sovellus suljetaan.

Niin kuin aiemmin asiaan jo viitattiin, ehtolauseella tarkistetaan monesti, onko käyttäjä muistanut syöttää vaaditut tiedot hakuehtoihin, mitä hänen täytyy muistaa syöttää. Ensimmäisessä lomakkeessa tarkistetaan, onko käyttäjä antanut hakuehtoja. Jos hän on antanut, haetaan ehdon perusteella, mutta jos hakuehtoja on annettu liikaa tai ei ollenkaan, sovellus informoi käyttäjää:

```
If Me("customerID2") <= " " And Me("PriceGroup2") <= " " And Me("LineDisc2") <= " " Then
    MsgBox "You did not give any customer or price group - the prices are the general prices valid for 'All'"""
End If

If Me("customerID2") > " " And Me("PriceGroup2") > " " Or Me("customerID2") > " " And Me("LineDisc2") > " " Then
    MsgBox "You gave too many search value - prices will be searched on the basis of 'Customer number' "
End If
```

KUVA 4. Hakuehtojen tarkistus VBA:lla

Yksi tapa on käyttää ehtolausetta liitettynä Recordsettiin. Työkalussa sitä käytetään, kun tutkitaan toisella lomakkeella, onko käyttäjän syöttämä Journal-numero olemassa. Jotta tiedoston siirto järjestelmään onnistuisi, siellä on oltava olemassa käyttäjän luoma Journal-kansio. Jos tätä Journal-kansion numeroa ei ole, järjestelmä ei suostu ottamaan vastaan siirtotiedostoa. Recordsetin BOF-ominaisuuden avulla verrataan järjestelmästä löytyviä ja käyttäjän antamaa Journal-numeroa keskenään, jos käyttäjän antamalle numeroa ei löydy vastinetta, pyydetään käyttäjää luomaan Journal ensin.

Recordsettiä hyödynnetään samassa lomakkeessa myös muiden tekstikenttien osalta, mitä käyttäjä syöttää. Tarkistuksessa, löytyykö ERP-järjestelmästä se hin-

taryhmä (PriceGroup), minkä käyttäjä on voinut halutessaan syöttää, käytetään apuna SQL-tietokannasta löytyviä käyttäjätietoja, Recordsettiä sekä SQL-lausetta. Käyttäjän asettaman hintaryhmän on oltava perustettuna järjestelmän tietokanta-tauluun, muuten tietojen vienti ei onnistu. Jos hintaryhmää ei löydy, käyttäjälle annetaan viesti, jossa pyydetään luomaan hintaryhmä järjestelmään ensin.

Samaan yhteyteen on laitettu tarkistukset, että käyttäjä on varmasti laittanut tiedot pakollisiin kenttiin. Jos käyttäjä on unohtanut esimerkiksi määrittää valuutan, sovellus antaa käyttäjälle ilmoituksen, jossa se pyytää syöttämään valuutan. Jokainen kenttä on määritetty samalla tavoin ilmoituksen kera, että käyttäjän on helppo paikantaa ja korjata virheensä.

Sovelluksen viimeisimpiä vaiheita on tarkistaa päivitetty hinnasto siltä varalta, että nimikekoodeihin on tullut vahingossa virheitä, eli niitä on muutettu tahtomattaan siten, että järjestelmä ei tunnista niitä enää. Jos tätä tarkistusta ei tehtäisi, virhe huomattaisiin vasta siinä vaiheessa, kun käyttäjä on kirjautuneena järjestelmässä ja siirtämässä päivitettyä hinnastoa takaisin, mutta tässä vaiheessa virhettä ei pystyttäisi kovin helposti enää paikantamaan.

Kun käyttäjä painaa siirtotiedoston teko painiketta, sovellus lähtee tarkistamaan tietoja. Se hakee nimikkeet uudelleen ERP-järjestelmästä käyttäjän kotiyhtiötunnuksen perusteella, ja vertaa sen jälkeen juuri haettuja ja käyttäjän päivittämiä nimikekoodeja keskenään. Toiminnon luomiseen tarvitaan tehdä kyselyjen avulla vastaavuustaulu, jossa vertaaminen tapahtuu käyttäjältä huomaamatta Recordsetin avulla.

```
'Tukitaan löytyykö virheellisiä item-koodeja, jos löytyy näytetään ne:
'*********************************************************************

Dim MyDB2 As Database
Dim MySetti2 As Recordset

    Set MySetti2 = CurrentDb.OpenRecordset("SELECT ITEMID FROM Vastaavuus_ei;")

    If MySetti2.BOF = False Then
        MsgBox "There is found incorrect item codes from data. Press OK and you "
        will see what they are. Please fix them and try to make the transfer file again."
        DoCmd.OpenTable ("Vastaavuus_ei") 'Näytetään virheelliset koodit
    Exit Sub
    Else

    End If
```

KUVA 5. Vertaaminen Recordsetin avulla

Jos löytyy nimikekoodeja, joille ei ole vastaavuutta järjestelmässä, sovellus aukaisee käyttäjälle ensin tiedotteen ja sen jälkeen taulun, jossa näkyvät virheelliset nimikekoodit. Tällä haluttaan helpottaa käyttäjän työtä, koska muuten virheiden löytäminen tuhansien nimikkeiden joukosta voisi olla vaikeaa. Näin käyttäjän on helpompi korjata virheet. Jos virheitä ei ole, taulua ei avata ja siirtotiedosto tehdään.

4.7 Työkalun toimivuuden testaaminen

Työkalua tehdessä on otettava huomioon moni asia, koska toimintoja sen takana on paljon ja yksikin virhe estää sovelluksen oikeanlaisen toiminnan. Tämän takia jatkuva sovelluksen testaaminen rakentamisen aikana on tarpeen. Se tulee esille myös rakentamisen aikana kerätystä tutkimusdatasta toistuvasti. On viisainta edetä hyvin pienissä osissa, ja jokaisen osan rakentamisen jälkeen työkalun toimivuus testataan. Jos sovellus lakkaa toimimasta, virhe on helpompi paikantaa. Tärkeimmäksi testimenetelmäksi tässä tutkimuksessa nousi esiintyminen käyttäjänä. Testikäyttäjän avulla pystyi testaamaan koko rakentamisen aikana sovelluksen toimivuutta.

Testikäyttäjän avulla kokeillaan heti aluksi toimiiko sovellus, jos käyttäjää ei löydykään tietokannasta. Testikäyttäjän tietoja muutetaan ja huomataan, että sovellus toimii niin kuin pitääkin, eli se ei aukea.

Sovelluksen toiminnan takaamiseksi erilaisia tietoa lajittelevia kyselyjä tarvitsee tehdä lukuisia. Kyselyjen järjestelmällinen testaaminen on ollut erityisen tärkeää, koska yhden kyselyn toimimattomuus riittää tuhoamaan koko työkalun toimivuuden. Työkalun toimivuuteen vaadittavia kyselyitä ajetaan manuaalisesti läpi vaihe vaiheelta ja poistetaan virheet. Näissä on oltava erityisen tarkka, koska yhden taulun tai kyselyn muuttaminen vaikuttaa usein moneen muuhunkin paikkaan.

Sovellusta tehdessä oppii koko ajan ja niin myös muistiinpanoista on huomattavissa, että monesti asiat tulivat esille juuri testaustilanteessa.

> *Huomasin sovellusta sulkiessa ja avatessa, että edelliskerralla hakemani hinnat jäävät alilomakkeelle, vaikka ohjelman sulkee. (Muistiinpano 15.8.2013)*

Tämä ongelma olisi johtanut huomaamattomana siihen, että sovelluksen seuraava käyttäjä olisi nähnyt edellisen käyttäjän järjestelmästä hakemat tiedot. Ratkaisuna tähän on lomakkeen ”on-close” asetus, jonka avulla Access luo automaattisesti uuden osion VBA-koodiin, johon syötetään komennot, mitä sovelluksen halutaan tekevän kun käyttäjä sulkee sovelluksen. Tässä tapauksessa se oli nollaus, jonka toteutus kokonaisuudessaan näkyy seuraavasta kuvasta.

```
Private Sub Form_Close()

DoCmd.SetWarnings False

DoCmd.OpenQuery ("GetData_00")
DoCmd.OpenQuery ("GetData_Items00")
DoCmd.OpenQuery ("GetData_Final00")
DoCmd.OpenQuery ("GetData_Items20")

DoCmd.SetWarnings True

DoCmd.Quit
End Sub
```

KUVA 5. Käyttöliittymälomakkeen tyhjennys suljettaessa

Nollaukseen tarvitaan käyttöön poistokyselyt, joiden avulla tyhjennetään halutut taulut ja kuvassa olevassa koodissa ajetaan nämä kyselyt.

Koko sovelluksen tekemisen ajan on mielessä pidetty käyttäjäystävällisyys, virheiden minimoiminen ja sovelluksen toiminnan muuttaminen niiden pohjalta eheämmäksi. Ensimmäisen käyttöliittymän taulukkoa jäsenneltiin siten, että sen katsottiin olevan käyttäjälle mahdollisimman selkeä, mutta hän saa siihen järjestelmästä juuri tarvittavat tiedot päivitysprosessia varten.

Siirtotiedoston tekeminen dat-päätteiseksi aiheuttaa erinäisiä ratkaisuja koodiin, kun huomataan, että siirtotiedoston nimeäminen suoraan dat-tiedostoksi ei onnistukaan. Tiedostoa varten joudutaan ensin luomaan oma spesifikaatio manuaalisesti, jossa ovat tiedostoon tulevat kentät ja järjestelmän vaatimien kenttäerottimien määrittely. Jos kenttien paikkaa tai määrää muuttaa ja ei muista luoda spesifikaatiota tämän jälkeen uudelleen, sovellus antaa error-sanoman eikä toimi.
Tämän lisäksi sovellus tarvitsee paljon lisäohjelmointia, koska tiedostoa ei pysty suoraan nimeämään dat-tiedostoksi. Sovelluksen täytyy ensin tehdä siirtotiedostosta txt-tiedosto, jonka jälkeen se nimetään dat-loppuiseksi, mutta tätäkin testa-

tessa törmätään ongelmaan. Uudelleen nimeäminen onnistuu, jos samannimistä tiedostoa ole olemassa, mutta koska sovellus tekee aina samannimisen tiedoston, se olisi ongelma. Ohjelmallisesti täytyy siis tuhota ensin mahdollinen samanniminen tiedosto ja vasta tämän jälkeen nimetä se uusiksi. Siirtotiedosto tehdään aina sovelluksen juureen, että käyttäjä löytäisi sen helposti.

Viimeisessä vaiheessa, kun sovellus alkaa olla valmis, sitä aletaan testata kokonaisuudessaan siirtämällä tietoa siten, kun tuleva käyttäjä tulee käyttämään sovellusta.

```
'Tehdään TXT-tiedosto nimeltä PriceFile:
'**************************************
DoCmd.TransferText acExportDelim, "MySpecification", "TheLastTable", _
CurrentProject.Path & "\PriceFile.txt", False

'Tuhotaan ensin mahdollinen samanniminen tiedosto (muuten tulee erroria)
'*********************************************************************
    Dim aFile As String
    aFile = CurrentProject.Path & "\PriceFile.dat"
    If Len(Dir$(aFile)) > 0 Then
    Kill aFile
    End If

'Nimetään tiedosto uudelleen .dat päätteiseksi:
'**********************************************
Name CurrentProject.Path & "\PriceFile.txt" As CurrentProject.Path & "\PriceFile.dat"
```

KUVA 5. Siirtotiedoston tekeminen

Loppuvaiheessa testaaminen on sovelluksen kokonaisuuden kannalta tärkeä vaihe. Sovellusta ajetaan läpi yhä uudelleen sekä pienellä, että isolla tietomäärällä ja etsitään virheitä. Kun sovellus todetaan toimivaksi, sen käyttöä kokeillaan vielä eri tietokoneelta ja eri käyttäjällä, mutta ongelmia ei tule.
Viimeisenä työkalu kompaktoidaan, joka poistaa turhat tyhjät tilat ja saa tietokannan pienemmäksi.

Kun työkalu on saatu tehtyä ja testattua, laaditaan uuden työkalun käyttämiseen selkeät, kirjalliset ohjeet (LIITE 2). Ohjeistus toteutetaan englannin kielellä, koska sovelluksen käyttäjät ovat eri maiden myyjiä. Ohjeet ovat tukena työkalun käyttöön opettelussa ja antavat käyttäjälle selkeän kuvan työkalun toiminnasta.

5 TULOSTEN ANALYSOINTI

Työkalun rakentaminen määritysvaiheesta lopulliseksi tuotokseksi tuotti paljon tietoa kyseisestä prosessista. Tässä luvussa keskitytään nyt miettimään tutkimuksen kannalta oleellisia seikkoja, kuten tämän artefaktin hyödyllisyyttä. Hyödyllisyyttä pystytään parhaiten esittämään vertaamalla hinnaston päivitysprosessia ennen ja jälkeen. Toiminnanohjausjärjestelmän omaa päivitystyökalua käsiteltiin luvussa 4.1 ja siksi tässä luvussa ensin käsitellään samaa prosessia uuden työkalun avulla, jotta vertaamista päästään tekemään. Luvun tarkoituksena on kiteyttää koko tutkimus siten, että sen avulla saadaan mahdollisimman todenmukaiset vastaukset tutkimuskysymykseen sekä -ongelmiin.

5.1 Hintojen päivitys uuden työkalun avulla

Kun hintojen päivitystä aletaan tehdä uuden työkalun avulla, ensimmäinen vaihe on Journal-kansion tekeminen ERP-järjestelmään. Sen jälkeen käyttäjä avaa sovelluksen, jossa hänellä on läpikäytävänä kaksi lomaketta. Ensimmäinen lomake on hinnaston hakemiseksi järjestelmästä ja sen suodattamiseksi oikeaan muotoon Exceliä varten ja toisen lomakkeen avulla palautetaan tieto Excelistä ja tehdään siitä siirtotiedosto. Käyttäjä voi itse suodattaa järjestelmästä tuotavaa dataa neljän hakumäärityksen avulla, mutta se ei ole pakollista. Käyttäjän ei minimissään tarvitse tehdä muuta kuin painaa kahta painiketta, ensimmäistä hintojen hakemista varten ja toista tietojen suodatusta Exceliä varten. Kun käyttäjä on kopioinut ja vienyt tiedot omaan Excel-tiedostoon, hän voi sulkea työkalun.

Kun käyttäjä on valmis siirtämään hinnat takaisin järjestelmään, hän avaa taas saman sovelluksen ja painaa palautuspainiketta ensimmäiseltä lomakkeelta. Toisella lomakkeella käyttäjän on laitettava kolme pakollista tietoa ja tämän jälkeen hän voi tehdä siirtotiedoston napin painalluksella. Jos päivitettävässä hinnastossa on nimikekoodi väärin, sovellus ilmoittaa käyttäjälle vialliset koodit, jolloin hän korjaa ne Excelissä ja palauttaa tiedoston uudelleen. Viimeiseksi hänen tarvitsee järjestelmästä käsin hakea tämä tiedosto, minkä jälkeen prosessi on ohi.

5.2 Keskeisimmät asiat työkalun tekemisessä

Työkalun tekemisen aikana yritettiin ottaa huomioon mahdollisimman hyvin erilaiset virhetilanteet, mitä käytännössä voi tulla. Koko työkalun rakentaminen perustui helppokäyttöisyyteen, mutta samalla se aiheutti hyvin paljon erillistä suunnittelua, koska oli vaikeuksia saada näinkin laaja toiminnallisuus ja helppokäyttöisyys samaan pakettiin.

Visual Basic-ohjelmointikieli osoittautui suhteellisen käteväksi käyttää tällaisen työkalun tekemiseen. Vaikka VBA:ssa on erilaisia ominaisuuksia hyvin paljon, tällaisen työkalun tekemiseen siitä vaadittiin käyttöön vain murto-osa. Hyvin hyödyllinen siinä oli olio-ohjelmointi, eli käyttämällä Recordsettiä pystyttiin suorittamaan aika monipuolisesti eri tapahtumia, mutta eritoten tietojen hakemisessa työkalun ulkopuolelta se oli korvaamaton.

Recordsetin lisäksi ODBC ja Accessin läpivientikyselyt tarjosivat keinon suoran yhteyden luomiseksi SQL-palvelimelle ja ERP-järjestelmään. ERP-järjestelmästä saatiin halutut tietokantataulut tuotua kokonaisuudessaan Accessiin ja SQL-palvelimelle tehtiin proseduuri, jonka avulla tuotiin käyttäjätiedot työkalun käyttöön.

Accessin muista ominaisuuksista eniten tarvittiin käyttöön erilaisia kyselyjä ja niiden tekeminen oli suurin työ koko aikana. Kyselyitä tuli sovelluksen taakse yhteensä 84 kappaletta, joilla jokaisella on sovelluksessa oma tehtävänsä ja joiden pitää toimia virheettömästi Visual Basicin ajaessa niitä. Kyselyiden avulla pystytään luotettavasti suodattamaan tietoa ja niihin pystytään helposti linkittämään ehtoja lomakkeen kentistä. Käyttäjälle apuna on käytetty runsaasti sanomalaatikoita, jotka helpottavat ja ohjaavat käyttäjän toimintaa eteenpäin.

Kriittisin kohta on, että käyttäjä muistaa noudattaa ohjetta ja jättää Exceliin juuri ne kentät, jotka tarvitaan. Lisäksi nimikekoodien pysyminen muuttumattomana on vaikea taata, koska tätä työkalua ohjelmoimalla ei voida vaikuttaa siihen muuten, kuin tarkistamalla tietojen oikeellisuus.

Seuraavan sivun taulukko on yhteenveto työkaluun käytetyistä Accessin ja VBA:n toiminnoista ja sen avulla saadaan vastaus tutkimusongelmaan: Mitä ominaisuuk-

sia ohjelmasta tarvitaan käyttöön, että saadaan tehdyksi tavoitteen mukainen ERP-järjestelmän lisätyökalu?

ACCESS	VISUAL BASIC

• 2 lomaketta
  ○ 6 painiketta
  ○ 12 tekstikenttää
  ○ 4 piilokenttää
  ○ On Open – asetus
  ○ On Close – asetus
  ○ Oletuspäivämäärä
• 2 alilomaketta
• 27 päivityskyselyä
• 1 läpivientikysely
• 15 poistokyselyä
• 37 liittämiskyselyä
• 4 valintakyselyä
• 14 lokaalia tietokantataulua
• 7 ODBC-tietokantataulua
• 1 perusmoduuli
• 2 luokka- eli lomakemoduulia
• Paste-Append-toiminto
• Vienti tekstitiedostoon
  ○ Spesifikaatio
• Tietokannan kompaktointi
	• SQL-lauseet (aliohjelma)
	• Läpivientikysely (aliohjelma)
	• ODBC:n rekisteröinti (aliohjelma)
	• Yhteyksien tekeminen (funktio)
	• Vanhojen kyselyjen poisto (funktio)
	• Käyttäjätietojen tallennus (aliohjelma)
	
	• Private Sub…End Sub…Exit Sub
	• Call
	• Dim...As…
	• If…Else/And/Or…End If
	• Set
	• MsgBox
	• On Error GoTo
	• Do.Cmd.
	  ○ SetWarnings
	  ○ OpenQuery
	  ○ CopyObject
	  ○ DeleteObject
	  ○ TransferText
	  ○ Close
	  ○ Quit
	• Me.Refresh
	• Kill
	• LTrim
	• Recordset
	  ○ MoveFirst
	  ○ BOF

TAULUKKO 1. Yhteenveto toiminnoista työkalun taustalla

Taulukon avulla nähdään, mitkä eri ominaisuudet on tarvittu käyttöön päivitys-työkalun rakentamiseen. Access sisältää monipuolisesti erilaisia ominaisuuksia,

joiden avulla tällaisen rakentaminen on täysin mahdollista. Erilaiset ominaisuudet tekevät työkalun rakentamisesta monipuolisempaa verrattuna sovellukseen, joka ohjelmoidaan kokonaan komentojen avulla. Accessissa sanomalaatikot ovat tärkeä keino ohjata käyttäjää ja niiden avulla sovelluksen toiminta saadaan vastavuoroiseksi, eli sovellus keskustelee käyttäjän kanssa jatkuvasti.

5.3 ERP- ja Access-työkalujen väliset erot

Jotta pystytään havaitsemaan uuden työkalun mahdolliset hyödyt, on verrattava vanhaa ja uutta työkalua keskenään ja etsiä eroavaisuuksia.
Kun tarkkaillaan myyjien tekemää hintojen päivitysprosessia ERP:n oman työkalun ja uuden Access-työkalun avulla, on ensiksi otettava huomioon prosessin vaatimien eri vaiheiden vaatimat toiminnot ja niiden helppous.

Molemmissa tavoissa prosessin aluksi on käyttäjän luotava ERP-järjestelmään Journal-kansio ja sitä ei uudella työkalulla pystytty kiertämään. Tässä vaiheessa käyttäjä antaa Journal-kansiolle nimen, jonka järjestelmä numeroi automaattisesti.

Ensimmäisen vaiheen lisäksi kovin paljon yhteistä työkalujen käytöllä ei ole. Seuraavaksi käyttäjä haluaa päivitettävät hinnat järjestelmästä. Vanhalla tavalla käyttäjän pitää etsiä tietojen vientiin tarkoitettu työkalu, velho. Velhon avulla ei pystytä suodattamaan järjestelmästä tullutta tietoa minkään hakuarvon avulla, vaan se tulee aina kokonaisuudessaan määrittämästään tietokantataulusta, halusipa sitä tai ei. Käyttäjä ei myöskään saa tuekseen ostohintoja, koska hintatiedot saadaan ulos näin vain käyttäjän kotiyhtiöstä. Excel tiedoston luomiseksi käyttäjän on käytävä läpi 8 eri lomaketta asennusvelhossa, jonka jälkeen järjestelmä luo automaattisesti Excel-tiedoston kaikilla hinnastotiedoilla. Access-työkalussa käyttäjä käy läpi ainoastaan yhden lomakkeen, johon hän voi halutessaan laittaa hakuehtoja. Tiedot tulevat taulukkoon, josta hän vie ne Exceliin itse avaten uuden työkirjan ja nimeten sen haluamakseen.

Näkyvin ero työkalujen välillä tietojen viemisessä Exceliin on siis tiedon saatavuuden, sen suodattamisen ja siihen tarvittavien eri vaiheiden määrän välillä. Access-työkalun avulla saadaan vaiheita vähemmäksi ja selkeämmäksi. Uusi työkalu tuo suoraan käyttäjälle oikean tietokantataulun tietoineen eikä hänen tarvitse itse

määrittää tuotavaa taulua satojen joukosta, niin kuin ennen. Uudella työkalulla käyttäjä saa hintojen määritykseen avukseen myös ostohinnat, kun vanhalla tavalla tämä ei ole mahdollista ollenkaan.

Kun päivitykset on tehty, Excel-tiedosto on siirrettävä takaisin ERP-järjestelmään. Ennen kuin tiedot voidaan siirtää Access-työkalulla, käyttäjän on ohjeiden avulla poistettava kaikki ylimääräiset kentät Excelistä siten, että jäljelle jää yhdeksästä Exceliin siirretystä sarakkeesta ainoastaan 4 saraketta, jotka ovat uuden työkalun kannalta välttämätöntä.

Vanhassa tavassa tämä vaihe jää välistä, koska ERP:n ohjattu toiminto on luonut Excel-tiedoston luotaessa juuri oikeanlaiseksi. Isona miinuksena tässä on se, että Excel-tiedosto on kovin iso, koska järjestelmävelho luo joka kerta tiedoston tuotteiden kaikilla tiedoilla. Käytännössä jokaisesta tuotteesta on kymmeniä sarakkeita tietoa, joista vain murto-osasta on hintojenpäivitysprosessin kannalta hyötyä. Käyttäjä joutuu työskentelemään keskellä tietoviidakkoa ja varomaan, ettei vahingossa muuta mitään oleellista, vaikka hän päivittää prosessissa jokaisesta tuotteesta usein vain hinnan.

Tätä vaihetta pohdittaessa on siis punnittava, onko käyttäjälle helpompaa työskennellä datan parissa, jossa on prosessiin nähden liikaa tietoa, vai datan parissa, jossa on prosessia varten juuri oikeat kentät helpottamassa myyjän työtä, mutta joista hän joutuu osan poistamaan, ennen kuin siirto järjestelmään pystytään tekemään.

Alkuperäisen ERP-työkalun avulla tieto siirretään järjestelmään siten, että käyttäjä kirjautuu järjestelmään, etsii siihen tarvittavan erillisen tuontityökalun ja hakee sen avulla Excel-tiedoston tietokoneeltaan. Tämän jälkeen, toiselta lomakkeelta, saadaan tiedot lähetettyä takaisin tietokantatauluun. Jos päivityksen aikana on tapahtunut virheitä siten, että data on vääristynyt, työkalu ei osaa ilmoittaa virheen paikkaa.

Access-työkalun avulla siirto tapahtuu siten, että käyttäjä avaa työkalun uudelleen, painaa painiketta, jolloin aukeaa toinen lomake. Tähän lomakkeeseen käyttäjä liittää tiedot Excelistä, antaa 3 arvoa kenttiin ja painaa painiketta, jolloin siirtotie-

dosto tehdään. Sen jälkeen käyttäjä kirjautuu järjestelmään ja painaa tietojen tuontinappia, joka on tehty järjestelmään ja tiedot menevät tietokantatauluun. Jos päivityksen aikana on tapahtunut virheitä siten, että vääränlaisia muutoksia on tapahtunut esimerkiksi nimikekoodissa, työkalu ilmoittaa käyttäjälle virheelliset koodit. Lisäksi työkalu neuvoo ja ilmoittaa erilaisista puutteista sanomalaatikoiden avulla koko prosessin ajan.

Viimeisessä vaiheessa hyödyt ovat pienemmät, mutta prosessia on saatu tässäkin selkeytettyä. Tehdyssä työkalussa käyttäjä poistaa ensin turhat kentät ja toiseksi siirtää ne Access-sovellukseen, jossa tapahtuu tietojen määritys kolmeen kenttään sekä siirtotiedoston tekeminen. Kolmanneksi käyttäjä siirtää tämän tiedoston ERP-järjestelmästä käsin takaisin järjestelmään. Kappaleessa 4.1 huomattiin, että ERP:n oman työkalussa käyttäjä joutui käyttämään kahta eri työkalua tämän prosessin aikaansaamiseksi ja työkalut sijaitsevat järjestelmässä eri paikoissa.

Alla olevan taulukon avulla on voidaan havaita uuden työkalun tuomat muutokset päivitysprosessiin.

Tapahtumat	**ERP:n oma työkalu**	**Access-työkalu**
Journal-kansion teko	Kyllä	Kyllä
Virheiden tarkistus	Lopussa	Kyllä, joka vaiheessa
Virheiden paikannus	Ei	Kyllä
Datan vienti Exceliin		
Lomakkeiden määrä	8	1
Oikean taulun etsiminen	Käyttäjä	Automaattinen
Hakuehtojen määritys	Ei	Kyllä, muutamia
Tuotteiden ostohinnat	Ei	Kyllä, valinnainen
Excel-tiedoston luonti	Automaattinen	Käyttäjä
Tietosarakkeiden määrä	28	9
Tarvittavia työkaluja	2	1
Datan tuonti järjestelmään		
Lomakkeiden määrä	3	2
Tarvittavia työkaluja	2	2

TAULUKKO 2. Vanhan ja uuden työkalun vertailu

Suurin kehitys on tapahtunut hinnaston päivitysprosessin siinä vaiheessa, kun hintatietoja lähdetään siirtämään järjestelmästä Exceliin päivitettäväksi. Vaihe on tehty järjestelmän omassa työkalussa liian monivaiheiseksi käyttäjää ajatellen, ja tästä voidaan vetää johtopäätös, että tämä on syy miksi osa myyntiyrityksistä ei ole päivittänyt hinnastojaan ollenkaan järjestelmään. Siksi on tärkeää, että uudessa työkalussa prosessin kehitys on keskittynyt juuri tähän kohtaan ja hyödyt ovat selvästi havaittavissa.

Hinnaston takaisin siirtämiseksi järjestelmään oli vaikeampi saada aikaa kehitystä, mutta sitä tuli jonkin verran. Yksi parannus on lomakkeen selkeyttäminen. Toiseksi hyödytään uuden työkalun tekemästä tarkistuksesta hinnaston takaisin siirron yhteydessä, joka etsii vialliset nimikekoodit ja näyttää ne käyttäjälle. Tietojen tuomiseen tarvittavaan työkalumäärään ei saatu parannusta, koska järjestelmän tietokantataulujen päivitys on tapahduttava järjestelmästä käsin ja tätä ei pystytty kiertämään.

6 YHTEENVETO

Tutkimuksen tarkoituksena oli tarkastella Microsoft Access 2010 hyödyntämistä ERP-järjestelmän aputyökaluna. Tutkimus sai alkunsa, kun tuotantoalan yritys antoi tehtäväksi toteuttaa ERP-työkalun, jolla organisaation myyntiyhtiöt pystyisivät päivittämään tuotehinnastojaan helpommin järjestelmään. Tämän organisaation käytössä oli Microsoft Dynamics AX-toiminnanohjausjärjestelmä. Tutkimusta suoritettiin samalla, kun yrityksen käyttöön rakennettiin uutta sovellusta Access 2010 avulla tarkastellen työkaluun tarvittavia ominaisuuksia sekä työkalun hyödyllisyyttä. Tutkimus perustuu suunnittelutieteeseen ja siksi erityistä huomiota kiinnitettiin tekniseen ympäristöön ja itse työkalun rakentamiseen.

Opinnäytetyö aloitettiin tutustumalla kirjallisuuteen ja muihin tutkimuksiin, millaisia ongelmia toiminnanohjausjärjestelmissä yleensä on. Näiden avulla haluttiin selvittää, ovatko yrityksessä esiin nousseet järjestelmän käyttöongelmat kuinka yleisiä. Tutkimuksista kävi ilmi, että käytettävyysongelmat ovat suurimpia tyytymättömyyden aiheuttajia eri yrityksissä järjestelmän käyttäjien keskuudessa. Nimenomaan järjestelmien omat työkalut tuottavat haastetta, niiden ollessa liian monivaiheisia ja epäloogisia.
Lisäksi etsittiin tietoa siitä, onko tämänkaltaisista ratkaisuista käyttää Accessia ERP-järjestelmän räätälöintiin, aiempia tutkimuksia saatavilla. Tulokseksi saatiin, että aiempia tutkimuksia tästä aiheesta joko ei ole tehty, tai on erittäin huonosti saatavilla.

Tutkimuksen alussa testattiin hinnaston päivitystä järjestelmän oman työkalun avulla ja havaittiin tämän toiminnon vaativan onnistuakseen kolmen eri työkalun käyttöä ja yhteensä 12 lomakkeen läpikäymisen. On siis varsin ymmärrettävää, että prosessi on aiheuttanut joidenkin myyjien keskuudessa innottomuutta prosessia kohtaan, koska koin itsekin sen aika hankalaksi hyvästä ohjeistuksesta ja IT-alan koulutuksestani huolimatta. Uutta työkalua tehdessä haluttiinkin keskittyä käytön helppouteen ja siihen, että käyttäjän tarvitsee tehdä mahdollisimman vähän käyttäessä työkalua. Samalla haluttiin minimoida kaikki mahdolliset virhetilanteen niin käyttäjän, kuin sovelluksen toiminnan kannalta.

ERP-järjestelmää oli tarkoitus kehittää uuden työkalun avulla siten, että päivitysprosessi olisi organisaation myyjille helppoa ja sen avulla saataisiin heille mielenkiintoa pitää ajantasaisia hinnastoja yllä järjestelmässä.
Rakennuksen vaiheista tärkeimmäksi asiaksi nousi testaaminen, jonka avulla löydettiin sovelluksessa ilmenevät virheet ja testikäyttäjän avulla nähtiin sovellus käyttäjän näkökulmasta. Kun työkalu saatiin valmiiksi, se testattiin lukuisia kertoja ja todettiin toimivaksi. Tutkimuksessa ei kiinnitetty huomiota työkalun käyttöönottoon, vaan se perustui työkalun rakentamiseen ja työkalun toimivuuteen.

Kokonaisuutta ajatellen voidaan sanoa, että uuden työkalun avulla saatiin hintapäivitysprosessia käyttäjäystävällisemmäksi ja helpommaksi. Käyttöliittymä on selkeä ja pelkistetty siten, että käyttäjällä ei ole kovinkaan montaa kohtaa, missä voisi epäonnistua. Käyttöliittymäksi tehtiin yksinkertainen Access-lomake, jota myyjä ohjaa painikkeiden ja tekstilaatikoiden avulla.
Suurin kehitys tapahtui prosessin ensimmäisen vaiheen osalta, jossa myyjä siirtää tiedot ERP-järjestelmästä Exceliin päivitettäväksi. ERP:n oman ohjatun toiminnan käyttö päivitysprosessissa osoittautui monivaiheiseksi ja hankalaksi, ja erehtymisen vaara on monessa kohtaa suuri. Siksi oli tärkeintä saada juuri tämä vaihe helpommaksi käyttäjän kannalta ja data-analyysin perusteella siinä onnistuttiin. Uuden työkalun avulla virheiden tekemisen mahdollisuus pieneni ja lisäksi avuksi tulivat jatkuvat ohjelmalliset tarkistukset taustalla, joiden avulla virheistä ilmoitetaan myyjälle yksityiskohtaisesti sellaisen sattuessa. Myyjän työtä helpottamaan saatiin tuotteiden ostohinnat, jota ominaisuutta ei ennen ole voitu käyttää, koska ERP:n oma työkalu ei pysty tähän ja muut, päivitysprosessin kannalta tarpeettomat tiedot tuotteista saatiin poistettua.

Uuden työkalun avulla ei saatu suurta kehitystä aikaiseksi viimeiseen prosessin vaiheeseen, jossa käyttäjä siirtää tiedot takaisin ERP-järjestelmään. Tämä johtuu siitä, että tiedot viedään tehdyssä työkalussa uudelleen Access-työkaluun, joka tekee siirtotiedoston ja jossa myyjä lisää tuotteisiin muutaman pakollisen tiedon. Jos Excel-tiedosto olisi haluttu viedä suoraan järjestelmään, se olisi edellyttänyt siinä olevan 28 saraketta, että järjestelmä olisi ottanut sen vastaan. Tällöin myyjä olisi edelleen joutunut päivitystehtävään turhien tietojen keskellä ja tämä ei olisi tuonut haluttua kehitystä prosessiin. Sen sijaan oli tärkeämpää, että tietojen vienti

järjestelmästä Exceliin käsiteltäväksi helpottui monelta osin tehden uudesta työkalusta käyttäjäystävällisemmän kuin ERP:n oma työkalu.

Tutkimuksen päätavoitteena oli selvittää Microsoft Access 2010 soveltuminen ERP-järjestelmän aputyökaluksi. Työkalua rakennettaessa kiinnitettiin erityistä huomiota Accessin tarjoamiin toimintoihin, joita pystytään hyödyntämään uutta sovellusta tehdessä. Access 2010 osoittautui käytännölliseksi käyttää juuri tähän tarkoitukseen, kun työkalun avulla käsitellään suurta määrää tietoa. Accessin suurin etuus olikin tietokantataulujen käsittelyominaisuudet, joiden avulla tietoa saatiin lajiteltua, järjestettyä ja suodatettua. Ohjelmasta oli helppo luoda yhteys eri palvelimille ODBC:n avulla ja tätä kautta ERP-järjestelmän tietokantataulujen tuominen oli helppoa ja nopeaa. Visual Basic – ohjelmointia tarvittiin ajamaan näitä Access-kyselyjä ja vastaamaan yleisesti sovelluksen toiminnoista ja se taipui siihen erinomaisesti. Työkalun käyttämiseen tarvittavat käyttäjäoikeuksien määritykset katsottiin turvallisemmaksi tehdä SQL-palvelimella, jolla minimoitiin käyttäjätietojen pääsy vääriin käsiin. Niiden istuttaminen Access-taulukkoon ei olisi ollut niin turvallinen vaihtoehto.

6.1 Tutkimuksen hyödyllisyys

Johdannossa mainitut ERP-järjestelmän käytettävyysongelmat ovat tätä päivää ja kuten huomattiin, ei edes vaihtaminen uudempaan ERP-järjestelmäversioon välttämättä korjaa havaittuja epäkohtia. Alussa mainittiin myös, että yritykset tarvitsevat ERP-järjestelmiä liiketoiminnassaan pysyäkseen kilpailukykyisenä ja tehokkaana. Järjestelmistä täytyy pyrkiä saamaan kaikki teho irti ja siksi on tärkeää etsiä ratkaisua niihin järjestelmän käytettävyysongelmiin, jotka katsotaan laskevan järjestelmän tehokkuutta.

Monissa yrityksissä Microsoftin toimisto-ohjelmistopaketti on jokapäiväisessä käytössä, joten Accessin käyttäminen tässä on luonnollinen ja helppo vaihtoehto. Näin ollessa, uuden työkalun tekeminen Accessin avulla on erittäin kustannustehokasta, koska uusia sijoituksia ohjelmistoihin ei tarvitse tehdä. Työkalu on nopea toteuttaa, ja jos sen avulla järjestelmän tehokkuus ja käyttäjien mielenkiinto käyttää järjestelmää nousevat, ovat työkalun hyödyt verrattuna sen tekemisestä aiheu-

tuviin kustannuksiin suuret. Sen perusteella voidaan päätellä tutkimuksessa tehdyn työkalun hyödyttävän pitkällä aikajänteellä sekä yritystä, että järjestelmän käyttäjiä.

Microsoft Dynamics AX on tilastojen mukaan maailman käytetyimpiä ERP-järjestelmiä ja siksi tutkimus tuo esille tärkeää tietoa myös muihin yrityksiin. Tämän kaltainen ERP-työkalu on käytettävissä eri alan erikokoisissa yrityksissä, joissa Microsoftin ERP-järjestelmää käytetään tukemaan yrityksen toimintoja. Tässä tutkimuksessa työkalu on toteutettu suurelle, kansainväliselle organisaatiolle, jolla myös ERP-järjestelmä on hyvin laaja ja lisäksi jaettu osiin maittain. Tämän takia työkalun toimintaan tarvittiin paljon suodatusta, lajittelua ja ohjelmointia tehden työkalun tekemisen haastavammaksi. Työkalun toteuttaminen pienemmälle yritykselle on huomattavasti helpompaa ja yksinkertaisempaa. Silti tekijältä vaaditaan jonkinlaista kokemusta ohjelmoinnista, Accessista sekä perustietämystä tietokannoista ja tietojärjestelmistä. Jos nämä asiat ovat hallinnassa, työkalu on nopea tehdä ja se helpottaa suuresti tietojen päivittämistä ERP-järjestelmään varsinkin silloin, jos päivitystä tarvitsee tehdä usein.

Yritys on alustavasti suunnitellut ajavansa uuden ERP-työkalun organisaation myyntiyhtiöiden käyttöön vuoden 2014 aikana, jolloin tarkoituksena on opastaa ja aktivoida myyntiyhtiöitä tähän.
Jatkotutkimuksen arvoinen olisikin uuden työkalun käyttöönotto, jolla saataisiin lisäarvoa myös tälle tutkimukselle sekä yleisesti. Tutkimuksessa voitaisiin tutkia työkalua käytössä ja saada selville kuinka paljon tällainen lisätyökalu tuo tehokkuutta lisää ERP-järjestelmän käyttöön tai säästetäänkö tällä esimerkiksi henkilöstön työajassa niin paljon, että tällaisia ratkaisuja kannattaisi yritysten pyrkiä ratkaisemaan aktiivisemmin.
Toisena tutkimusaiheena voisi olla, kuinka tällainen Access-pohjainen työkalu soveltuu muiden valmistajien toiminnanohjausjärjestelmiin. Tätä kautta saataisiin myös perusteluja tällaisten aputyökalujen hyödyllisyydestä.

6.2 Tulosten pätevyys ja yleistettävyys

Tutkimuskysymyksenä tässä opinnäytetyössä oli: miten Access 2010-tietokantaohjelmalla voidaan toteuttaa tuotehintoja ERP-järjestelmään päivittävä

helppokäyttötyökalu? Kysymykseen etsittiin vastausta kahden tutkimusongelman avulla. Työssä tarkasteltiin varsin yksityiskohtaisesti työkalun rakentamista ja Accessin käyttämistä sen tekemiseen ja niiden avulla saatiin vastaus tutkimusongelmiin sekä tutkimuskysymykseen. Tutkimuksen tuloksia ei voida pitää yleisesti ainoana oikeana, koska Accessin ominaisuuksista voidaan tarvita käyttöön enemmän tai vähemmän eri tilanteisiin ja tarkoituksiin tehtävissä ERP-työkaluissa. Tutkimustuloksen voidaan katsoa pätevän kokonaisuudessaan ainoastaan tässä tutkimuksessa olleeseen tapaukseen. Tutkimustulos antaa kuitenkin mallin ja kuvaa pääpiirteet kyseisen ohjelman hyödynnettävyydestä ERP-järjestelmän lisätyökaluna. Tulokset kertovat, että Access 2010 avulla pystytään tekemään toimiva työkalu räätälöimään Microsoft Dynamics AX – toiminnanohjausjärjestelmää. Soveltavuudesta muiden valmistajien ERP-järjestelmiin ei tämä tutkimus tuo esiin.

Työkalun määritys vaihe on tehty haastattelemalla yrityksen ERP-manageria ja sitä kautta on etsitty vastaukset työkaluun tarvittaviin ominaisuuksiin. Tässä tulee esille määrittäjäongelma, koska määritykset on tehty ja työkalu rakennettu yhden henkilön avulla. Tosin on muistettava, että tällä henkilöllä on ollut parhaiten tiedossa toiminnanohjausjärjestelmää koskevat asiat, mutta silti voidaan olettaa, että työkalusta olisi tullut jonkin verran erilainen, jos työkalun vaatimuksiin olisi haastateltu myös itse käyttäjiä.

Tutkimuksen etenemiseen vaikutti jonkin verran työkalun testaajien taustat. Vaikka uusi työkalu testattiin useita kertoja valmiina ollessa, työkalun käytön seuraaminen todellisessa tilanteessa, päivitysprosessin aikana, olisi voinut paljastaa joitain kehitettäviä kohtia lisää uuteen työkaluun. Työkalun testaajat ovat nyt olleet aika kokeneita ja IT-koulutuksen saaneita henkilöitä, joka ei vastaa aivan todellista käyttöä, koska kaikilla yrityksen myyjillä ei välttämättä ole niin vahvaa tietoteknistä osaamista.

Tutkimuksen tuottaman datan ja muiden perusteltujen seikkojen avulla voidaan todeta Access 2010 soveltuvan hyvin Microsoft AX ERP-järjestelmän räätälöintiin. Sen avulla pystytään luomaan suuren tietomassan käsittelyyn toimivuudeltaan varma työkalu, johon Accessilla on tarjota monia eri valmiuksia ja ominaisuuksia.

LÄHTEET

Ahtola, J., Rajamäki, M. 2011. Liiketoiminnan muuttuvat tarpeet pakottavat investoimaan ERP-järjestelmiin. Marketvisio 7.9.2011. [viitattu 3.4.2014]. Saatavissa: http://www.marketvisio.fi/fi/ajankohtaista/uutiset-marketvisio/866-liiketoiminnan-muuttuvat-tarpeet-pakottavat-investoimaan-erp-jarjestelmiin

Aladwani, A. M. 2001. Change management strategies for successful ERP implementation. Kuwait University. [viitattu 29.1.2014]. Saatavissa: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.5630&rep=rep1&type=pdf&embedded=true

Couch, A. 2011. Microsoft Access 2010 VBA Programming Inside Out. Sebastopol, California: O'Reilly Media, Inc.

Foster, S. Hawking, P., Stein, A. 2004. Revisiting ERP Systems: Benefit Realisation. Victoria University & Monash University. [viitattu 31.1.2014]. Saatavissa: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.387&rep=rep1&type=pdf

Hannu, V. 2013. Koeajon tietojärjestelmä. Opinnäytetyö. Tampereen ammattikorkeakoulu. [viitattu 26.1.2014]. Saatavissa: http://publications.theseus.fi/bitstream/handle/10024/58067/Hannu_Ville.pdf?sequence=1

Hevner A., March S., Park J. 2004. Design Science in Information Systems Research. Lahden Ammattikorkeakoulu: Reppu opintomateriaali. [viitattu 26.1.2014]. Saatavissa: http://reppu.lamk.fi/mod/folder/view.php?id=314245

Huotari J. 1999–2013. SQL-kielen perusteet. [viitattu 16.1.2014]. Jyväskylän Ammattikorkeakoulu. Saatavissa: http://homes.jamk.fi/~huojo/opetus/IIZO3030/SQLopas.pdf

Järvelä, S. 2012. Toiminnanohjausjärjestelmän käytettävyyden kehittäminen. Opinnäytetyö. Tampereen ammattikorkeakoulu. [viitattu 26.1.2014]. Saatavissa: http://publications.theseus.fi/bitstream/handle/10024/45257/Jarvela_Sanna.pdf?sequence=1

Järvinen, P., Järvinen, A. 2011. Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Keinonen, K.J. 2010. Microsoft Access 2010. Edistynyt käyttö. Ornanet Koulutuksen e – kirjat. Turku: DatumPoint.

Kontio, M. 2013. ERP-Manager. Peikko Group Oy. Haastattelut: 1.8.2013, 8.8.2013, 30.8.2013 ja 20.11.2013.

Microsoft. 2012. Microsoft Dynamics AX. Microsoft Corporation. [viitattu 18.12.2013]. Saatavissa: http://www.microsoft.com/dynamics/fi/fi/products/ax-overview.aspx

Microsoft. 2012. Microsoft Dynamics ERP. Microsoft Corporation. [viitattu 18.12.2013]. Saatavissa: http://www.microsoft.com/dynamics/fi/fi/erp.aspx

Microsoft. 2014. Tietoja SQL-kyselyistä (MDB). Microsoft Corporation. [viitattu 31.3.2014]. Saatavissa: http://office.microsoft.com/fi-fi/access-help/tietoja-sql-kyselyista-mdb-HP005188412.aspx

Osintsev, A. 2012. Usability Still a Problem for ERP Users. Technology Evaluation Centers 20.11.2012. [viitattu 29.1.2014]. Saatavissa: http://www.technologyevaluation.com/research/article/Usability-Still-a-Problem-for-ERP-Users.html

Moilanen J. 2002. Sql-tietokannan viritys. MicroPC nettilehti 14/2002. [viitattu 5.3.2014]. Saatavissa: http://mpc.fi/nettilehti/pdf/pc2111200252.pdf

Mäntysaari, L. 2013. CGI ja Axapta vahvoja alle 500 hengen yritysten ERPissä. Marketvisio 17.6.2013. [viitattu 26.1.2013]. Saatavissa: http://www.marketvisio.fi/fi/ajankohtaista/uutiset-marketvisio/1722-cgi-ja-axapta-vahvoja-alle-500-hengen-yritysten-erpiss

Peikko. 2014. Betoniliitosten ja liittopalkkien asiantuntija. Peikko Group. [viitattu 29.1.2014]. Saatavissa: www.peikko.fi/tietoa-peikosta

Peikko. 2014. Luotettava kumppani. Peikko Group. [viitattu 29.1.2014]. Saatavissa: http://www.peikko.fi/tietoa-peikosta/peikko-lyhyesti

Peikko. 2014. Tavoitteena johtava markkina-asema ja kannattava kasvu. Peikko Group. [viitattu 29.1.2014]. Saatavissa: http://www.peikko.fi/tietoa-peikosta/strategia

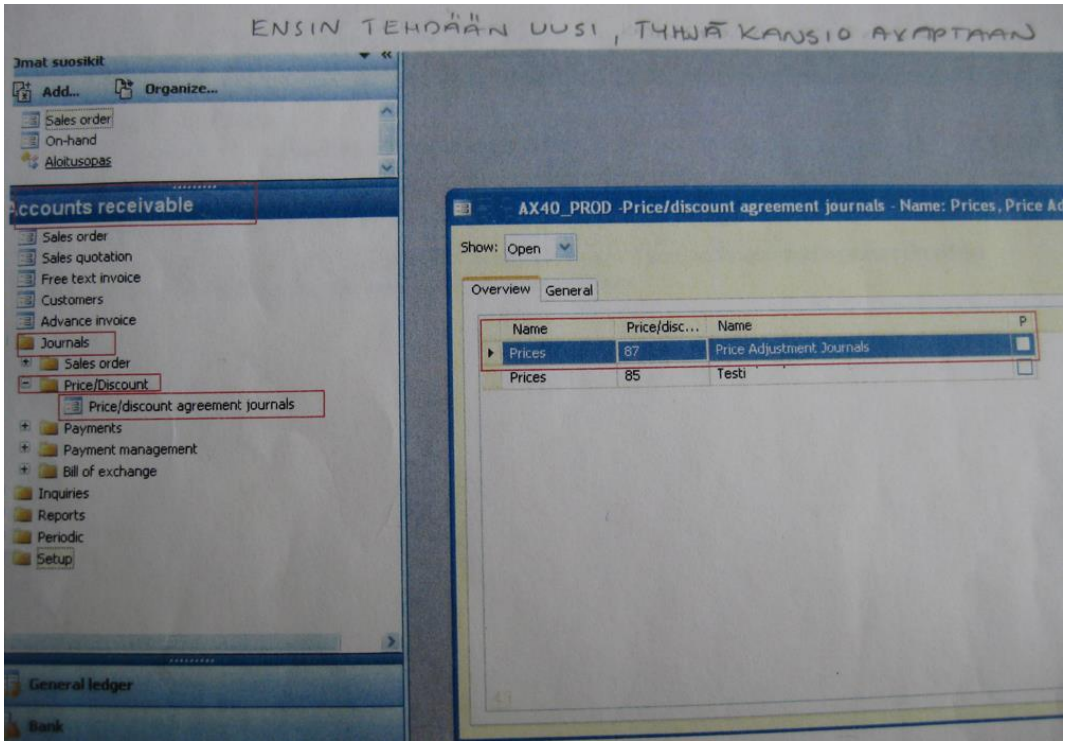
Wikipedia. 2013. Microsoft Access. [viitattu 18.12.2013]. Wikimedia Foundation. Saatavissa: http://fi.wikipedia.org/wiki/Access

Saaranen-Kauppinen A., Puusniekka A. 2006. KvaliMOTV - Menetelmäopetuksen tietovaranto [verkkojulkaisu]. [viitattu 11.1.2014]. Tampere: Yhteiskuntatieteellinen tietoarkisto [ylläpitäjä ja tuottaja]. Saatavissa: http://www.fsd.uta.fi/menetelmaopetus/kvali/L6_3_1.html

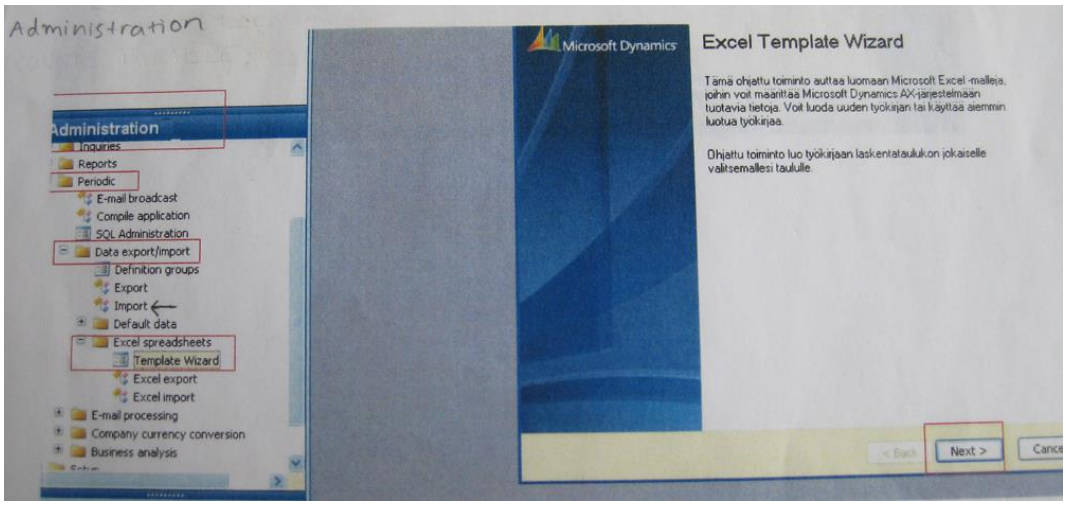
LIITTEET

LIITE 1. Ohje ERP:n päivitystyökalujen käyttöön

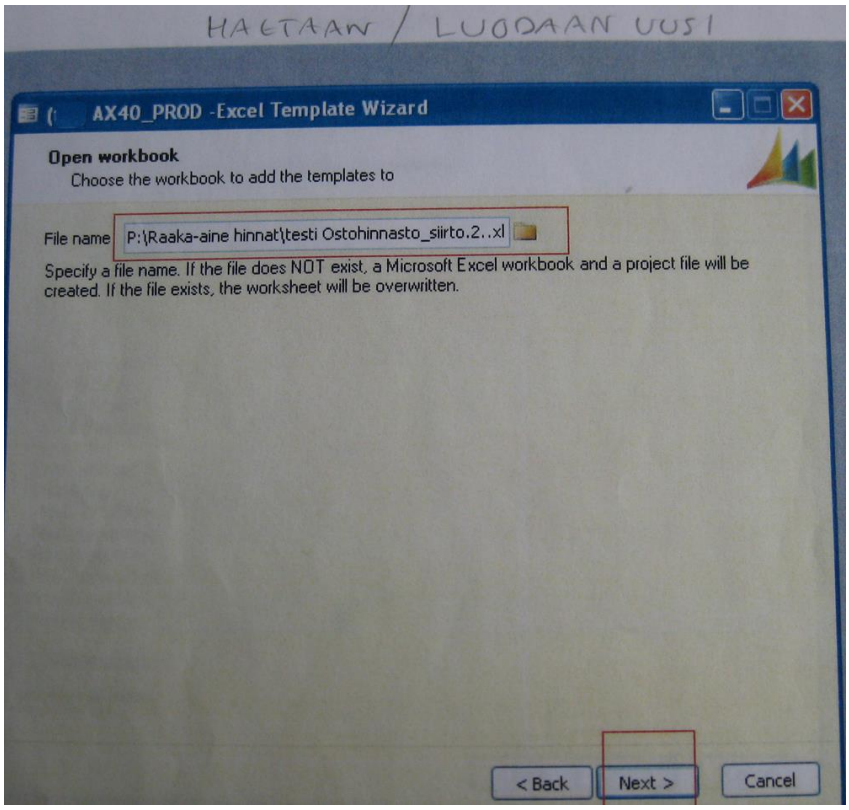
Lomake 1/12:



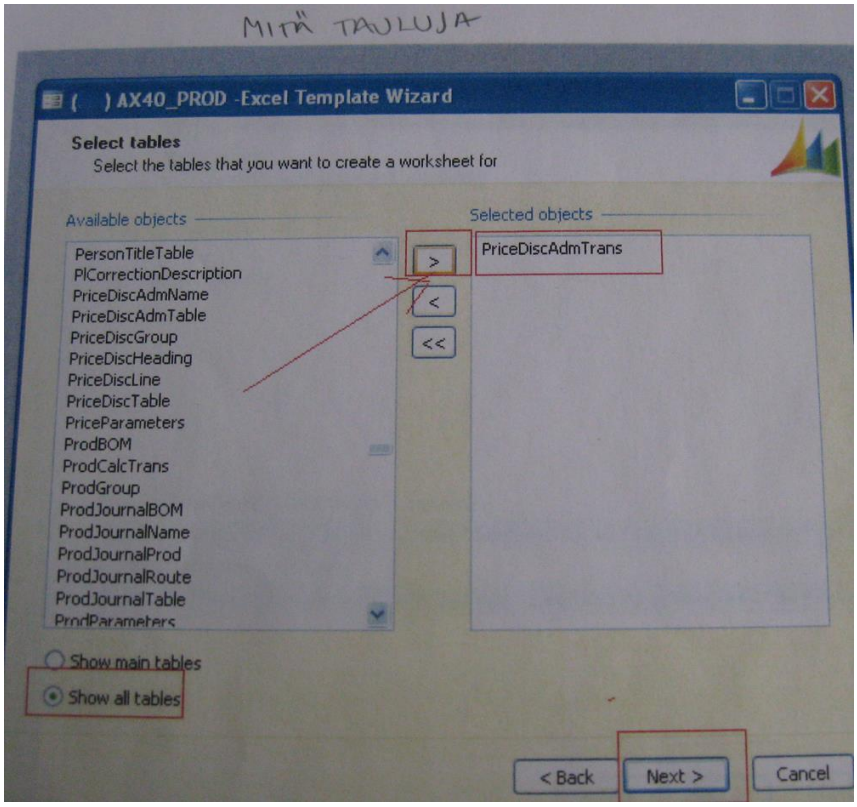
Lomake 2/12:



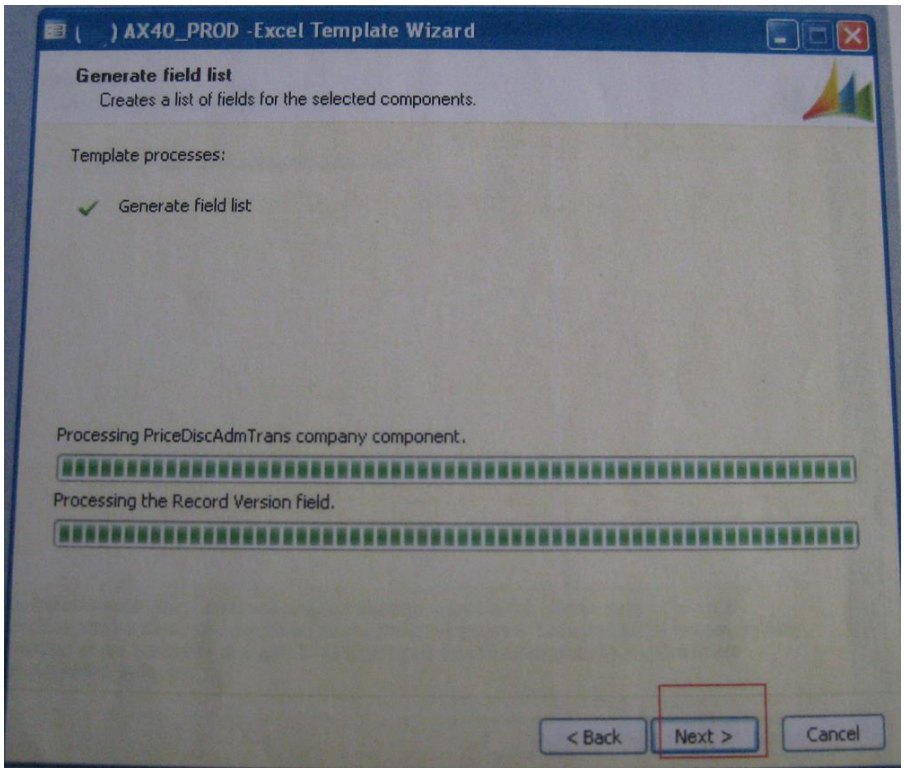
Lomake 3/12:



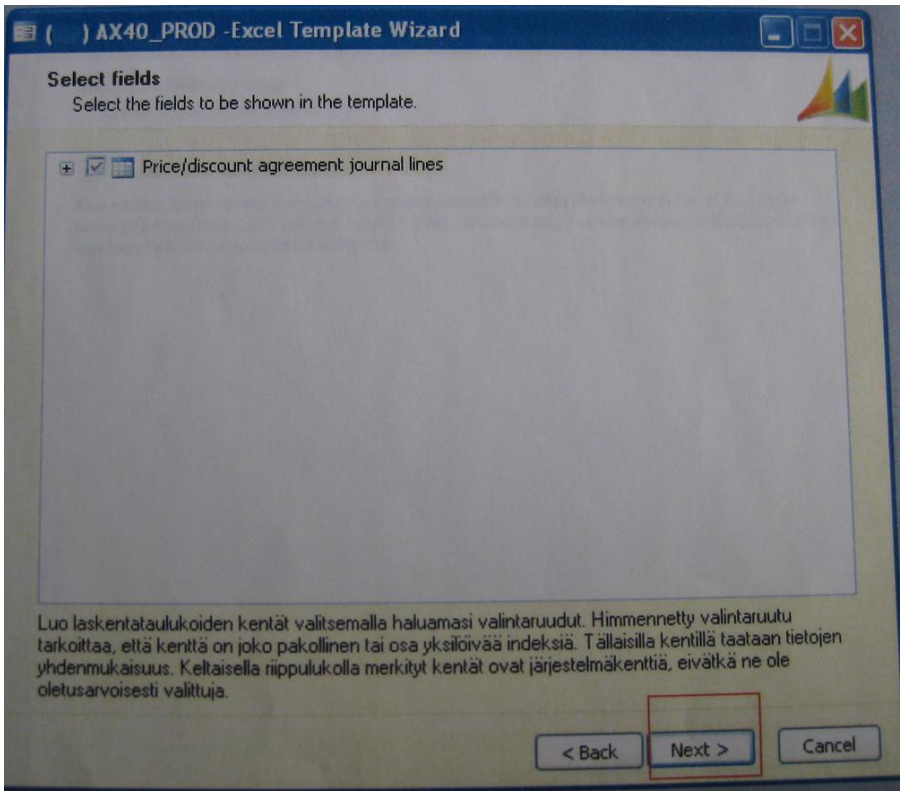
Lomake 4/12:



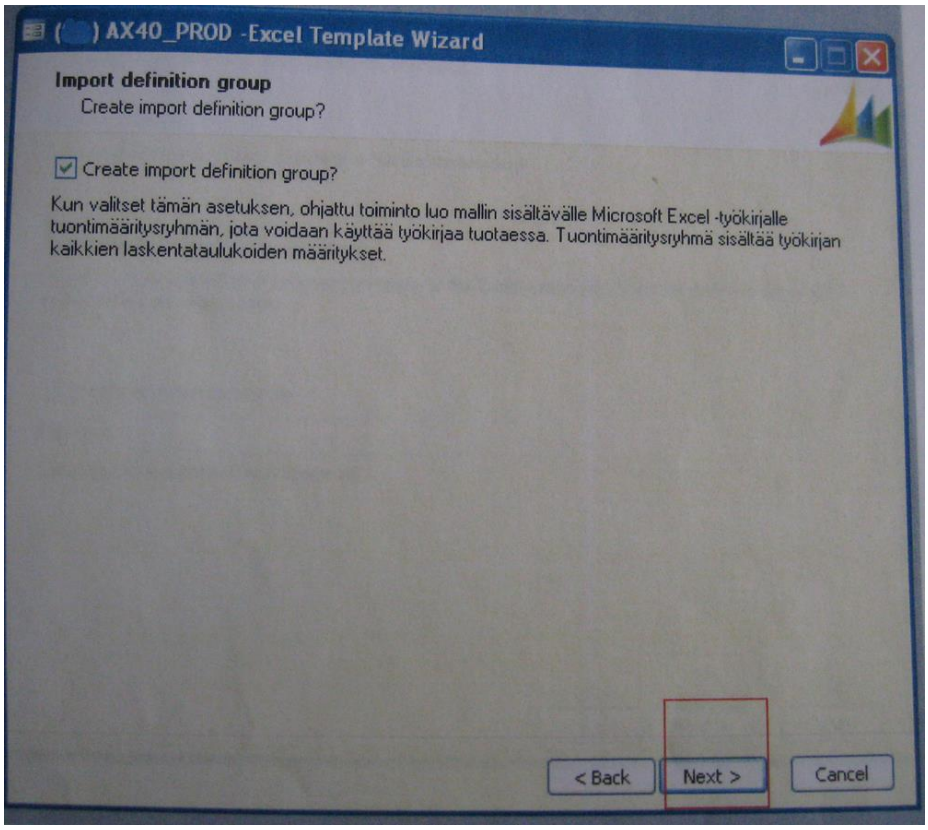
Lomake 5/12:



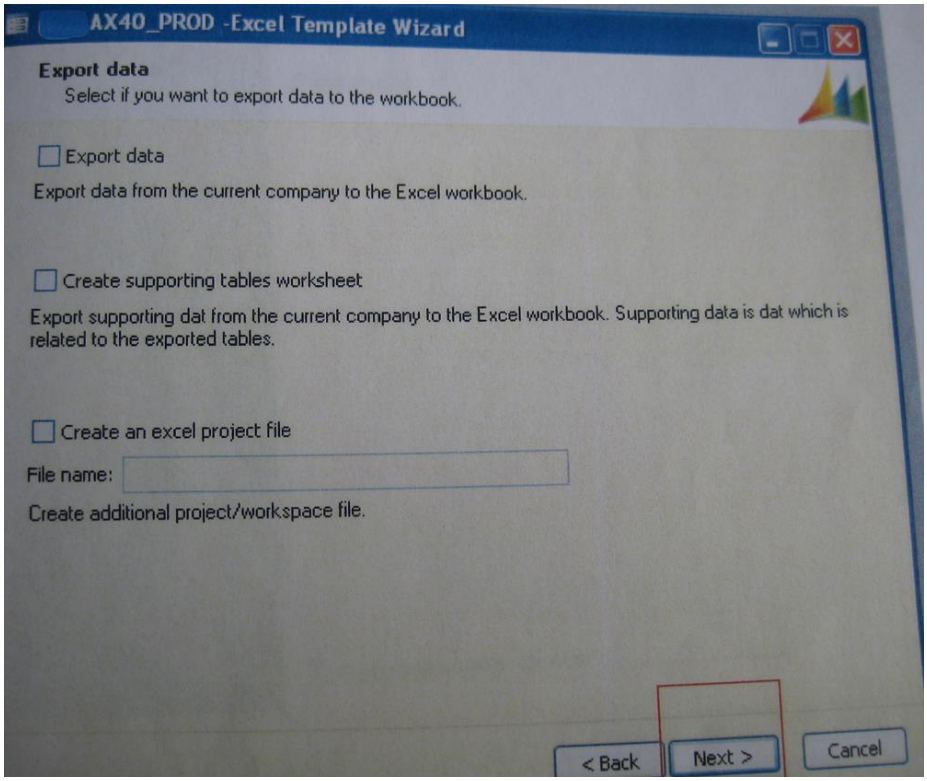
Lomake 6/12:



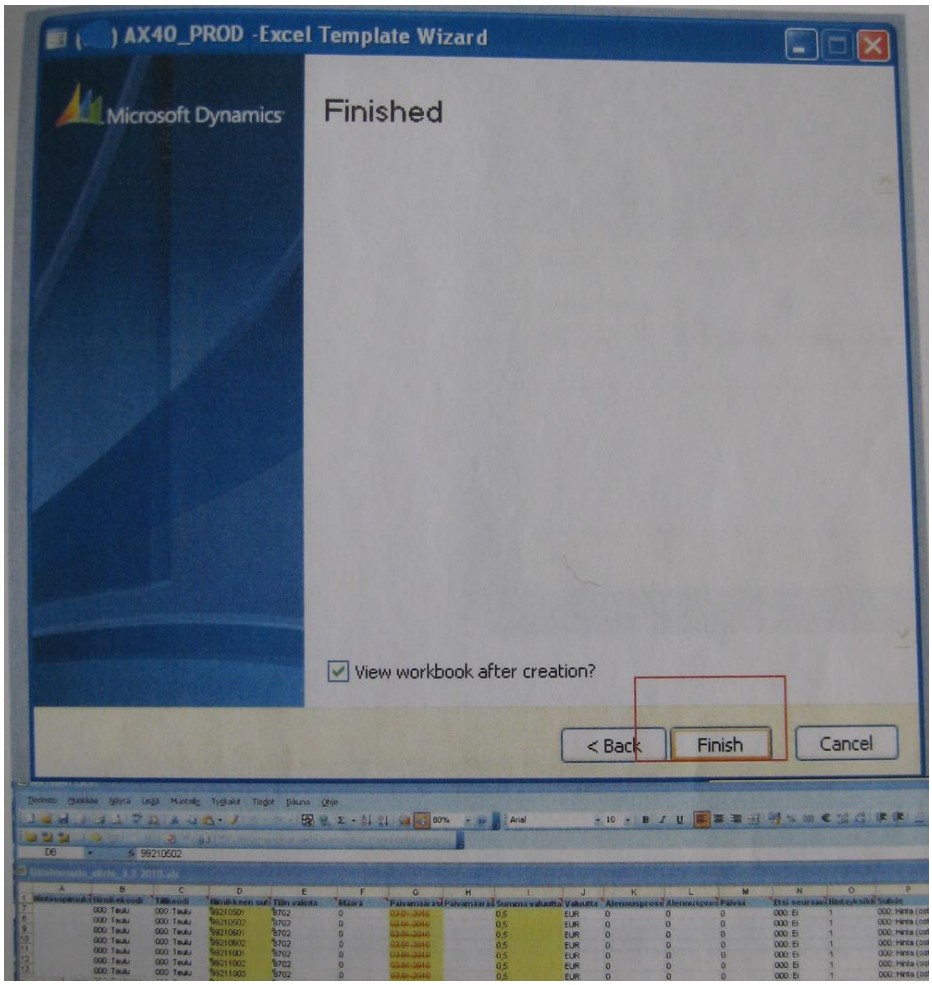
Lomake 7/12:



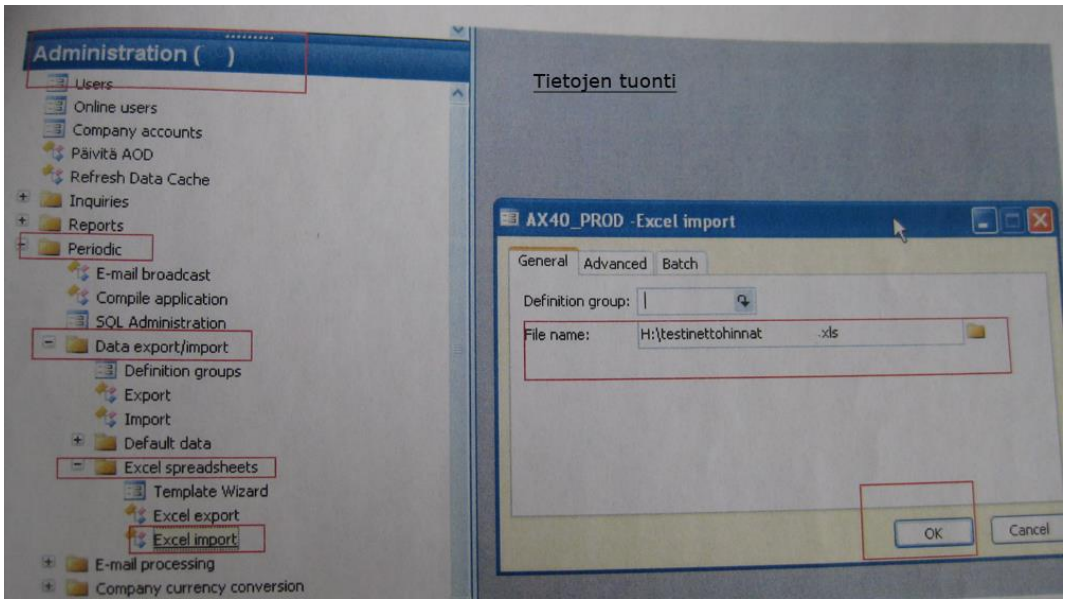
Lomake 8/12:



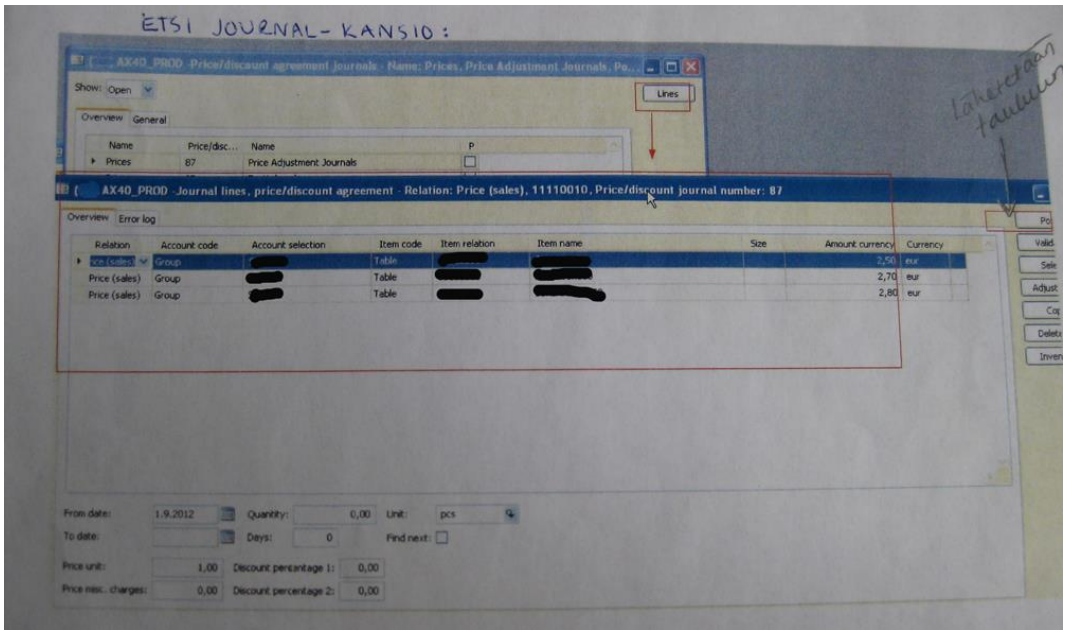
Lomake 9/12:



Lomake 10/12:



Lomakkeet 11–12/12:



LIITE 2. Uuden työkalun käyttöohjeet

Sivu 1/3:

THE MANUAL FOR PRICES UPDATE APPLICATION

We have developed the new tool which is intended to help updating prices to AX. The main principle of the tool is that you can easily fetch purchase and sales prices from AX, to handle price data in Excel and return it back to AX.

An application includes two forms; the first form searches the price data what you want to handle and the second form returns it back to the application and makes the transfer file. Forms are easy to use; only things you should to do is push the buttons, copy-paste prices to Excel and return the data back to the application. An application does transfer file automatically when you click "make transfer file" button. After transfer file making, you log in to AX and import the data. The file, that you should to import, is named "PriceFile". The first form shows you the file path where you can find that file.

This application is protected. It means that users of the application are inserted to the SQL-database and others can't get in to the application. If you don't get to use application, it may mean that you are not a user in SQL yet - please contact the data administration.

USE OF THE TOOL – STEP BY STEP

The first step is to create journal to AX, where you import data after handling. You need to know "price journal number" later.

"GetData" -form

On the first form, you have a choice of five buttons and five text boxes where you can insert some searching definitions. The price data of AX comes to this form.
On the top of form, you see the file path where the transfer file will be located and the
company where the purchase prices are fetched (can't be changed on the form).

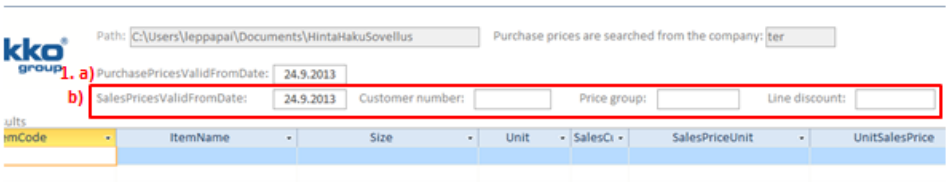


Figure 1. Top of the first form

1. CHOOSE WHAT PRICES YOU WANT TO FETCH FROM AX

a) GET PURCHASE PRICES
- Insert the date to "PurchasePricesValidFromDate" text box [Figure 1: 1.a]. If you want, you can keep the default value which is this day.
- Click the button – application searches purchase price data from company which you can see on the top of the form.

b) GET SELLING PRICES
- Insert the date to "SalesPricesValidFromDate" text box [Figure 1: 1.b]. If you want, you can keep default value which is this day.
- Insert the search value you want to use (Customer number, Price group or Line discount). If any value is not given, an application fetches the general prices valid for "All".
- Click the "Get selling prices"-button – an application searches selling prices from your home company

c) GET PURCHASE AND SELLING PRICES
- Insert search values as described in earlier sections a) and b)
- Click the "Get purchase and selling prices" -button – an application searches both prices from AX

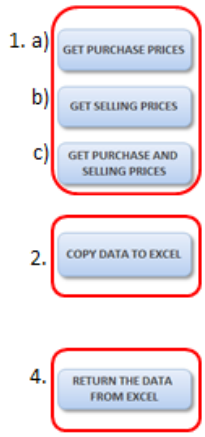


Figure 2. Buttons of the first form

2. COPY DATA TO EXCEL

- After data fetch from AX, click the button "copy data to Excel" – the new data table will opened and data is arranged.
- Choose all data (clicking grey triangle from the top left) from the table and copy it.
- Now you can open the Excel-file from your own computer where you want to handle this data - paste data there.
- Close the price update application

3. MODIFY DATA IN EXCEL

- After you have updated prices: Delete unnecessary columns (if this is not done, data transfer will fail) – leave only the following:

  - ItemCode
  - Size
  - PriceUnit
  - NetPrice

- Copy data

"Main_ReturnExcelData" -form

4. RETURN DATA

- Open price update application
- Click "return the data from Excel" –button – "Main_ReturnExcelData" form will opened
- Click the first row on the ReturnExcelData –form
- Paste data by clicking green "paste append" shortcut. It located on the top toolbar:

Figure 3. Paste append -shortcut

5. MAKE A TRANSFER FILE

The next, there is defined some values to the transfer file:

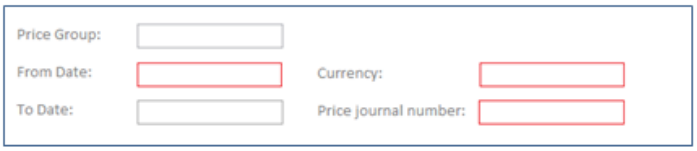


Figure 4. Definitions of the second form

- "From Date", "Currency" and "Price journal number" –fields **are required.**
- Application checks that "Price journal number" has already exist in AX - if it is not, you have to create it first.
- If you get the message box "The Transfer file is ready now…" you can close an application.

6. LOG IN TO AX AND IMPORT THE DATA