

Kenji Saito

Mobiilipäätteen tiedonsiirtosovellus

Ongelmien kartoitus ja kehityssuunnitelman luonti

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikan koulutusohjelma
Insinöörityö
Päivämäärä

Tekijä(t) Otsikko	Kenji Saito Mobiilipäätteen tiedonsiirtosovellus
Sivumäärä Aika	23 sivua + 2 liitettä 27.4.2012
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Pääsuunnittelija Pekka Tiainen Yliopettaja Auvo Häkkinen
<p>Insinööriyö tehtiin mobiilipäätteisiin perustuviin tiedonkeruuratkaisuihin erikoistuneelle yritykselle. Työssä kartoitettiin yrityksen kehittämän tiedonsiirtosovelluksen ongelmat, etsittiin mahdollisia ratkaisuja havaittuihin ongelmiin ja tehtiin suunnitelma ongelmien ratkaisemiseksi. Osa työn tuloksista on vain työn tilaajan käyttöön eikä sisälly kirjalliseen raporttiin.</p> <p>Ongelmien kartoitus suoritettiin haastattelemalla yrityksen sovellusasiantuntijoita. Ongelmien todettiin liittyvän pääasiassa sovelluksen käytettävyyteen, tiedonsiirron luotettavuuteen sekä puutteelliseen versionhallintaan ja dokumentointiin.</p> <p>Ratkaisuiksi ehdotettiin sovelluksen käyttöliittymän ja rajapintojen uudelleensuunnittelua tunnettujen käytettävyyssperiaatteiden mukaisesti, versionhallinta- ja dokumentointikäytäntöjen muuttamista sekä käytettävän tiedonsiirtoteknologian vaihtamista. Ratkaisut on mahdollista toteuttaa toisistaan erillisinä projekteina.</p> <p>Mahdolliset vaihtoehtoiset tiedonsiirtoteknologiat selvitettiin. Potentiaalisimman vaihtoehdon soveltuvuutta käytettävässä toimintaympäristössä tutkittiin tarkemmin, ja se todettiin ominaisuuksiltaan käytössä olevaa tiedonsiirtoteknologiaa paremmaksi.</p> <p>Uusi tiedonsiirtoteknologia mahdollistaa TCP/IP-protokollan käytön tiedonsiirrossa. TCP/IP-protokollan avulla tiedonsiirtosovelluksen toimintaympäristöä voidaan laajentaa. Uudella teknologialla voidaan myös välttää osa vanhaan teknologiaan liittyvistä käytettävyysongelmista.</p> <p>Uudesta TCP/IP-protokollaan perustuvasta tiedonsiirtosovelluksesta tehtiin suunnitelma, joka perustuu käytössä olevan sovelluksen toiminnallisiin vaatimuksiin. Suunnitelmassa on myös uusia toiminnallisuuksia, jotka TCP/IP-protokollan käyttö tekee mahdolliseksi.</p>	
Avainsanat	mobiilipäätte, Windows CE, tiedonsiirto, USB, TCP/IP

Author(s) Title	Kenji Saito Data Transfer Application for Mobile Device
Number of Pages Date	23 pages + 2 appendices 27 April 2012
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Pekka Tiainen, Lead Designer Auvo Häkkinen, Principal Lecturer
<p>This thesis was done for a company which is specialized in mobile device based data collection solutions. In this thesis, problems related to a data transfer application designed by the company were mapped out, possible solutions were studied and a plan to solve the problems was made.</p> <p>The mapping out of the problems was accomplished by surveying the software specialists of the company. The problems were discovered to be mostly related to the usability of the application, the reliability of the data transmission and both inadequate version control and documentation practices.</p> <p>Suggested solutions include redesigning of the user and programming interfaces according to common usability principles, changing of the version control and documentation practices and replacing of the used data transmission technology. The solutions can be realized as individual projects separate from each other.</p> <p>Plausible alternatives for data transmission technology were examined. The most potential candidate was further studied for suitability, and its features were found to be better than those of the currently used technology.</p> <p>The new data transmission technology enables the use of TCP/IP protocol in data transfer. With TCP/IP protocol, the operational environment of the data transfer application can be extended. The new technology also eliminates some of the usability problems caused by the current technology.</p> <p>A plan of a new data transfer application based on TCP/IP protocol was made. The plan is based on the functional requirements of the data transfer application currently in use, but includes new functionality made possible by the usage of TCP/IP protocol.</p>	
Keywords	mobile device, Windows CE, data transmission, USB, TCP/IP

Sisällys

Lyhenteet

1	Johdanto	1
2	Mobiilipäätteisiin perustuvat tiedonkeruuratkaisut	2
2.1	Arkkitehtuuri	3
2.2	Mobiilipäätteet	3
2.2.1	Käyttöjärjestelmä	4
2.2.2	Laitteisto	4
2.3	Työasema	5
3	Nykyinen tiedonsiirtosovellus	6
3.1	Tiedonsiirtoprotokolla	6
3.2	Työasemasovellus	7
3.3	Tiedonsiirrossa käytettävät teknologiat	7
4	Keskeisimmät ongelmat ja puutteet	8
4.1	Sovelluksen asentaminen ja konfigurointi	8
4.2	Tiedonsiirron toimintavarmuus	9
4.3	Työasemasovelluksen käytettävyys	9
4.4	Tiedonsiirtorajapinnan käytettävyys	10
4.5	Versionhallinta	10
4.6	Johtopäätökset	11
5	Ehdotukset ongelmien ratkaisemiseksi	12
5.1	Työasemasovelluksen käyttöliittymän uudelleensuunnittelu	12
5.2	Tiedonsiirtorajapinnan uudelleensuunnittelu	12
5.3	Tiedonsiirtoteknologian vaihtaminen	12
5.4	Versionhallinta ja riittävä dokumentointi	14
5.5	Johtopäätökset	15
6	USB:hen perustuva tiedonsiirto	16
6.1	USB:n arkkitehtuuri	16
6.2	Mobiilipäätteiden sisältämät USB Function -ajurit	17

6.3	Muut vaihtoehdot USB-tiedonsiirron toteuttamiseksi	18
6.4	Johtopäätökset	18
7	TCP/IP-protokollaan perustuva tiedonsiirtosovellus	19
7.1	Vaatimukset	19
7.2	Toimintaympäristö ja käytettävät teknologiat	21
8	Yhteenveto	22
	Lähteet	23
	Liitteet	
	Liite 1. Mobiilipäätteiden USB Function -ajurit, 12 sivua (ei julkinen)	
	Liite 2. Tiedonsiirtosovelluksen tekninen dokumentaatio, 20 sivua (ei julkinen)	

Lyhenteet ja määritelmät

ActiveSync	Microsoftin kehittämä teknologia ja protokolla mobiililaitteen tietojen synkronointiin.
CE	<i>Compact Edition</i> , Microsoftin kehittämä käyttöjärjestelmä mobiilipäätteille.
Bluetooth	Langaton tiedonsiirtoprotokolla.
DLL	<i>Dynamic Link Library</i> . Ajonaikainen sovelluskirjasto Microsoftin käyttöjärjestelmissä.
GPRS	<i>General Packet Radio Service</i> . Pakettikytkentäinen, langaton tiedonsiirtopalvelu.
HID	<i>Human Interface Device</i> . USB-laiteluokka, johon sisältyvät käyttöjärjestelmän hallintaan tarkoitettut laitteet kuten hiiri ja näppäimistö.
IrDA	<i>Infrared Data Association</i> . Infrapunasäteilyyn perustuva langaton tiedonsiirtoprotokolla.
.NET	Microsoftin kehittämä ohjelmointikehys.
RAPI	<i>Remote Application Programming Interface</i> . Microsoftin kehittämä ohjelmointirajapinta.
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i> . Tietoverkkoprotokollien yhdistelmä, jota käytetään Internet-tietoverkon liikennöinnissä.
USB	<i>Universal Serial Bus</i> . Sarjaväyläarkkitehtuuri tietokoneeseen liitettäville oheislaitteille.
WLAN	<i>Wireless Local Area Network</i> . Langaton lähiverkkoteknologia.

1 Johdanto

Opinnäytetyö tehtiin yritykselle, joka toimittaa asiakkailleen räätälöityjä ratkaisuja tiedon keräämiseen ja tallentamiseen mobiiliympäristössä. Tiedon keräämiseen käytetään viivakoodilukijalla varustettuja mobiilipäätteitä, joista tieto siirretään edelleen työasemalle. Tiedonsiirtoon käytetään yrityksessä kehitettyä tiedonsiirto-sovellusta, jolla voidaan toteuttaa kaksisuuntainen tiedostosiirto mobiilipäätteen ja työaseman välillä.

Tiedonsiirto-sovelluksessa on ilmennyt luotettavuus- ja käytettävyyso ongelmia, jotka aiheuttavat lisätyötä sovelluksen asennus- ja ylläpitovaiheissa. Opinnäytetyön tavoitteena oli kartoittaa sovellukseen liittyvät ongelmat, tutkia mahdollisia ratkaisuja sekä tehdä suunnitelma ongelmien ratkaisemiseksi.

Opinnäytetyö jakautui neljään vaiheeseen. Ensimmäisessä vaiheessa kartoitettiin tiedonsiirto-sovelluksen ongelmat. Toisessa vaiheessa etsittiin mahdollisia ratkaisuja ongelmiin. Kolmannessa vaiheessa keskityttiin tutkimaan uuteen tiedonsiirtoteknologiaan perustuvaa ratkaisua. Neljännessä vaiheessa tehtiin suunnitelma uudesta tiedonsiirto-sovelluksesta, joka perustuu tutkittuun tiedonsiirtoteknologiaan.

Opinnäytetyössä kuvataan tyypilliset tiedonkeruuratkaisut ja niissä käytettävä tiedonsiirto-sovellus. Haastatteluiden perusteella selvitettyt keskeisimmät ongelmat esitetään ehdotettuine ratkaisuineen. Vaihtoehtoiset USB-sarjaväyläarkkitehtuuriin perustuvat tiedonsiirtoteknologiat esitetään lyhyesti – tarkempi selvitys, joka sisältää yksityiskohtaista tietoa mm. valitun tiedonsiirtoteknologian käyttöönotosta, ei sisälly opinnäytetyön julkiseen osaan.

Uusi tiedonsiirtoteknologia mahdollistaa TCP/IP-protokollan käytön tiedonsiirrossa. TCP/IP-protokollaan perustuvasta tiedonsiirto-sovelluksesta tehtiin suunnitelma, joka kuvataan lyhyesti. Sovelluksen tekninen dokumentaatio, jossa on esitetty mm. sovellusarkkitehtuuri, ohjelmointirajapinnat ja pakettiperustainen tiedonsiirtoprotokolla, ei sisälly työn julkiseen osaan.

2 Mobiilipäätteisiin perustuvat tiedonkeruuratkaisut

Yrityksen asiakkaat toimivat monilla eri toimialueilla kuten teollisuudessa, terveydenhuollossa ja kuljetuksessa. Asiakkaiden tarvitsemat tiedonkeruuratkaisut ovat kuitenkin luonteeltaan samankaltaisia: tuotettu tieto tallennetaan, jonka jälkeen se välitetään haluttuun paikkaan. Tiedonvälitys voi olla kaksisuuntaista.

Tiedon keräämiseen ja välittämiseen käytetään mobiilipäätteitä ja niissä suoritettavia asiakaskohtaisia sovelluksia. Mobiilipäätteen avulla tietoa voidaan tuottaa ja kuluttaa paikasta riippumatta, esim. varastossa tai sairaalan osastoilla liikuttaessa.

Hyvin yleinen tiedonkeruutarve liittyy varastointiin: varastossa olevien tuotteiden lukumäärää halutaan seurata. Käytännössä tämä tarkoittaa varastoon saapuvien ja varastosta otettavien tuotteiden kirjaamista sekä varaston säännöllistä inventointia. Esimerkiksi varaston inventointi voidaan suorittaa seuraavalla tavalla:

- 1) Inventoitavan tuotteen viivakoodi luetaan mobiilipäätteen viivakoodinlukijalla.
- 2) Inventoitavan tuotteen kokonaismäärä varastossa lasketaan ja syötetään mobiilipäätteeseen.
- 3) Syötetyt tiedot tallennetaan ja välitetään tietokantaan.

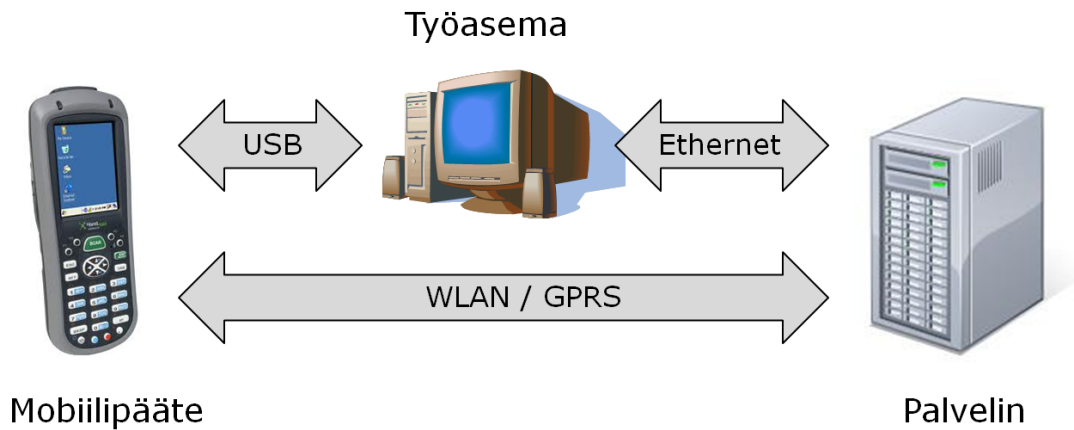
Toinen yleinen tiedonkeruutarve liittyy tilausten tekemiseen. Esimerkiksi sairaalan osastoille tilattavat tuotteet voidaan tilata seuraavalla tavalla:

- 1) Osaston tunnus luetaan mobiilipäätteen viivakoodinlukijalla.
- 2) Tilattava tuote valitaan mobiilipäätteessä olevasta tuoterekisteristä.
- 3) Tilattavan tuotteen lukumäärä syötetään mobiilipäätteeseen.
- 4) Syötetyt tiedot tallennetaan ja välitetään tilausjärjestelmään.

Tässä luvussa kuvataan mobiilipäätteisiin perustuvien tiedonkeruuratkaisujen arkkitehtuuri ja ratkaisussa käytettävät komponentit.

2.1 Arkkitehtuuri

Tiedonkeruuratkaisut perustuvat mobiilipäätteisiin, joiden avulla tuotettu tieto kerätään ja välitetään eteenpäin. Kuva 1 esittää tiedonsiirtoratkaisun arkkitehtuuria.



Kuva 1. Tiedonsiirtoratkaisun arkkitehtuuri

Tieto voidaan WLAN- ja GPRS-yhteyksien avulla välittää palvelimelle, jossa on jokin tietoa tallentava tai analysoiva järjestelmä. Tieto voidaan myös siirtää ensin työasemalle USB-yhteyden avulla, jonka jälkeen se siirretään palvelimelle Ethernet-yhteyden avulla. Tieto voidaan hyödyntää myös suoraan työasemalla.

2.2 Mobiilipäätteet

Mobiilipäätteellä voidaan tarkoittaa periaatteessa mitä tahansa mukana kuljetettavaa laitetta, jossa on tiedon syöttämiseen ja/tai näyttämiseen tarvittava fyysinen käyttöliittymä. Yksi tällainen on matkapuhelin.

Tiedonkeruuratkaisuissa käytettävät mobiilipäätteet muistuttavat ominaisuuksiltaan matkapuhelimia. Ammatillinen käyttö, esimerkiksi teollisuus- tai varastoympäristössä, asettaa kuitenkin rajoitukset laitteen fyysiselle kestävyydelle. Lisäksi tiedon keräämistä varten laitteessa on oltava viivakoodinlukija.

Mobiilipäätteistä on useita eri malleja, joiden ominaisuudet eroavat toisistaan niiden käyttöjärjestelmän ja laitteiston osalta. Seuraavat luvut kuvaavat yrityksen käytössä olevia mobiilipäätteitä.

2.2.1 Käyttöjärjestelmä

Kaikissa yrityksen käyttämissä mobiilipäätteissä on Microsoftin kehittämä mobiilikäyttöjärjestelmä, joka perustuu Windows CE -käyttöjärjestelmään [1, s. 6]. Käytössä olevat versiot ovat

- Windows Mobile 6 (WM6)
- Windows Mobile 6.5 (WM6.5)
- Windows Embedded Handheld 6.5 (WEH6.5).

Windows CE -käyttöjärjestelmät tukevat useita eri teknologioita, kuten

- .NET Compact Framework -sovelluskehystä, joka on suppeamman käsikytkänsä omaava versio työpöytäkäyttöön tarkoitettusta .NET-sovelluskehyksestä
- Global Positioning Systemiä (GPS)
- Microsoft SQL Server -tietokantaa
- TCP/IP, Winsock ja Remote API -tiedonsiirto-protokollia [1, s. 6 - 8].

Tuettujen teknologioiden avulla on mahdollista suunnitella monenlaisia kannettavia laitteita, kuten kameroita, puhelimia ja multimedialaitteita [1, s. 9 - 10].

2.2.2 Laitteisto

Mobiilipääte on tietokone, jossa on prosessori, keskusmuisti sekä lukuisia muita tiedonkäsittelyyn ja laitteiston hallintaan vaadittavia integroituja piirejä. Kaikissa yrityksen käyttämissä mobiilipäätteissä on kosketusnäyttö, näppäimistö ja viivakoodilukija, jotka ovat oleellisia tiedon esittämisessä ja syöttämisessä. Kaikki päätteet sisältävät USB- ja RS-232-tiedonsiirtotoiminnallisuuden toteuttavat komponentit, joita käytetään langalliseen tiedonsiirtoon. Pätteet voivat sisältää myös langattoman tiedonsiirron toteuttavia komponentteja, kuten WLAN-, GPRS-, Bluetooth- ja IrDA-komponentteja.

USB-toiminnallisuus on toteutettu USB-laiteohjaimen avulla, mikä rajoittaa mobiilipäätteen toimimisen vain USB-laitteena. Tämä tarkoittaa sitä, että mobiilipääte ei voi toimia USB-isäntänä toiselle USB-laitteelle, kuten hiirelle, näppäimistölle tai USB-muistitikulle [7].

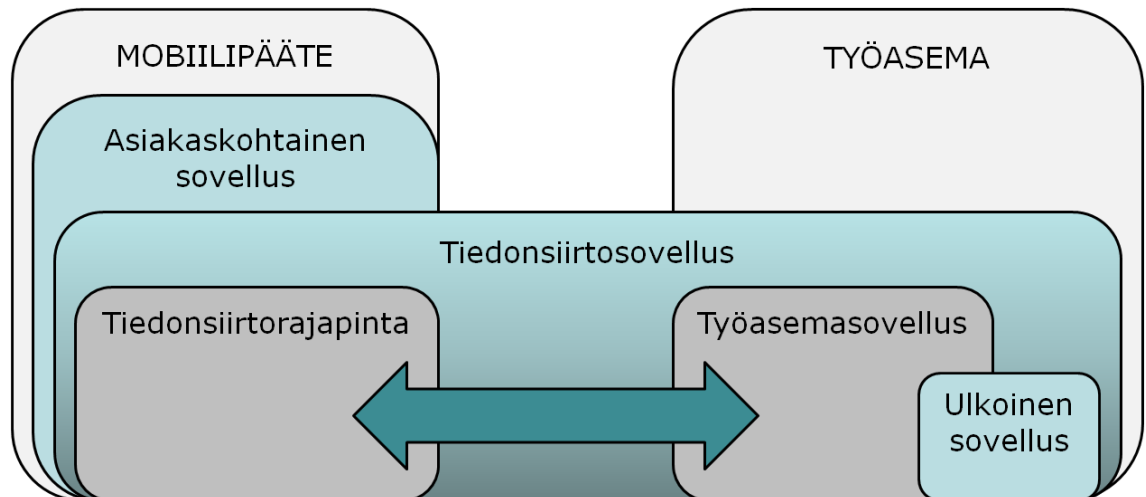
2.3 Työasema

Työasema on tyypillisesti työpöytätietokone, jonka käyttöjärjestelmä on Microsoftin Windows XP, Windows Vista tai Windows 7. Työasema sisältää tiedonsiirtoon vaadittavan USB-liitännän. Mikäli asiakaskohtainen ratkaisu sitä vaatii, on työasemassa myös oltava ulkoiseen verkkoon kytketty verkkolaite, kuten WLAN- tai Ethernet-kortti. Työasema voi sisältää myös näytön ja syötteiden antamiseen käytetyt oheislaitteet kuten hiiren ja näppäimistön, joskaan nämä eivät ole tiedonsiirtosovelluksen toiminnan kannalta olennaisia osia.

3 Nykyinen tiedonsiirtosovellus

Tiedonsiirtosovellusta käytetään tiedon siirtämiseen mobiilipäätteen ja työaseman välillä. Siirrettävä tieto on kerätty paikallisiin tiedostoihin, joiden siirtäminen ei vaadi jatkuvaa siirtoyhteyttä. Tiedostojen muodolla ja sisällöllä ei ole merkitystä. Ne voivat olla esim. teksti-, kuva- tai tietokantatiedostoja. Tiedonsiirto on kaksisuuntaista.

Kuva 2 havainnollistaa tiedonsiirtosovelluksen arkkitehtuuria.



Kuva 2. Tiedonsiirtosovelluksen arkkitehtuuri

Tiedonsiirtosovellus koostuu työasemasovelluksesta ja tiedonsiirtorajapinnasta. Nämä kaksi komponenttia pysyvät samoina toteutettavasta ratkaisusta riippumatta. Asiakaskohtaiseen sovellukseen lisätään tiedonsiirron toteuttava toiminnallisuus tiedonsiirtorajapinnan avulla. Työasemasovellus ei sisällä asiakaskohtaista ohjelmakoodia. Työasemasovelluksesta voidaan kuitenkin kutsua ulkoista asiakaskohtaista sovellusta tarvittaessa.

3.1 Tiedonsiirtoprotokolla

Tiedonsiirtosovellus perustuu tiedostojen kopiointiin mobiilipäätteen ja työaseman välillä: kaikki tiedonvaihto, tiedonsiirtoprotokolla mukaan lukien, perustuu tiedostojen siirtämiseen. Mobiilipääte on aina tiedonsiirron aloittava osapuoli. Tiedonsiirtoprotokolla toimii yksinkertaistetusti seuraavalla tavalla:

1. Mobiilipäätsovellus luo TransferOrders.xml-tiedoston. Tiedosto sisältää siirrettävien tiedostojen nimet ja siirron suunnan.
2. Työasemasovellus hakee mobiilipäätteestä TransferOrders.xml-tiedoston ja tulkitsee sen sisällön.
3. Työasemasovellus siirtää halutut asiakaskohtaiset tiedostot TransferOrders.xml-tiedoston mukaisesti ja käynnistää ulkoisen sovelluksen tarvittaessa.
4. Työasemasovellus luo Status.xml-tiedoston, joka sisältää tiedon tiedonsiirron onnistumisesta ja siirtää sen mobiilipäätteeseen.

3.2 Työasemasovellus

Työasemasovelluksessa on graafinen käyttöliittymä, jonka avulla voidaan muokata sovellukseen ja tiedonsiirtoon liittyviä asetuksia sekä seurata tiedonsiirron tilaa.

Sovellus käynnistyy automaattisesti silloin, kun mobiilipääte asetetaan telakkaan ja yhteys on muodostunut. Tiedonsiirron tila näkyy käyttäjälle reaaliaikaisesti. Tiedonsiirron päätyttyä sovellus sulkeutuu automaattisesti.

3.3 Tiedonsiirrossa käytettävät teknologiat

Tiedostojen kopiointi mobiilipäätteen ja työaseman välillä on toteutettu RAPI-ohjelmointirajapinnan avulla. RAPI:n avulla on mahdollista päästä käsiksi Windows CE -käyttöjärjestelmällä varustetun mobiilipäätteen tiedostojärjestelmään työasemalta käsin. Tuettuja toimintoja ovat esimerkiksi tiedoston luominen, poistaminen ja siirtäminen. RAPI:n käyttö edellyttää ActiveSync- tai Mobile Device Center -sovelluksen asentamista työasemalle. [2.]

ActiveSync ja Mobile Device Center ovat Microsoftin kehittämiä sovelluksia, joilla voidaan vaihtaa tietoa mobiilipäätteen ja työaseman välillä. ActiveSync on tarkoitettu Windows XP -käyttöjärjestelmälle, Mobile Device Center puolestaan Windows Vista - ja Windows 7 -käyttöjärjestelmille. Molempien sovellusten ominaisuuksiin kuuluvat esim. sähköpostien ja kalenterimerkintöjen automaattinen synkronointi. Ne myös tarjoavat pääsyn mobiililaitteen tiedostojärjestelmään Windowsin resurssienhallinnan avulla. [3; 4.]

4 Keskeisimmät ongelmat ja puutteet

Tiedonsiirtosovellukseen liittyviä ongelmia kartoitettiin haastatteleamalla sovelluksen suunnitteluun, toteutukseen ja käyttöönottoihin osallistuneita sovellusasiantuntijoita. Haastattelut olivat luonteeltaan vapaamuotoisia keskustelutilanteita, joissa sovellusasiantuntijat kertoivat kokemistaan ongelmista ja puutteista. On huomattava, että näin saadut tutkimustulokset ovat subjektiivisia ja voivat olla osin keskenään ristiriitaisia.

Keskeisimmät ongelmat kirjattiin saadun haastatteluaineiston perusteella. Ongelmat ryhmiteltiin seuraaviin kategorioihin:

- sovelluksen asentaminen ja konfigurointi
- tiedonsiirron toimintavarmuus
- ohjelmointirajapinnan käytettävyys
- työasemasovelluksen käytettävyys
- versionhallinta.

4.1 Sovelluksen asentaminen ja konfigurointi

Asiakaskohtaisen ratkaisun käyttöönotossa tiedonsiirtosovellus asennetaan asiakkaan työasemalle ja tiedonsiirtosovellukseen tehdään tarvittavat asiakaskohtaiset asetukset. Asiakaskohtaisissa asetuksissa voidaan määrittää mm. käynnistettävä ulkoinen sovellus ja tiedonsiirtokansioiden sijainnit.

Asiakkailla on vaihtelevia käyttöjärjestelmän hallintaan liittyviä asetuksia, kuten kansioiden luku- ja kirjoitusoikeudet sekä sovellusten suoritusoikeudet. Nämä oikeudet voivat vaihdella työasema- ja käyttäjäkohtaisesti. Lisäksi Windows XP - käyttöjärjestelmän hallintaoikeudet poikkeavat Windows Vistan ja Windows 7:n hallintaoikeuksista. Hallintaan liittyvät oikeudet on otettava huomioon sovellusta asennettaessa, koska riittävien oikeuksien puuttuminen voi aiheuttaa sovelluksen toimimattomuuden. Tämän vuoksi standardoitua toimintatapaa työasemasovelluksen asentamiseksi ei voida käyttää.

Työasemasovelluksen asetukset voidaan määrittää käyttöliittymän tai erillisen xml-muotoisen asetustiedoston avulla. On havaittu, että joissain tapauksissa asetusten muuttaminen käyttöliittymän avulla ei ole ollut mahdollista – annetut asetukset eivät ole tallentuneet. Tällöin sovellus on jouduttu asentamaan uudelleen, jonka jälkeen asetukset on jouduttu tekemään ennen sovelluksen käynnistämistä xml-tiedoston avulla. Tämä ongelma liittyy osittain versionhallinnan puutteisiin (luku 4.5).

4.2 Tiedonsiirron toimintavarmuus

Joissakin tapauksissa, joissa mobiilipäätte irrotetaan telakasta tiedonsiirron aikana, tiedostot siirtyvät työasemalle onnistuneesti, mutta mobiilipäätte ei ehdi saada tietoa tiedonsiirron onnistumisesta. Tällöin on vaarana, että sama tieto välitetään kahteen kertaan.

4.3 Työasemasovelluksen käytettävyys

Työasemasovellus kertoo tiedonsiirron tilan reaaliaikaisesti, mitä voidaan pitää Nielsenin ensimmäisen heurestiikan avulla arvioitaessa hyvänä ominaisuutena. Tiedonsiirtoa suorittava henkilö ei kuitenkaan välttämättä tarvitse tätä annettua tietoa, koska sama tieto on yleensä saatavilla myös mobiilipäätteen sovelluksesta. Tämä on vastoin Nielsenin kahdeksatta heurestiikkaa, jonka mukaan pitäisi välttää käyttäjälle välitettävää tarpeetonta informaatiota. [5, s. 449.]

Tiedonsiirtoa suorittavalla henkilöllä on mahdollisuus tehdä muutoksia sovelluksen asetuksiin erillisen asetuskunan kautta. Asetuskuna avautuu pääikkunassa olevaa painiketta painamalla. Yleensä tiedonsiirtoa suorittavalla henkilöllä ei ole tarvetta muuttaa näitä asetuksia, koska asetukset tehdään jo asennusvaiheessa. Myös tämä sotii jossain määrin Nielsenin kahdeksatta heurestiikkaa vastaan, koska asetuskunan tarjoaminen tiedonsiirtoa suorittavalle henkilölle vähentää käyttöliittymän yksinkertaisuutta.

Mobile Device Center -sovellus käynnistyy, kun mobiilipäätte yhdistetään työasemaan. Näytölle avautuva käyttöliittymä, jolla mobiililaitteen käyttäjä voi halutessa synkronoida päätteen ja työaseman välisiä tietoja, on tarpeeton tiedonsiirtosovelluksen kannalta.

Tämä lisää edelleen turhaa tietoa, jolle tiedonsiirtoa suorittava henkilö altistuu ja on niin ikään vastoin Nielsenin kahdeksatta heurestiikkaa.

4.4 Tiedonsiirtorajapinnan käytettävyys

Työasema välittää mobiilipäätteelle tiedon tiedonsiirron onnistumisesta. Kun siirretään monta tiedostoa kerralla, saadaan vain tieto siitä, onko kaikkien tiedostojen siirto onnistunut. Yhdenkin tiedoston epäonnistunut siirto aiheuttaa kaikkien tiedostojen uudelleenlähettämisen.

Tiedonsiirtorajapinta sisältää lukuisia metodeja, joita käyttämällä tiedonsiirto voidaan toteuttaa. Ohjelmoijan on käytettävä useita näistä metodeista ohjelmoidakseen tiedoston siirron. Sovellussuunnittelijoilta saatujen arvioiden mukaan rajapinnan käyttäminen on tarpeettoman hankalaa.

Ohjelmointirajapinnan käytettävyyttä voidaan arvioida tietyissä rajoissa käyttöliittymän käytettävyyden arviointiin tarkoitetuilla menetelmillä. Esimerkiksi Schneidermanin ja Plaisantin mukaan [6, s. 14] käyttöliittymän suunnittelussa ja käytettävyyden arvioinnissa voidaan käyttää mittareina mm. aikaa, joka kuluu

- tehtävän oppimiseen
- tehtävän suorittamiseen
- opitun tehtävän unohtamiseen.

Voidaan päätellä, että tiedonsiirtoon vaadittavien metodien lukumäärä on suoraan verrannollinen tiedonsiirron opetteluun, toteuttamiseen ja toteuttamisen unohtamiseen kuluvaan aikaan. Tiedonsiirtorajapinta on siten käytettävyydeltään huonompi kuin rajapinta, joka tarjoaa samat toiminnot suppeammalla metodivalikoimalla.

4.5 Versionhallinta

Tiedonsiirtosovellusta on sen elinkaaren aikana kehitetty vastaamaan erilaisia asiakaskohtaisia tarpeita. Sovelluksen muuttuneita ominaisuuksia ei ole dokumentoitu, ja uusi versio on yleensä korvannut vanhan version sovelluksen

versionhallintajärjestelmässä. Sovelluksen versioiden yksilöintiin ei ole käytetty versionumerointia.

Puutteellinen muutosten hallinta on johtanut tilanteeseen, jossa tiedonsiirtosovelluksen toimivuudesta asiakkaalla ei ole täyttä varmuutta. Toimivuus on varmistettava asiakaskohtaisesti. Ongelmien ilmetessä on usein kokeiltava jotain muuta versiota sovelluksesta.

4.6 Johtopäätökset

Tiedonsiirtosovelluksen ongelmat ja puutteet ovat suurelta osin subjektiivisia ja liittyvät sekä tiedonsiirtotoiminnallisuuden ohjelmointiin käytettävän tiedonsiirtorajapinnan että työasemasovelluksen käytettävyyteen.

Haastatellut sovellusasiantuntijat pitävät käytettäviä teknologioita (ActiveSync, RAPI) epäluotettavina ja epäkäytännöllisinä. Näiden tekijöiden järjestelmällinen tai johonkin mitattavaan ominaisuuteen perustuva arviointi osoittautui vaikeaksi – tosin halu korvata nämä teknologiat jollain vaihtoehtoisella teknologialla on lähes yksimielinen. On myös todennäköistä, että tiedonsiirtovarmuuteen liittyvä ongelma O4 johtuu RAPI-rajapinnan puutteista.

Versionhallintaan ja dokumentointiin liittyvät puutteet ovat selkeästi todettavissa. Ohjelmistotuotteen elinkaaren aikana on huolehdittava kattavasta dokumentoinnin tuottamisesta ja järjestelmällisestä versionhallinnasta. Vain näin voidaan varmistaa riittävä jäljitettävyyden ohjelmistotuotteen ominaisuuksien ja eri versioiden välillä.

5 Ehdotukset ongelmien ratkaisemiseksi

Havaittujen ongelmien ja puutteiden korjaamiseksi tehtiin neljä ehdotusta: työasemasovelluksen käyttöliittymän uudelleensuunnittelu, tiedonsiirtorajapinnan uudelleensuunnittelu, tiedonsiirto ilman ActiveSynciä sekä versionhallinta ja riittävä dokumentointi. Näistä jokainen voidaan toteuttaa erikseen, toisistaan riippumatta.

5.1 Työasemasovelluksen käyttöliittymän uudelleensuunnittelu

Työasemasovelluksen käyttöliittymän käytettävyysongelmat voidaan korjata käyttöliittymän osittaisella uudelleensuunnittelulla. Poistamalla käyttöliittymästä kaikki tiedonsiirtoa suorittavan henkilön näkökulmasta tarpeeton informaatio voidaan käyttöliittymän ulkoasua selkeyttää huomattavasti.

On huomattava, että kaikki tiedonsiirtoon liittyvä informaatio on mahdollista välittää mobiilipäätesovelluksen käyttöliittymässä. Työasemasovellus ei siis välttämättä tarvitse käyttöliittymää. Asetusten tekemiseen on tosin oltava mahdollisuus, mutta sen ei tarvitse olla koskaan näkyvissä tiedonsiirtoa suorittavalle henkilölle.

5.2 Tiedonsiirtorajapinnan uudelleensuunnittelu

Ohjelmointirajapinta on mahdollista suunnitella esim. Schneidermanin ja Plaisantin [12, s. 14] käytettävyysohjeita noudattamalla. Yksinkertaistamalla tiedonsiirtorajapintaa siten, että se silti täyttää kaikki tiedonsiirron asettamat vaatimukset, voidaan sovellussuunnittelijan työtä helpottaa merkittävästi.

Tiedonsiirtorajapintaa ei ole välttämättä suunniteltava alusta asti uudelleen. Siihen voidaan suunnitella uusi sovellussuunnittelijalle näkyvä rajapinta, jonka toiminnallisuus perustuu jo olemassa olevaan tiedonsiirtorajapintaan.

5.3 Tiedonsiirtoteknologian vaihtaminen

ActiveSync-protokollaan ja RAPI-rajapintaan perustuva tiedonsiirtosovellus on riippuvainen ActiveSync- tai Mobile Device Center -sovelluksesta. Käyttämällä tiedonsiirtoon jotakin muuta tiedonsiirtoteknologiaa, voitaisiin välttää ainakin yksi

työasemasovelluksen käytettävyyteen (ks. luku 4.3) liittyvistä ongelmista. Uusi tiedonsiirtoteknologia on mahdollista ottaa käyttöön siten, että olemassa oleva julkinen tiedonsiirtorajapinta pysyy samana.

Seuraavaksi esitetään ActiveSync-protokollasta riippumattomat tiedonsiirtoteknologiat, jotka ovat mobiilipäätteiden laitteiston perusteella ainakin teoreettisesti mahdollisia vaihtoehtoja.

USB-protokollaan perustuva tiedonsiirto

Nykyisessä tiedonsiirtosovelluksessa tiedonsiirto tapahtuu USB-väylää pitkin, mutta on riippuvainen ActiveSync-protokollasta. USB mahdollistaa kuitenkin myös lukuisten muiden tiedonsiirtoprotokollien käytön. Näitä ovat esim. TCP/IP-tietoverkkoprotokolla, RS-232-sarjaväyläprotokolla ja massamuistiprotokolla.

Jos ActiveSynciin perustuva tiedonsiirto toteutettaisiin jollakin vaihtoehtoisella USB:hen perustuvalla teknologialla, olisi tiedonsiirtosovellusta mahdollista käyttää hyvin samaan tapaan kuin aiemmin – tiedonsiirtoarkkitehtuuri voidaan pitää samana.

Langaton tiedonsiirto

Tiedonsiirto on mahdollista toteuttaa langattomasti WLAN-, GPRS- ja Bluetooth-tiedonsiirtoväylien avulla. Näin vältetään ActiveSync-protokollaan liittyvät ongelmat ja lisätään toimintasädetä, jossa mobiilipäättesovelluksen tiedonsiirto voidaan suorittaa. USB-yhteyteen perustuvan tiedonsiirron korvaaminen langattomalla tiedonsiirrolla vaatii muutoksen tiedonsiirtosovelluksessa käytettävään tiedonsiirtoarkkitehtuuriin.

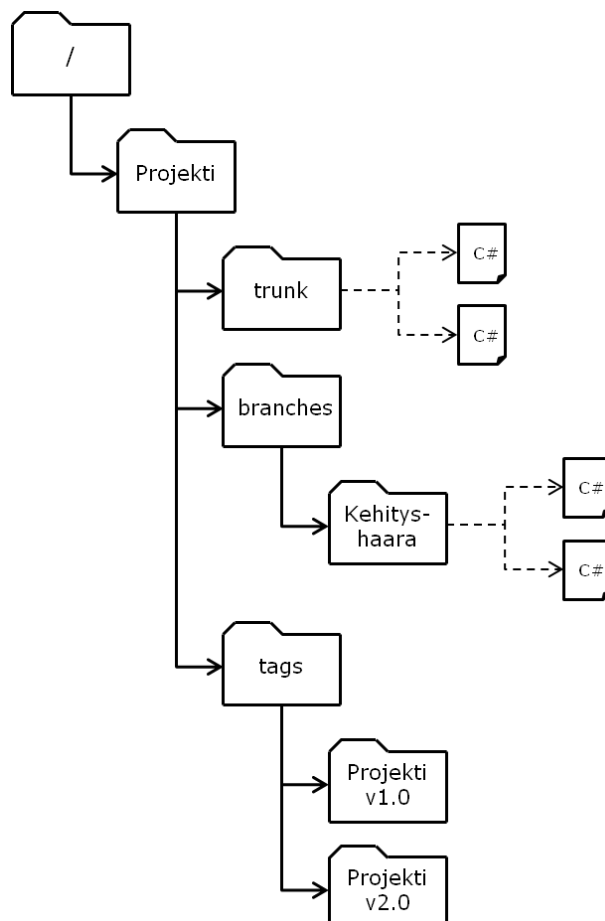
Langaton tiedonsiirto on käytössä nykyisissäkin tiedonsiirtoratkaisuissa, jos tiedonvälitys vaatii reaaliaikaisuutta (esim. mobiilipäätteessä on oltava ajan tasalla oleva tuoterekisteri). Langaton tiedonsiirto ei kuitenkaan sovellu kaikkiin toimintaympäristöihin. Esimerkiksi asiakkaan tietoturvallisuusvaatimukset saattavat kieltää salaiseksi julkistetun aineiston välittämisen langattomasti, koska tällä tavoin välitetty tieto on teoriassa mahdollista kaapata ja analysoida.

5.4 Versionhallinta ja riittävä dokumentointi

Sovellukseen sen elinkaaren aikana tehdyt muutokset vaativat järjestelmällistä muutosten hallintaa. Yrityksen käytössä olevalla Subversion-versionhallintasovelluksella on mahdollista hallita ja jäljittää sovelluksen lähdekoodiin tehtäviä muutoksia.

Subversion-sovellus toimii tietovarastona, joka sisältää kansioita ja tiedostoja. Tieto kansioihin ja tiedostoihin tehdyistä muutoksista tallennetaan, mikä mahdollistaa palaamisen mihin tahansa aiempaan tilanteeseen.

Suosittelava tapa hallita sovellukseen tehtäviä muutoksia on käyttää erityistä kansiorakennetta, jossa sovelluksen pääkehityslinjan lähdekoodit on erotettu sovelluksen kehityshaaran lähdekoodeista. Kuva 3 esittää Projekti-nimiselle sovellukselle luotua kansiorakennetta, joka on suosituksen mukainen.



Kuva 3. Suositeltu kansiorakenne

Suosittelussa kansiorakenteessa pääkehityslinjan lähdekoodit sijaitsevat trunk-kansiossa. Mikäli sovellukseen halutaan tehdä merkittäviä muutoksia, tulee trunk-kansion sisällöstä luoda kopio branches-kansion alle. Branches-kansio voi sisältää useita rinnakkain kehitettäviä kehityshaaroja, jotka voidaan yhdistää muutostyön valmistuttua takaisin päähaaraan. Kehityshaara voidaan myös eriyttää kokonaan omaksi sovellukseksi, jolloin sille luodaan oma kansiorakenne hakemiston juureen.

Tags-kansioon voidaan kopioida erityiset pääkehityslinjan tilanteet, jotka halutaan tallentaa. Tämä on syytä tehdä aina, kun sovelluksesta julkaistaan uusi versio. Tällä tavoin kaikki sovelluksen versiot ovat helposti löydettävissä ilman, että niitä joudutaan etsimään versionhallintasovelluksen muutoshistoriasta. Julkaistavat versiot tulee yksilöidä versionumeroinnilla, jotta eri versiot voidaan tunnistaa.

Jokaisesta julkaistavasta versiosta tulee luoda dokumentaatio, jossa kuvataan sovelluksen ominaisuudet, puutteet ja täydellinen historia sovellukseen siihen asti tehdyistä muutoksista. Dokumentaation avulla voidaan varmistaa sovelluksen elinkaaren aikana tehtyjen muutosten jäljitettävyyttä.

5.5 Johtopäätökset

Selvityksen perusteella päädyttiin tutkimaan ActiveSync-protokollan korvaavaa teknologiaa tiedonsiirron toteuttamiseksi. Koska langatonta tiedonsiirtoa ei voida pitää ratkaisuna, joka soveltuu kaikkiin toimintaympäristöihin, valittiin USB-protokollaan perustuva tiedonsiirto ensisijaiseksi tutkimuksen kohteeksi.

Mikäli todetaan, että ActiveSync-riippumaton, USB-protokollaan perustuva tiedonsiirto on mahdollista toteuttaa, tullaan muut ongelmien ratkaisemiseksi ehdotetut toimenpiteet huomioimaan uutta tiedonsiirtosovellusta toteutettaessa.

6 USB:hen perustuva tiedonsiirto

USB (Universal Serial Bus) on tietokoneen ja siihen kytkettävän ulkoisen laitteen välinen tiedonsiirtoarkkitehtuuri. USB-spesifikaatio määrittelee mm. käytettävät liitännät, tiedonsiirtoprotokollan ja laitteilta vaadittavat sähköiset ominaisuudet. [7, s. 1.]

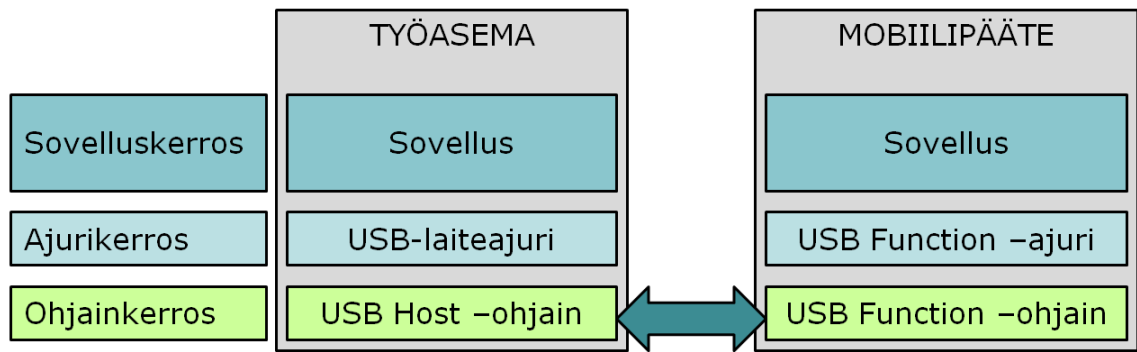
USB määrittelee useita yleisiä laiteluokkia, joita ovat esim. audio-, kommunikaatio-, massamuisti- ja tulostinlaiteluokat. Tietokoneen käyttöjärjestelmä voi tarjota ajurin yleiseen laiteluokkaan kuuluvalle laitteelle, jolloin erillistä laitekohtaista ajuria ei tarvita. Tämä mahdollistaa esim. USB-muistitikkujen kytkemisen mihin tahansa käyttöjärjestelmään ilman käyttäjältä vaadittuja toimenpiteitä. Myös laitevalmistajat selviävät vähemmällä työmäärällä, koska ajurien suunnitteluun ja toteuttamiseen ei tarvitse kiinnittää resursseja. Laitevalmistajat joutuvat kehittämään omat ajurinsa vain siinä tapauksessa, että laitteelta vaaditut toiminnallisuudet eivät sovi mihinkään yleiseen laiteluokkaan. [8, s. 7, 178.]

Tässä luvussa

- kuvataan USB:n arkkitehtuuri mobiililaitteen ja työaseman näkökulmasta.
- arvioidaan yrityksen käytössä olevien mobiilipäätteiden sisältämien USB Function -ajureiden soveltuvuus tiedonsiirtosovelluksen tarpeisiin.
- kuvataan, kuinka USB-tiedonsiirto voidaan toteuttaa jonkin muun kuin mobiilipäätteen sisältämän laiteajurin avulla.

6.1 USB:n arkkitehtuuri

USB:n arkkitehtuuri missä tahansa käyttöjärjestelmässä voidaan yksinkertaistaen jakaa kolmeen kerrokseen: sovellus- ja ajuri- ja ohjainkerrokseen. Kuva 3 esittää USB:n arkkitehtuuria. Työasema on USB-protokollan mukaisesti USB-isäntä, ja mobiilipääte on USB-laite.



Kuva 4. USB:n arkkitehtuuri

Ohjainkerros toteuttaa USB:n matalan tason tiedonsiirtoprotokollan. Se on käyttöjärjestelmästä riippumaton osa, joka on yleensä toteutettuna omana integroituna piirinään. Ajurikerros on osa käyttöjärjestelmää, ja se keskustelee suoraan ohjainkerroksen kanssa. Ajurikerros on vastuussa tietyille laitteille tyypillisestä korkean tason tiedonsiirtoprotokollan toteuttamisesta. Sovelluskerros on tiedon pääteipiste: se voi sekä tuottaa tietoa että kuluttaa sitä.

Ajurikerros on merkittävässä roolissa USB-laitteen toiminnallisuuden kannalta. USB-laitteen ajurikerros sisältää laitetyypille ominaisen USB-deskriptorin, jonka avulla USB-isäntä osaa ladata tarvittavan ajurin käyttöjärjestelmänsä ajurikerrokseen. Mikäli deskriptoriin sopivaa laiteajuria ei ole saatavilla, USB-isännän käyttöjärjestelmä ei voi käyttää kytkettyä laitetta.

6.2 Mobiilipäätteiden sisältämät USB Function -ajurit

Yrityksen käytössä olevien mobiilipäätteiden käyttöjärjestelmien mukana toimitettavista USB Function -ajureista tehtiin selvitys, joka on esitetty liitteessä 1. Selvitys on vain työn tilaajan käyttöön.

Selvityksen perusteella voidaan päätellä, että mobiilipäätteissä on yleiseen laiteluokkaan kuuluvia USB Function -ajureita ja niille on valmis tuki työasemien käyttöjärjestelmissä. Erästä USB Function -ajuria ja sen tarjoamia toiminnallisuuksia tutkittiin tarkemmin, jotta voitiin selvittää soveltuvuus tiedonsiirtosovelluksen tarpeisiin. Tutkimus osoitti, että ajurin TCP/IP-protokollaan perustuva tietoliikennetoiminnallisuus

ei ole oletusarvoisesti käytettävissä, mutta se voidaan tarvittaessa ottaa käyttöön ohjelmallisesti.

6.3 Muut vaihtoehdot USB-tiedonsiirron toteuttamiseksi

USB-tiedonsiirto voidaan toteuttaa myös muiden kuin mobiilipäätteen mukana toimitettavien ajurien avulla. Tällöin vaihtoehtoina on toteuttaa ajurit itse tai käyttää kolmannen osapuolen ajureita.

USB Function -ajuri on mahdollista toteuttaa Microsoftin Platform Builder -ohjelmointiympäristössä [9]. Toteutettava ajuri voidaan suunnitella siten, että se noudattaa jonkin yleiselle laiteluokalle asetettuja vaatimuksia. Laiteluokkaa valittaessa on syytä varmistaa, että työaseman käyttöjärjestelmässä on tuki tälle laiteluokalle.

Kaupallisia, kolmannen osapuolen USB Function -ajuriratkaisuja on saatavilla rajoitetusti. Microsoftin ylläpitämä lista tällaisista ajureista sisältää mm. rinnakkaisportti-, audio- ja HID-ajurit [10]. Näiden ajurien soveltuvuutta käytössä olevien mobiilipäätteiden tiedonsiirtoon ei kuitenkaan voida arvioida ilman tarkempaa perehtymistä.

6.4 Johtopäätökset

Yrityksen käytössä olevissa mobiilipäätteissä on USB Function -ajureita, jotka soveltuvat tiedonsiirtosovelluksen tarpeisiin ja joita käyttämällä voidaan välttää ActiveSync-protokollaan ja RAPI-rajapintaan perustuva tiedonsiirto. Koska mobiilipäätteen ajureita käyttämällä vältetään omien USB Function -ajureiden suunnittelutarpeelta, ne valittiin ensisijaisiksi vaihtoehdoiksi uutta sovellusta suunniteltaessa.

Selvitysten perusteella tehtiin päätös TCP/IP-protokollaan perustuvan uuden tiedonsiirtosovelluksen suunnittelusta.

7 TCP/IP-protokolla perustuva tiedonsiirtosovellus

Tiedonsiirtosovellusta käytetään mobiilipäätteen ja työaseman välillä tapahtuvan tiedonsiirron toteuttamisessa. TCP/IP-protokolla perustuvan uuden tiedonsiirtosovelluksen on tarkoitus korvata käytössä oleva, ActiveSync-protokolla ja RAPI-rajapintaan perustuva tiedonsiirtosovellus.

Uudesta tiedonsiirtosovelluksesta tehtiin alustava suunnitelma, joka perustuu osittain vanhan tiedonsiirtosovelluksen toiminnallisiin vaatimuksiin. Suunnitelman tekninen dokumentaatio sisältää sovellusarkkitehtuurin, rajapintakuvaukset ja paketteihin perustuvan tiedonsiirtoprotokollan. Tekninen dokumentaatio on liitteessä 2, joka on vain työn tilaajan käyttöön.

Seuraavissa luvuissa kuvataan tiedonsiirtosovelluksen toimintojen perustana olevat asiakasvaatimukset ja sovelluksen toimintaympäristö.

7.1 Vaatimukset

Uudella tiedonsiirtosovelluksella tulee olla mahdollista suorittaa vähintään ne toiminnot, joita käytössä olevalla sovelluksella voidaan toteuttaa. Näitä toimintoja ovat tiedoston kaksisuuntainen siirtäminen, mobiilipäätteen kellon synkronointi työaseman kellonaikaan ja ulkoisen, asiakaskohtaisen ohjelmakoodin käynnistäminen työasemalla.

TCP/IP-protokollan toteuttava, .NET-ohjelmointirajapinnan tarjoama Socket-rajapinta mahdollistaa tiedonsiirron tietovirtana, jonka avulla on mahdollista siirtää muutakin kuin tiedoston muodossa olevaa tietoa (vrt. luku 3.1). Uudessa tiedonsiirtosovelluksessa tulee olla mahdollisuus siirtää tietoa sisältävien luokkien instansseja. Siirto voidaan toteuttaa mm. XML- ja binääriserialisoinnin avulla. Instanssien siirtäminen mahdollistaa esim. suorituskäytävien vaativien prosessien suorittamisen mobiilipäätteen sijaan työasemalla.

Käytettävä laitteisto, ohjelmointiympäristö ja tietoliikenneprotokolla asettavat tiedonsiirtosovellukselle joukon ei-toiminnallisia vaatimuksia. Toiminnalliset ja ei-toiminnalliset vaatimukset on esitetty taulukossa 1.

Taulukko 1. Toiminnalliset ja ei-toiminnalliset vaatimukset

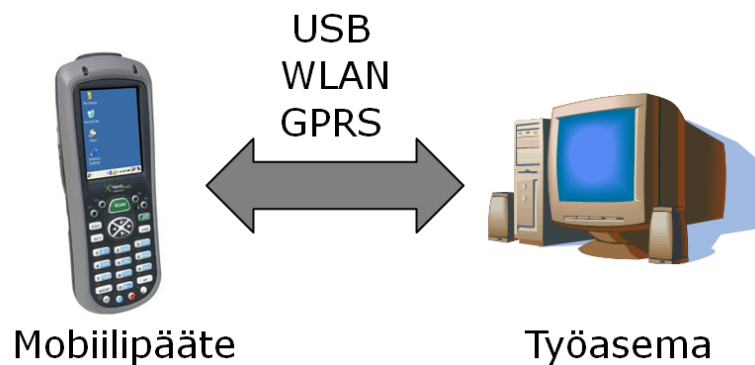
Toiminnalliset vaatimukset	
ID	Kuvaus
TV1	Tiedonsiirto on kaksisuuntaista.
TV2	Siirrettävä tieto voidaan siirtää tiedostona.
TV3	Siirrettävä tieto voidaan siirtää objektina.
TV4	Siirrettävä tiedosto voidaan nimetä uudelleen vastaanottavassa päässä.
TV5	Siirrettävän tiedoston lähde- ja kohdekansio työasemalla määritetään erillisessä asetustiedostossa, joka sijaitsee työasemalla.
TV6	Mobiilipäätteen aika voidaan tahdistaa työaseman aikaa vastaavaksi.
TV7	Mobiilipäätteeltä voidaan käynnistää työasemalla sijaitsevan ohjelmakoodin suoritus.
TV8	Tiedonsiirtoon liittyvät tapahtumat tallennetaan lokitiedostoon sekä mobiilipäätteessä että työasemalla.
TV9	Tiedonsiirto voidaan keskeyttää.
TV10	Tiedonsiirron tilaa voidaan seurata mobiilipäätteen käyttöliittymässä.
TV11	Työasemasovellukseen voidaan ottaa samanaikainen yhteys useasta mobiilipäättesovelluksesta.
TV12	Tiedoston siirron onnistuminen voidaan varmistaa tiivistealgoritmin avulla.
Ei-toiminnalliset vaatimukset	
ID	Kuvaus
EV1	Tiedonsiirto toteutetaan TCP/IP-yhteyden avulla.
EV2	Tiedonsiirtoon voidaan käyttää USB-väylää.
EV3	Tiedonsiirtoon voidaan käyttää WLAN-yhteyttä.
EV4	Tiedonsiirtoon voidaan käyttää GPRS-yhteyttä.
EV5	Mobiilipäättesovellus toimii WM6-käyttöjärjestelmässä.
EV6	Mobiilipäättesovellus toimii WM6.5-käyttöjärjestelmässä.
EV7	Mobiilipäättesovellus toimii WEH6.5-käyttöjärjestelmässä.
EV8	Työasemasovellus toimii Windows XP -käyttöjärjestelmässä.
EV9	Työasemasovellus toimii Windows 7 -käyttöjärjestelmässä.
EV10	Työasemasovellus toimii Windows 8 -käyttöjärjestelmässä.
EV11	Työasemasovellus toimii Linux-käyttöjärjestelmissä.
EV12	Mobiilipäättesovelluksessa on tiedonsiirron tilaa monitoroiva graafinen käyttöliittymä.

Uusi tiedonsiirtosovellus toteutetaan useassa vaiheessa. Ensimmäisessä vaiheessa toteutetaan vain tiedostonsiirtotoiminnallisuus. Sovellus on kuitenkin suunniteltava

siten, että uusien ominaisuuksien lisääminen joustavasti myöhemmässä vaiheessa on helppoa eikä vaadi toteutettujen ominaisuuksien uudelleensuunnittelua. Ensimmäisessä vaiheessa toteutetaan toiminnalliset vaatimukset TV1, TV2, TV4, TV9, TV11 ja TV12 sekä ei-toiminnalliset vaatimukset EV1 – EV10.

7.2 Toimintaympäristö ja käytettävät teknologiat

Kuten käytössä oleva, myös toteutettava sovellus mahdollistaa tiedonsiirron mobiilipäätteen ja työaseman välillä USB-väylässä (kuva 8).



Kuva 5. Tiedonsiirtoväylät

TCP/IP-protokollan käyttäminen mahdollistaa tiedonsiirron myös langattomien WLAN- ja GRPS-verkkoyhteyksien välityksellä, mikä laajentaa tiedonsiirtosovelluksen käyttömahdollisuuksia. Sama asiakaskohtainen sovellus voidaan mobiilipäätteessä olevia verkkoasetuksia muuttamalla sopeuttaa erilaisten toimintaympäristöjen asettamiin vaatimuksiin.

Tiedonsiirtosovellus ohjelmoidaan C#-kielellä ja Microsoftin .NET-sovelluskehysten tarjoamien sovelluskirjastojen avulla. Mobiilipäätteen käyttöjärjestelmä vaatii .NET Compact Framework -sovelluskehysten käytön, jota voidaan käyttää osin myös työasemasovelluksen ohjelmoinnissa. Näin samoja sovelluskomponentteja voidaan käyttää sekä mobiilipäätteessä että työasemassa.

8 Yhteenveto

Insinööriyössä kartoitettiin käytössä olevaan tiedonsiirtosovellukseen liittyviä ongelmia. Ongelmien kartoitus suoritettiin yrityksen sovelluskehittäjiä haastatteleamalla, minkä vuoksi saadut tulokset perustuvat lähinnä subjektiivisiin arvioihin. Kirjatut ongelmat saattavat tästä syystä painottua erityisesti sovelluksen käytettävyysoongelmiin.

Voitiin havaita, että vaikka käytettävyyden arviointiin käytetyt heurestiset menetelmät on suunniteltu käyttöliittymien arviointiin, ne soveltuvat hyvin myös ohjelmointirajapintojen arviointiin. Voidaan ajatella, että käyttöliittymä on käyttäjän rajapinta sovellukseen, ohjelmointirajapinta puolestaan sovelluskehittäjän käyttöliittymä sovelluskirjastoon. Rajapintojen käytettävyyttä voidaan parantaa huomioimalla esim. Nielsenin ja Schneidermanin heurestiset arviointimenetelmät jo suunnitteluvaiheessa.

Tiedonsiirtosovelluksen versionhallinnassa ja dokumentoinnissa havaittiin selkeät puutteet. Käytäntöjä muuttamalla voidaan parantaa sovelluksen toiminnallisuuksien ja puutteiden jäljitettävyyttä sovelluksen elinkaaren aikana.

Tiedonsiirtoteknologiaa vaihtamalla voidaan osittain parantaa tiedonsiirtosovelluksen käytettävyyttä. Insinööriyössä tehtiin selvitys käytettävissä olevista tiedonsiirtoteknologioista sekä niiden käyttöönottamisesta. Selvitys perustuu suhteellisen vaikeasti löydettävissä oleviin internetlähteisiin sekä lähteiden oikeellisuutta tukeviin mobiilipäätteillä tehtyihin testeihin. Mahdollisesta kilpailuedusta johtuen selvitys luovutetaan vain yrityksen omaan käyttöön ja eikä se sisälly työn kirjalliseen osioon.

Uusi tiedonsiirtoteknologia mahdollistaa TCP/IP-protokollaan perustuvan tiedonsiirtotavan, jolla voidaan mm. laajentaa tiedonsiirtosovelluksen toimintaympäristöä.

Uudesta TCP/IP-protokollaan perustuvasta tiedonsiirtosovelluksesta tehtiin alustava suunnitelma, joka sisältää sovellusarkkitehtuurin, rajapintakuvaukset ja paketteihin perustuvan tiedonsiirtoprotokollan. Tulevaisuuden kehitystarpeet otettiin huomioon, jotta lisätoiminnallisuutta voidaan tarvittaessa toteuttaa vaiheittain.

Hyödyt, jotka selvitysten perusteella tehtyjen toimenpiteiden avulla saavutetaan, voidaan arvioida perusteellisesti vasta toimenpiteiden toteuttamisen jälkeen.

Lähteet

- 1 Pavlov, Stanislav. 2008. Windows Embedded CE 6.0 Fundamentals. Redmond WA, USA: Microsoft Press.
- 2 Microsoft. 2010. Remote API (RAPI). Verkkodokumentti. <msdn.microsoft.com/en-us/library/aa920177.aspx>. 8.4.2010. Luettu 21.4.2012.
- 3 Microsoft. 2010. Data Synchronization With ActiveSync. Verkkodokumentti. <msdn.microsoft.com/en-us/library/aa913369.aspx>. 8.4.2010. Luettu 21.4.2012.
- 4 Microsoft. 2010. Description of Windows Mobile Device Center. Verkkodokumentti. <support.microsoft.com/kb/931937>. 29.11.2010. Luettu 21.4.2012.
- 5 Lauesen, Soren. 2005. User Interface Design: A Software Engineering Perspective. Essex, Englanti: Pearson Education Limited.
- 6 Schneiderman, Ben. 2009. Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition). Boston MA, Yhdysvallat: Addison-Wesley.
- 7 USB Implementers Forum. 2000. Universal Serial Bus Specification, Revision 2.0.
- 8 Axelson, Jan. 2005. USB Complete: Everything You Need To Know to Develop USB Peripherals, Third Edition. Madison WI, USA: Lakeview Research LLC.
- 9 Microsoft. Driver Development Tools. Verkkodokumentti. <msdn.microsoft.com/en-us/library/aa447515.aspx>. Luettu 21.4.2012.
- 10 Microsoft. Windows Embedded Compact Drivers. Verkkodokumentti. <www.microsoft.com/windowseembedded/en-us/downloads/windows-embedded-driver-chooser.aspx>. Luettu 21.4.2012.