Sytyke ry:n jäsenlehti
Nro 4/2011

OHJELMISTOKEHITYKSEN ASIANTUNTIJA

SYSTEEMITYÖ

s.4 Kilpailu ja yhteistyö mobiiliekosysteemeissä

s.12 Tietämyksen hallintaa ihmisen ehdoin: Front-end intelligence

Sytykkeen kevätseminaari ajankohtaisesta aiheesta to 29.3.2012! Laita jo kalenteriisi! Tarkempi seminaarinaihe tiedotetaan lehdessä, webbisivuilla ja sosiaalisessa mediassa!

LIIKETOIMINTAPROSESSIT SUJUVIKSI

HYÖDYNNÄ HAAGA-HELIAN LAAJA-ALAISTA LIIKETOIMINTA- JA IT-OSAAMISTA JA OHJAUSTA

Mahdollistamme organisaation tarpeista lähtevän toiminnan ja prosessien kuvaamisen, analysoinnin ja kehittämisen.

Palvelutarjontamme

- Selvitystyöt ja kartoitukset
- Mallinnukset ja analysoinnit
- Kehittämistyöpajat
- Kehittämisprojektit
- Koulutukset

http://prolab.haaga-helia.fi

Lisätiedot:
Jarmo Sarkkinen
jarmo.sarkkinen@haaga-helia.fi
050 412 6913

HAAGA-HELIA
ammattikorkeakoulu

TTL Tietotekniikan liitto ry

Hyvä tietotekniikan liiton jäsen!

Näin päivität tietosi

Voit päivittää jäsentietosi verkkosivuillamme www.ttlry.fi. Tietojen päivittämiseen tarvitset käyttäjätunnuksen (= jäsennumerosi, merkitty jäsenlehtiin) ja salasanasi (= postinumerosi). Jos olet muuttanut salasanasi tai kirjautuminen ei muutoin onnistu, voit lähettää tunnusten tarkistuspyynnön osoitteella jasenasiat@ttlry.fi.
Toivomme sinun erityisesti varmistavan, että sähköpostiosoitteesi jäsentiedoissa on oikea. !

Henkilökohtaisempaa palvelua - Sinun eduksesi

Tietotekniikan liitto jäsenyhdistyksineen, osaamisyhteisöineen ja kerhoineen haluaa palvella jäseniään henkilökohtaisemmin ja paremmin, tarjota tietoa juuri Sinua kiinnostavista aiheista. Palvelun parantamiseksi olemme uusineet verkkopalvelumme.

Päivität vain tiedot itseäsi kiinnostavista aiheista ja saat tietoa juuri niistä. Voit päivittää valintasi aina halutessasi. Tietoja ei anneta ulkopuolisille tahoille vaan niitä käytetään ainoastaan TTL:n ja sen piirissä toimivien yhteisöjen tarkoituksiin.

Tietotekniikan liitto ry
Lars Sonckin kaari 12
02600 Espoo
www.ttlry.fi
etunimi.sukunimi@ttlry.fi
jasenasiat@ttlry.fi
p. 020 741 9898
f. 020 741 9889

Tietotekniikan liiton koulutuksia

12.1.-1.3.	Innovation Manager
16.1.-22.3.	Software Process Improvement Manager (SPI Manager)
18.1.-16.2.	Scope Managerin tutkinto
20.1.	COBIT johdanto
26.1.	ITIL johdanto
27.1.	Hiljainen tieto, Innovaatio ja IT
31.1.	Julkisten IT-hankintojen erityispiirteet
2.2.	PRINCE2 johdanto
20.-21.3.	Tietojärjestelmän hankinta

Lisätiedot www.ttlry.fi/koulutus

Julkaisija
Systeemityöyhdistys Sytyke ry
Tietotekniikan liitto ry
Lars Sonckin kaari 12
02600 Espoo
vaihde: 020 741 9898
fax 020 741 9889

Päätoimittaja
Juha Jääskinen
juha.jaaskinen@digia.com
puh. 050 420 0822

Toimitussihteeri
Anne Hakala

Taitto
Speaking Bark Tmi
Katja Tamminen
taitto.kt@speakingbark.com

Toimituskunta 4/2011
Minna Oksanen
Hannu Savolainen
Ismo Penttinen

Lisätietoja lehdestä
www.sytyke.org/st

Tilaukset
Systeemityö-lehti sisältyy Sytykkeen jäsenmaksuun.
Vuositilaus 30 €
Irtonumerot 8 €
min. 50kpl lisätilaus 2€/lehti
(viim. 1kk ennen julkaisupäivää).
Tilaukset yhdistyksen toimistosta.

Kansikuva
Katja Tamminen

2012 Numerot

Nro	Ilmestyy
1. Kokonaiselinkaari	07.03.2012
2. Innovaatiot, mentorointi	02.05.2012
3. "Laivaseminaari"	03.10.2012
4. Yrittäjyys	12.12.2012

Painopaikka
T-Print
Ahokaari 1-3
05460 Hyvinkää
Puh. (019) 475 8500

Painos: 3500 kpl
ISSN 1237-0525
18. vuosikerta, nro. 4

Ilmoitushinnat
Takakansi A4 1200 €
Sisäkannet A4 1000 €
Sisäsivut 1/1 800 €
Sisäsivut 1/2 600 €
Sisäsivut 1/4 400 €

Arvonlisävero 0%
Vakiopaikan vähintään vuodeksi varanneille 20% alennus.

Kirjoituksia ja kuvia saa lainata lehdestä vain toimituksen luvalla. Toimitus on sitoutumaton eikä ota vastuuta kirjoittajien mielipiteistä. Toimitus ei vastaa asiavirheistä.

Pääkirjoitus

Minna Oksanen

Tämä lehti sai ideansa vuosi sitten, kun kysyin kaveriltani, että mitä hän haluaisi lukea lehdestä. Ensimmäinen kommentti oli mobiiliaiheisia asiaartikkeleja.
Samanaikaisesti toisaalla alkoi tapahtua. Nokian myllerrysten aikana termi ekosysteemi on tullut esiin kaikessa mediassa. Mutta mikä on mobiiliekosysteemi?
Tämän lehden tarkoitus on pureutua aiheeseen, ja näkökulmana, koska olemme systeemityöammattilaisia on tietenkin mobiiliekosysteemin systeemityö eli tuttavallisemmin mobiiliekosysteemityö.

Noin kuukausi sitten tulin itse tilanteeseen, että piti valita uusi työpuhelin. Se ei enää ole niin yksinkertaista, kun pitää samalla valita mihin ekosysteemiin haluan kuulua. Se, että puhelimella on päässyt jo useita vuosia internetiin ei enää olekaan in. Vaan nyt puhutaan sovelluksista, widgeteistä ja appseista riippuen vähän ekosysteemistä. Ja mikä mielenkiintoisinta eri ekosysteemien välillä ei voi vaihtaa sovelluksia vaan jokaiseen ekosysteemiin on ne koodattava erikseen. Hetkinen, koodattava sovelluksia – sehän on perinteistä systeemityötä client server ympäristössä, jota me ammattilaiset olemme tehneet onnistuneesti jo viime vuosituhannella. Paluu juurille siis ja vanhat hyvät opit käyttöön oli ensimmäinen ajatukseni. Siksi siis on tärkeätä, että ymmärrämme miten ekosysteemit toimivat ja miten rakennetaan hyviä sovelluksia eli tehdään mobiiliekosysteemityötä.

Kiitos kaikille kirjoittajille ansiokkaista teksteistä. Lehteen ei valittu tiettyä ekosysteemiä vaan on haluttu kuvata artikkeleissa eri ekosysteemien eroja ja yhdenmukaisuuksiakin. Toki kaikissa artikkeleissa ei ole voitu käyttää kaikkia olemassaolevia kombinaatoita, vaan artikkelien viesti on sovellettavissa muissakin ekosysteemeissä.

Hyvää Joulua kaikille!

3 Pääkirjoitus
Minna Oksanen

4 Kilpailu ja yhteistyö mobiiliekosysteemeissä
Kimmo Karhu

7 Yleinen kehitys mobiilimaailmassa
Ismo Penttinen

8 Mobiilit tiedonsiirtoarkkitehtuurit
Hannu Savolainen

12 Tietämyksen hallintaa ihmisen ehdoin: Front-end intelligence
Ilpo Kojo

16 Kuinka aloittaa mobiili-kehitys iOSille?
Ilkka Pirttimaa

20 Kehitystyötä Androidilla
Rolf Sormo

22 Tulevaisuudessa menestyvän softan pitää myös näyttää ja tuntua hyvältä
Mika Olkkola

24 OSY palsta

26 Kolumni: Kuutamolla

Kimmo Karhu

Kirjoittaja toimii tutkijana Aalto yliopiston perustieteiden korkeakoulussa. Väitökseen tähtäävässä tutkimuksessaan hän tutkii tapoja miten digitaalisia ekosysteemejä voitaisiin mallittaa ja analysoida. [kimmo.karhu@aalto.fi]

Kilpailu ja yhteistyö mobiiliekosysteemeissä

Artikkelissa tarkastellaan merkittävimpien älypuhelinalustojen, iPhone, Windows Phone ja Android, ympärille syntyneitä ekosysteemejä ja niissä vaikuttavia toimijoita. Mitä ovat nämä mobiiliekosysteemit, ja mitä eroja niissä on havaittavissa eri toimijoiden välisissä kilpailu- ja yhteistyösuhteissa?

Taustaa

Stephen Elopin otettua komennon Nokiassa hän julisti nettiin vuodetussa muistiossa[1], että ”laitteiden taistelusta on siirrytty ekosysteemien sotaan”. Lisäksi hän määritteli, että ”laitteiden ja sen käyttöjärjestelmän lisäksi ekosysteemin piiriin kuuluvat niin kehittäjät, sovellukset, sähköinen kaupankäynti, mainos- ja hakupalvelut, sosiaaliset sovellukset, paikkatietoon perustuvat palvelut ja monet muut asiat.” Elopin toteamuksessa tiivistyy useampi huomion arvoinen seikka. Ensinnäkin yksittäisten laitteiden ja mallien sijasta mobiililiiketoimintaa on tarkasteltava kokonaisuutena, ekosysteeminä, jossa ovat mukana niin laitteet, palvelut, sovellukset, käyttäjät kuin yhteistyökumppanit. Toiseksi, kehittäjä ei varmasti vahingossa ollut Elopin listan kärjessä. Viimeisen kolmen vuoden aikana olemme älypuhelinten saralla nähneet ennennäkemättömän ilmiön, kun yksittäiset kehittäjät ja yritykset ovat yhdessä kehittäneet iPhonelle yli puoli miljoonaa sovellusta ja laittaneet ne kaupan tai ilmaiseksi jakoon Applen kauppapaikkaan App Storeen. Näitä sovelluksia oli ladattu tämän vuoden heinäkuuhun mennessä käsittämättömät 15 miljardia kertaa[2]. Tämä Applen innovoima ja katalysoima kehitys muutti pelin säännöt täysin. Muun muassa Nokia, joka toi älypuhelimet ensimmäisenä markkinoille, joutui luomaan nahkansa uudelleen, ja kuten nyt tiedämme, päätti liittoutua Microsoftin kanssa. Elopin vertaus sodasta on siinä mielessä osuva, että pienten taistelujen sijasta tätä kamppailua käydään isommalla tykistöllä ja strategisella tasolla. Vain muutama kuukausi sitten Google julkisti ostavansa Motorola Mobilityn ja toimitusjohtaja Larry Page hehkutti blogikirjoituksessaan[3]: ”Tulemme yhdessä [Motorolan kanssa] luomaan uskomattomia käyttäjäkokemuksia, jotka virittävät koko Android ekosysteemin kuluttajien, yhteistyökumppanien ja kehittäjien hyödyksi.” Ekosysteemilähtöinen ajattelu ja kehittäjien tärkeän roolin ymmärtäminen ovat myös Googlen ajattelun keskiössä.

Mitä ovat mobiiliekosysteemit?

Ekosysteemiajattelun määrittelyn voidaan jossain määrin sanoa alkaneen, kun James F. Moore loi Business Ecosystem -käsitteen Harward Business Review -lehdessä julkaistussa artikkelissaan vuonna 1993. Esimerkkinä ekosysteemistä hän käytti artikkelissa henkilökohtaisen tietokoneen (PC) ympärille syntyneitä IBM:n ja myöhemmin Microsoft-Intel ekosysteemejä. Mooren ajattelussa ekosysteemin muodostivat lähinnä yritykset, mutta hän tunnisti myös käyttäjän roolin ”vuorovaikuttavana organismina” yritysten muodostamassa ekosysteemissä. Digitalisoituva maailma ja Web 2.0:n kehitys ovat viimeisenä vuosikymmenenä mahdollistaneet ja luoneet käyttäjille entistä merkittävämmän ja monipuolisemman roolin ekosysteemissä. Käyttäjät eivät ole vain käyttäjiä, vaan myös aktiivisia osallistujia ja

kehittäjiä, jotka luovat uutta sisältöä ja sovelluksia ekosysteemiin. Digitaalisen ja business maailman yhteen sitomiseksi Nachira ja kumppanit (2007) määrittelivät Digital Business Ecosystem –käsitteen, jossa tekninen infrastruktuuri (digital ecosystem) ja yritykset, tuotteet sekä palvelut (business ecosystem) nivoutuvat toisiinsa. Älypuhelinalustojen ympärille muotoutuneet mobiiliekosysteemit ovat erinomainen esimerkki ekosysteemeistä, joissa nämä kaikki toimijat linkittyvät ja vuorovaikuttavat toistensa kanssa.

Yksi tapa visualisoida syntynyttä ekosysteemiä on kuvata vuorovaikutusketjut ekosysteemiin osallistuvien toimijoiden välillä (ks. kuva 1). Kuvassa on punaisilla nuolilla kuvattu, miten käyttäjä löytää sovelluksen kauppapaikassa ja käyttää sitä, ja vihreillä nuolilla vastaavasti, miten kehittäjä kehittää ja julkaisee uuden sovelluksen kauppapaikassa. Ketjuun osallistuvien erilaisten yritysten ja palveluntarjoajien, kuten mainospalvelutarjoajat, vuorovaikutukset on kuvattu sinisillä nuolilla.

Kilpailu ja yhteistyö mobiiliekosysteemeissä

Ekosysteemissä yritys ei ole enää yksin vaan vuorovaikutuksessa muiden toimijoiden kanssa. Minkälainen strategia täytyisi tällöin valita suhteessa muihin ekosysteemin toimijoihin? Yksinkertaistaen voidaan ajatella, että yritys voi kunkin toimijan kohdalla päättää joko tehdä yhteistyötä tai kilpailla.

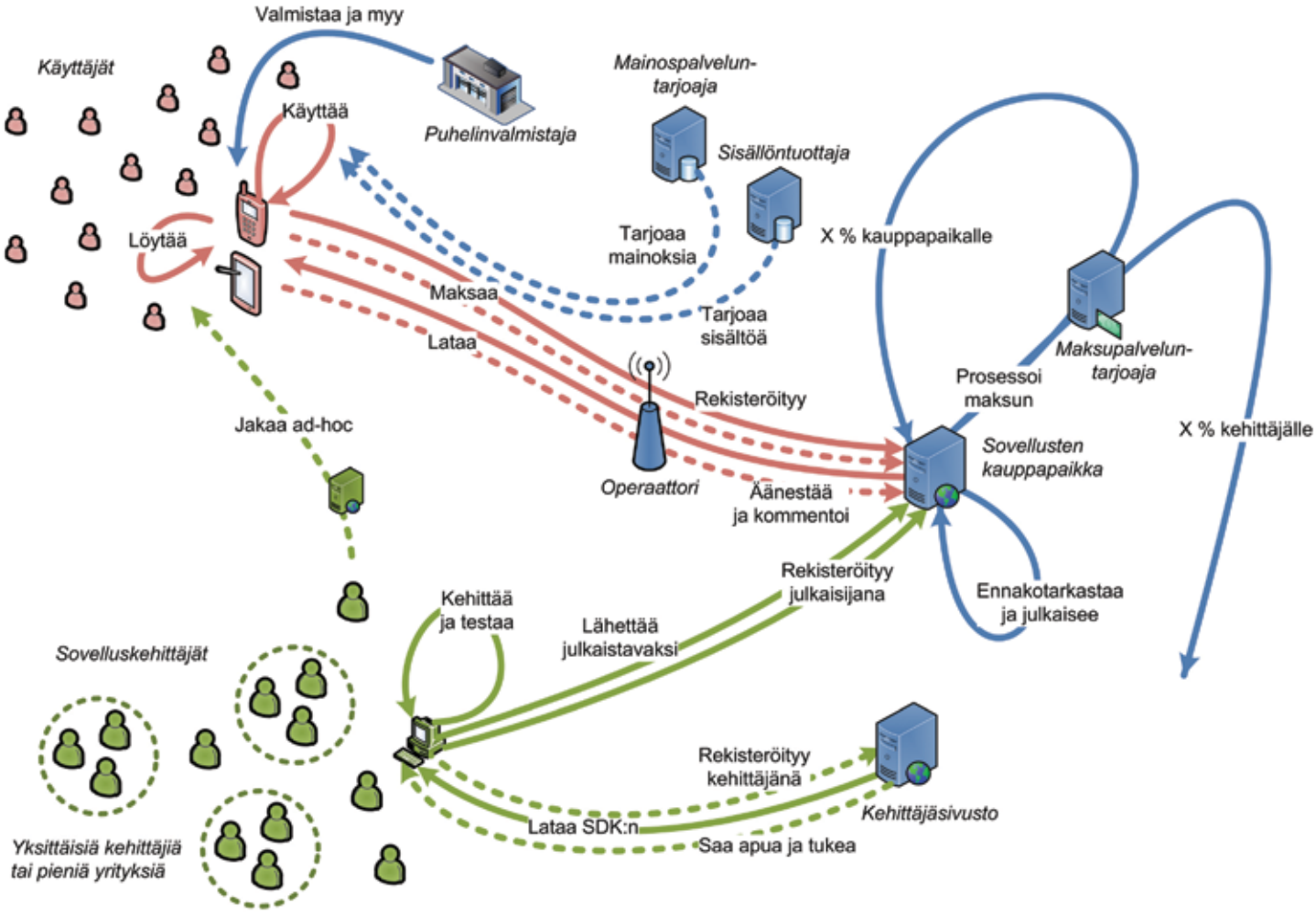
Esimerkki 1. *Apple on päättänyt kilpailla muiden puhelinvalmistajien kanssa valmistamalla omia puhelimia ja olemalla lisensoimatta iOS alustaansa Apple on valinnut olla tekemättä yhteistyötä heidän kanssaan.*
Bengtsson ja Kock (2000) ovat esittäneet, että yritykset saattavat myös valita coopetition-strategian eli kilpailla ja tehdä yhteistyötä samaan aikaan samojen yritysten kanssa tietyllä liiketoiminta-alueella.

Esimerkki 2. *Google on valinnut puhelinvalmistajien suhteen coopetition-strategian ostettuaan Motorola Mobilityn ja siten kilpailemalla muiden valmistajien kanssa ja samaan aikaan tekemällä myös yhteistyötä lisensoimalla Android-alustaansa halukkaille valmistajille.*

Samanlainen tarkastelu voidaan tehdä kaikkien ekosysteemin toimijoiden osalta. Taulukossa 1 on vertailtu kolmea johtavaa älypuhelinalustaa (iPhone, Android ja Windows Phone) tarkastelemalla alustojen takana olevien yritysten (Apple, Google ja Microsoft) tekemiä päätöksiä kunkin ekosysteemin toimijan kohdalla joko kilpailla, tehdä yhteistyötä tai valita coopetition-strategia.

Taulukosta voidaan havaita, että suurimman osan toimijoista suhteen eri älypuhelinalustat ovat valinneet saman strategian. Huomattavaa on, että muista toimialoista poiketen, mobiililiiketoiminnassa coopetition-ajattelu on selvästi korostunut. Kahden toimijan, puhelinvalmistajien ja sovellusten kauppapaikan, suhteen eri alustoilla on hyvinkin erilainen strategia. Applella on selvimmin hyvin kilpailullinen ja ekosysteemin kontrollia korostava strategia. Google pelaa avoimimmilla korteilla kilpailemalla ja tekemällä yhteistyötä samaan aikaan molempien edellä mainittujen toimijoiden suhteen. Microsoft on taas valinnut kaksijakoisen strategian nojaamalla puhelinten valmistuksessa täysin ulkopuolisiin toi-

Kuva 1. Älypuhelinalustojen ympärille syntynyt mobiiliekosysteemi



	Apple iPhone	Google Android	Microsoft Windows Phone
Puhelinvalmistajat	Kilpailu	Coopetition	Yhteistyö
Operaattorit	Yhteistyö	Yhteistyö	Yhteistyö
Sovelluskehittäjät	Coopetition	Coopetition	Coopetition
Mainospalveluntarjoajat	Coopetition	Coopetition	Coopetition
Sisällöntuottajat	Yhteistyö	Yhteistyö	Yhteistyö
Maksupalveluntarjoajat	Coopetition	Coopetition	Coopetition
Sovellusten kauppapaikka	Kilpailu	Coopetition	Kilpailu

mijoihin, mutta sen sijaan sovellusten kauppapaikan suhteen pitää kontrollin tiukasti omissa käsissään.
Jotta analyysissä päästäisiin syvemmälle, niin pelkistetyn kilpailu, yhteistyö ja coopetition luokittelun lisäksi on myös katsottava kilpailun ja yhteistyön aste-eroja kunkin toimijan kohdalla. Esimerkiksi suhtautumisessa kehittäjiin ekosysteemeillä on suuriakin eroja. Apple veloittaa kehittäjiltä 99 dollaria vuodessa kauppapaikassa julkaisemisesta kun taas Google ottaa ainoastaan 25 dollarin kertamaksun. Sovellusten ennakkotarkastusten suhteen Apple on ehdottomasti tiukin ja Google taas ei tee ennakkotarkastusta ollenkaan.

Johtopäätöksiä

Selvästikin mobiililiiketoiminnan markkinoilla on tilaa monenlaisille strategioille. Kolmen suurimman ekosysteemin joukossa ovat edustettuina niin yhdistettyyn kilpailuun ja yhteistyöhön nojaava kuin hyvin kilpailullinen ja kontrolloiva strategia. Voimme kysyä, olisiko markkinoilla tilaa vielä enemmän avoimuuteen ja yhteistyöhön nojaavalle strategialle? Samsung ja Intel julkistivat tämän vuoden syyskuun lopussa uuden Linuxiin pohjautuvan avoimen Tizen-alustan[4], joka saattaisi olla askel tähän suuntaan. Samoin Mozilla on käynnistänyt Boot to Gecko –projektin[5] täysin avoimen mobiilialustan rakentamiseksi. Tizen-alusta pohjautuu osin Nokian ja Intelin aikaisempaan Meego-yhteistyöhön ja retrospektiivisesti voidaan todeta, että itseasiassa Nokian lopettamistuomion saaneet Symbian, Maemo ja Meego -alustat tavallaan jo edustivat avoimempaa ajattelua.

Kaikki toimijat tunnustavat kehittäjien merkittävän roolin ekosysteemin menestyksessä. Ovatko yritykset kuitenkaan ottaneet tätä todellisuudessa huomioon strategiassaan ja pyrkineet parhain keinoin käyttäjälähtöisen innovaation ja sovelluskehityksen vauhdittamiseen? Esimerkiksi iPhone ja Windows Phone ekosysteemissä kehittäjän mahdollisuuksia jakaa omia sovelluksiaan (edes ilmaiseksi) ohi kauppapaikan ennakkosensuurin on merkittävästi rajoitettu. Puhelimen roolin kasvaessa ihmisten välisessä viestinnässä voidaan jopa kärjistäen kysyä, loukkaako sovellusten ennakkosensuuri jo nyt käyttäjien ja kehittäjien ”ihmisoikeuksia”.

Ekosysteemiajattelu ja nimenomaan digitaalisen näkökulman huomioonottaminen on olennaista ja sovellettavissa myös muille toimialoille kuten media, musiikkiteollisuus ja sähköisen kirjan toimialat. Mielenkiintoiseksi tämän kehityksen tekee se, että digitalisoitumisen myötä nämä toimialat konvergoituvat ja erojen tekeminen eri toimialojen välille käy entistä vaikeammaksi. Tämän kehityksen seuraaminen tulee olemaan todella mielenkiintoista ja käänteet ovat varmasti nopeita, kuten olemme viimeaikoina saaneet todistaa.

1) *http://www.engadget.com/2011/02/08/nokia-ceo-stephen-elop-rallies-troops-in-brutally-honest-burnin/*
2) *http://www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html*
3) *http://googleblog.blogspot.com/2011/08/supercharging-android-google-to-acquire.html*
4) *https://www.tizen.org/*
5) *https://wiki.mozilla.org/B2G*

Lähteet

Bengtsson, M., & Kock, S. (2000). “Coopetition” in Business Networks--to Cooperate and Compete Simultaneously. Industrial Marketing Management, 29(5), 411-426. doi:10.1016/S0019-8501(99)00067-X

Karhu, K., Botero, A., Vihavainen, S., Tang, T., & Hämäläinen, M. (2011). A Digital Ecosystem for Co-Creating Business with People. Journal of Emerging Technologies in Web Intelligence, 3(3). Retrieved from http://ojs.academypublisher.com/index.php/jetwi/article/view/jetwi0303197205

Moore, J. F. (1993). Predators and prey: a new ecology of competition. Harvard Business Review, 71, 75–75.

Nachira, F., Nicolai, A., Dini, P., Rivera Lèon, L., & Le Louarn, M. (2007). Digital Business Ecosystems. European Commission. Retrieved from http://www.digital-ecosystems.org/book/de-book2007.html

Yleinen kehitys mobiilimaailmassa

Ismo Penttinen

Kirjoittaja on mobiiliverkkomaailman asiantuntija NSN:n organisaatiosta.

Mitä merkitystä on sanoilla Bada, Carla, Donna, Harmattan, MeeGo, Meltemi, Tizen, Mango, Tango, Apollo ja iOS5? Ne kertovat todellisista hankkeista ja ohjelmistopäivityksistä, joilla mobiilimaailmaa viedään eteenpäin tällä hetkellä.

Nokian yllättävä päätös luopua Symbianista mullisti älypuhelimien markkinaosuuksia vähintään yhtä paljon kuin Applen ja Googlen mukaantulo mobiilimarkkinoille omine tuotteineen. Nokian strategianmuutoksen merkittävyyttä lisäsi myös luopuminen Symbian seuraajaksi kaavaillusta Linux-pohjaisesta MeeGo-Harmattanista. Näin markkinoille tuli "tilaa" monille uusille mielenkiintoisille hankkeille. Kuten Samsungin omalle Linuxiin pohjautuvalle mobiilikäyttöjärjestelmäalustalle Badalle, joka sai lisäpiristystä myös Googlen ostaman Motorolan aiheuttamasta uhasta (nykyisille Android-puhelimien valmistajille).

Meltemi taas kertoo Nokian pyrkimyksestä korvata nykyinen pitkään halpapuhelimissa käytetty S30/S40-alusta kilpailukykyisemmällä Linux-pohjaisella alustalla. Symbian Anna, Belle, Carla (nfc-lähiverkkotuki) ja Donna päivitykset (tuki moniydin prosessoreille) jatkavat Symbian 3 –pohjaisten laitteiden käyttöikää vielä muutamalla vuodella.

Tizen on vaihtoehto Androidille (vaikka molemmat pohjautuvat Linuxiin). Tizen jatkaa siitä mihin MeeGo jäi, mutta avoimempana ja Nokian QT-käyttöliittymä on korvautunut HTML5-pohjaisuudella nopeuttaen entisestään tuotekehitystä. Hanketta vetää operaattoreiden lisäksi Samsungin ja Intelin muodostama liittouma, johon liittyi tänä syksynä myös Linux Foundation ja LiMo Foundation. Tähän asti käytännössä kaikki älypuhelimien valmistajat Nokiaa ja Applea lukuun ottamatta ovat olleet riippuvaisia Googlen Androidista.

Windows Phone eli Mango oli jo lähes valmis, kun Nokia ilmoitti uudesta strategiastaan. Nokian kädenjälki näkyykin selvästi vasta seuraavassa ohjelmistopäivityksessä, joka tunnetaan tällä hetkellä nimellä Windows Apollo. Mutta väliin mahtuu pienempi päivitys nimeltään Windows Tango, joka on suunnattu hitaammille prosessoreille ja halvemman hintaluokan markkinoille.

Ensimmäiset 3D näytön omaavat kännykät ovat jo esitelty, mutta sitä ennen kiinnostavimmat uutuudet löytyvät epäilemättä tulevien 4G-puhelimien joukosta. Myös ensimmäiset 4G-tablet tietokoneet, joissa on sisäänrakennettu 4G-mokkula + sim-paikka, ovat jo tulossa lähikuukaisina - aluksi lähinnä vain Aasian-markkinoille.

Adobe yllätti hieman ilmoittamalla lopettavansa välittömästi Flashin kehittämisen mobiiliselaimiin. Myös Microsoft on antanut ymmärtää, ettei kilpailevalla Silverlightillakaan ole sijaa tulevaisuudessa ja on kääntänyt katseensa myöskin HTML5 tuomiin mahdollisuuksiin. Myös puheentunnistus ja pilvipalvelut ovat saaneet tuulta alleen mobiilimaailmassa lähinnä iOS5:n mukanaan tuomina.

Entä seuraavat uudet suuret mobiilimaailman pelurit? Ne voisivat olla esim. Amazon ja Facebook, joilla on jo valtavat käyttäjärekisterit sekä pilvipalvelut että identiteetin hallinnat. Amazonin Fire-tablet on jo esimakua tulevasta, jolla suuri yleisö houkutellaan maailman suurimman verkkokaupan digitaaliseen sisältöön. Kindle testasi konseptin toimivuuden e-kirjojen kanssa ja Applen iPad vahvisti sen toisaalla lehtimaailmassa. Lisäksi Amazonin suunnitelmat tuoda myös n. 150 dollarin Kindle-älypuhelimen integroituna omiin sisältöpalveluihinsa vahvistaa sen kykyä laajentua voimakkaasti myös uusillle mobiilialueille.

Myös Facebookilla on ollut jo pitkään suunnitteilla hanke koodinimellä Buffy, jossa muokattuun Androidiin olisi integroitu erityisesti Facebookin käyttöä helpottavia ominaisuuksia. Näin Google+ palvelun sekä Google Nexuksen tiukka integrointikaan ei takaisi helppoa menestystä sosiaalisen median mobiilimarkkinoillla.

Hannu Savolainen

Kirjoittaja on pitkän linjan ohjelmistosuunnittelija ja -arkkitehti, jonka aiemmat ideat mobiilisovelluksista ovat aina kaatuneet käytettävissä olevan laite- ja/tai tietoliikennekapasiteetin rajoituksiin. Mutta ehkä vielä joskus ...

MOBIILIT TIEDONSIIRTOARKKITEHTUURIT

Kuusikymmenluvun alussa syntyneenä muistan hyvin tietokoneet, jotka sarjakuvalehdissä ja television tieteissarjoissa antoivat vastauksen kysymykseen kuin kysymykseen. Ne olivat uskottavan kokoisia järkäleitä, jotka täyttivät kokonaisen hallin. Tieteissarjoista tulivat tutuksi myös kädessä pidettävät kommunikaattorit, joiden avulla sankari otti käden käänteessä videopuheluita ja pyysi apua kaukaisessa galaksissa olevasta päämajasta. Nyt 40 vuotta myöhemmin ei vieläkään oteta videopuheluita toiseen galaksiin, varsinkaan kun siihen ei ole mitään suurempaa tarvetta.

Tieto- ja kommunikaatioalan kehitys on jo vuosia sitten ohittanut tason, jonka tieteiskirjailijat pystyivät kuvittelemaan. Vieläpä niin, että muinaisten hehtaarihallin kokoisten tietokoneiden kyvyt ylittävä laskentakapasiteetti löytyy nykyisin monien teinienkin taskusta löytyvästä mobiili-päätelaitteesta, eli älypuhelimesta. Langattoman tietoliikenneyhteyden ansiosta niiden avulla on mahdollista löytää liki kaikki mahdollinen olemassa oleva tieto, esimerkiksi Wikipediasta. Erilaisten yhteisöpalveluiden kautta on mahdollista pitää yhteyttä kavereihin, vaikka vanhemmat pakottaisivat lähtemään viikonlopuksi kesämökille jonnekin jumalan selän taakse.

Mitä me oikein käytettiin ennen kuin oli kännykkä?

Mobiilili kommunikaatio perustuu radioaaltojen käyttöön. Ensimmäisissä kokeiluissa, yli sata vuotta sitten, käytettiin sähkötystä kipinälähettimillä, joiden signaali peitti lähes koko käytettävissä olevan radiotaajuisen osan sähkömagneettisesta spektristä. Myöhemmin, tekniikan kehityttyä, pystyttiin myös puheen ja musiikin lähettämiseen, ensin AM ja sitten FM-tekniikalla. Yleisradio- ja TV-lähetysten lisäksi radiotekniikkaa käytettiin myös sotilas-, ammatti- ja radioamatööriliikenteeseen. Mobiili kommunikaatio oli teknisesti mahdollista, mutta vain kahden, tarkoitukseen sopivan, radiolaitteen omistajan välillä. Vasta 40 vuotta sitten siirryttiin todelliseen mobiilikommunikaatioon, joka oli mahdollista myös puhelinverkon ja mobiilien päätelaitteiden välillä.

0G eli käsivälitteinen autoradiopuhelin (ARP)

Posti- ja telelaitos aloitti Suomessa käsivälitteisen ARP -palvelun (Auto Radio Puhelin) vuonna 1971. Palvelu toimi 150 MHz taajuusalueella, joka mahdollisti jopa yli sadan kilometrin yhteydet ajoneuvon ja tukiaseman välillä. Puhelut tilattiin keskuksesta, joka ilmoitti mobiililaitteen käyttäjälle käytettävän kanavan. Palvelu oli luotettava, mutta sen kapasiteetti kävi ajan myötä riittämättömäksi kasvavalle käyttäjämäärälle. ARP säilyi kuitenkin käytössä uudempien teknologioiden rinnalla. Se muuttui automaattivälitteiseksi 1990 ja poistui käytöstä vasta joulukuussa 2000. Dataliikenteeseen palvelua ei tiettävästi käytetty, vaikka se olisi ehkä ollut teknisesti mahdollista.

1G eli NMT450 (Nordisk Mobil Telefon)

NMT-matkapuhelinpalvelu otettiin käyttöön 1981 sekä Suomessa, Ruotsissa, Norjassa että Tanskassa. Myöhemmin palvelu otettiin käyttöön monissa muissakin maissa. Palvelu oli analoginen, mutta puhelut muodostettiin automaattikeskuksen kautta. Vaihto tukiasemalta toiselle ei ollut mahdollista vaan yhteys muuttui heikkolaatuiseksi ja katkesi, kun siirryttiin liian kauaksi alkuperäisestä tukiasemasta. Ensimmäiset päätelaitteet painoivat liki 10 kiloa ja herättivät ansaittua huomiota, kun ne hosti asiakastapaamisessa neuvotteluhuoneen pöydälle. Hintakin oli samaa luokaa kuin Lada-henkilöautolla. Autokäytössä se sijoitettiin auton takatilaan asennettuun autotelineeseen

ja sitä ohjattiin kojelautaan asennetun luurin avulla.

Datayhteydet olivat mahdollisia radio-osaan liitetyn modeemin avulla, mutta siirtonopeudet eivät juurikaan päätä huimanneet. NMT900-palvelu otettiin käyttäen vuodesta 1987 alkaen ja se mahdollisti hieman järkevämmän kokoiset päätelaitteet.

Mobitex-verkko

Mobitex oli Ericssonin ja ruotsin Televerketin kehittämä palvelu, joka oli suunniteltu mobiiliin tiedonsiirtoon. Palvelu otettiin käyttöön Ruotsissa vuonna 1986 ja hieman myöhemmin myös suomessa. Palvelu mahdollisti 512 tavun mittaisten sanomien lähettämisen, mihin kului aikaa noin 7 sekuntia. Pidempiä viestejä oli mahdollista lähettää jakamalla ne useisiin 512 tavun osiin. Päätelaitteet olivat suurikokoisia eli niitä voitiin käyttää lähinnä ajoneuvoon kiinteästi asennettuna. Sovelluksia olivat esimerkiksi tehtäväpyyntöjen lähettäminen huoltomiehen ajoneuvoon.

Toisen sukupolven mobiiliteknologiat (2G)

Varsinainen mobiilikommunikaation vallankumous alkoi vuonna 1992, kun digitaalinen GSM-palvelu otettiin käyttöön Suomessa. Tästä alkoi myös Nokian nousu telekommunikaatioalan jättiläiseksi. GSM oli ensimmäinen täysin digitaalinen matkapuhelinteknologia, joka mahdollisti yhteyden säilymisen myös tilanteessa, jossa siirryttiin tukiaseman alueelta toiselle. Tekniikan kehittymisen myötä päätelaitteiden koko ja hinta painui tasolle, jolla tavallisten kuluttajien oli mahdollista hankkia niitä. Teknologia mahdollisti uusia palveluita, joille löytyi heti kysyntää. Näitä olivat mahdollisuus data-, FAX- ja SMS -viestintään.

GSM data

Jo melko varhaiset GSM-kännykät, kuten Nokia 2110 (vuonna 1994), mahdollistivat datayhteydet matkapuhelimen kautta, esimerkiksi kannettavasta tietokoneesta. Siirtonopeudet olivat alhaisia, noin 9600 kbaud) ja kalliita, mutta yhdessä internetin yleistymisen kanssa se mahdollisti jopa rajoitetun etätyön tekemisen mobiililaitteen kautta, vaikkapa kesämökin laiturilta. Vuonna 1996 julkistetun Nokia 9000 Communicator -laitteen jälkeen siihen ei edes välttämättä tarvittu edes erillistä kannettavaa tietokonetta. Varsinainen weppiselailu pelkän kännykän avulkla ei silti ollut edelleenkään mahdollista. Alkuperäisen GSM-data -palvelun nopeus oli riittämätön, mitä yritettiin korjata high-speed-data -palvelun avulla, mutta silloinkin tiedonsiirtonopeudet olivat vain noin kaksinkertaisia, samoin hinta.

SMS (Short Message Service) eli tekstiviestipalvelu ja M2M

Tekstiviestipalvelu otettiin käyttöön vuonna 1995. Tässä vaiheessa kännyköiden hinnat olivat laskeneet tasolle, jolla niiden hankkiminen teini-ikäisten käyttöön oli mahdollista. Palvelu saavutti pian suuren suosion nuorison keskuudessa. Myöskään sen mahdollisuudet ammattikäytössä eivät jääneet huomaamatta. Tekstiviestejä käytettiin erilaisiin M2M (Machine-to-Machine) -sovelluksiin, kuten mittaustietojen etälähetykseen ja erilaisten hälytysten välittämiseen taustajärjestelmiin. Tekstiviestit ovat edelleenkin suhteellisen edullinen tapa välittää pieniä määriä informaatiota erilaisten järjestelmien välillä.

GPRS (Generalized Packet Radio Service)

Alkuperäisen GPS-data -palvelun aikaperusteinen hinnoittelu osoittautui melko pian epätyydyttäväksi käyttäjän kannalta. Internet-selailussa tai vaikkapa sähköpostia lukiessa varsinaiseen tiedonsiirtoon käytettävä aika on suhteellisen lyhyt varsinaiseen tekstin lukuun käytettävään aikaan verrattuna. Jos yhteys on koko ajan päällä,

niin hinta muodostuu melko pian kohtuuttomaksi. GPRS -palvelussa siirryttiin aikaperustaisesta hinnoittelusta siiretyn datan määrään perustuvaan hinnoitteluun. Samalla teoreettinen (hetkellinen) tiedonsiirtonopeus kasvoi. Silti siirrettyyn datamäärääkin sidottu hinnoittelu oli useimmille käyttäjille liian kallista. GPRS sopi kohtuullisesti sähköpostin lukemiseen ja päätepohjaisten (SSH, Telnet) -sovellusten käyttöön. Weppiselailu ja suurten tiedostojen siirto oli hidasta ja kallista, mutta toisaalta silloisilla kännyköillä se olisi ollut epämukavaa kuitenkin.

WAP (Wireless Application Protocol)

Vuonna 1999 Internet oli jos jokaisen tiedostavan kotitalouden käytässä, ainakin Suomessa. Oli tarvetta sille, että samoja palveluita voitaisiin käyttää myös mobiililaitteilla. Valitettavasti laiteteknologia ja käytettävissä oleva mobiili tietoliikenneteknologia eivät olleet vielä tilanteen vaatimalla tasolla. Ongelmaa yritettiin ratkaista tuomalla markkinoille mobiililaitteille sopiva kevytversio internetistä, joka oli WAP. Ajatus sinänsä oli hyvä. Toteutetaan internetpalveluita, jotka on optimoitu mobiililaitteille. Operaattorin hallinnoima WAP -yhdyskäytävä pakkaa sisällön mobiililaitteelle sopivaan muotoon ja laskuttaa siitä sovitun innaston mukaan. Tarpeen mukaan sisältö voitiin toimittaa joko SMS -viesteinä tai GPRS -yhteyden kautta. Valitettavasti harvat saatavilla olevat päätelaitteet, kuten Nokia 7110 eivät olleet aivan tilanteen tasalla. WAP oli yritys ratkaista siirtymävaiheen ongelmia, mutta se ei saavuttanut merkittävää suosiota. Palveluntarjoajille WAP -ratkaisuiden tukeminen olisi merkinnyt kaksinkertaista työmäärää verrattuna jo yleisesti käytössä olevien HTTP-pohjaisten palveluiden lisäksi.

Kamerakännykät ja multimediaviestit (MMS)

Mobiililaitteiden teknologia jatkoi nopeaa kehitystään 2000-luvun alussa. Tuli mahdolliseksi liittää kännyköihin kamera ja suhteellisen korkearesoluutioinen värinäyttö. Alkoi olla mahdollista ottaa kännykällä valokuvia, kuvata videota ja/tai tallettaa ääntä. Sen myötä synti tarve lähettää tallenteita myös kavereille. Tätä tarkoitusta varten luotiin MMS (Multimedia Messaging Service) -palvelu. Palvelussa mobiililaite (kamerakännykkä) tallettaa kuvasta, äänestä ja/tai tekstistä koostuvan viestin operaattorin palvelimelle, joka välittää sen edelleen vastaanottajalle.

EDGE (Enhanced data rate for Global Evolution)

Alkuperäinen GSM-data tai edes GPRS eivät täyttäneet niitä tarpeita, joita yleistyvä mobiili internet -käyttö asetti. Siksi kehitettiin uusi EDGE -järjestelmä, joka mahdollisti (teoriassa) 236,8 kb/s -tiedonsiirron. Järjestelmä otettiin käyttöön Yhdysvalloissa vuona 2003. Nykyisin EDGE on yhteyskäytäntö, jota käytetään tilanteissa, joissa uudemmat 3G-palvelut eivät ole saatavilla. EDGE mahdollistaa jo kohtuullisen nopean www-selailun ja sähköpostin lukemisen, ainakin tilanteissa, joissa nopeampaa yhteyttä ei ole saatavilla.

Kolmannen sukupolven mobiiliverkot (3G)

Aivan 2000-luvun alussa internet oli jo lyönyt itsensä läpi ja mobiililaitteiden kehitys kulki kohti vaihetta, jolloin todelliset internet -yhteydet olisivat mahdollisia myös mobiililaitteilla. Kasvuodotukset olivat valtavia, minkä vuoksi useat Euroopan maat, mukaan lukien Saksa, järjestivät huutokaupan halukkaille palveluntarjoajille. Valitettavasti odotukset olivat suhteettomia senhetkisiin tek-

nisiin mahdollisuuksiin verrattuna ja huutokaupoissa maksetut tähtitieteelliset summat hidastivat 3G-teknologioiden läpimurtoa liki 10 vuotta.

3G verkkoteknologiat, etenkin nykyinen HSPA-teknologia, mahdollistavat liki laajakaista-tasoiset verkkoyhteydet myös mobiililaitteisiin. Myös mobiililaitteiden, lähinnä älypuhelimien, suorituskyky on noussut tasolle, jolla ne voivat hyödyntää nopeita verkkoyhteyksiä esimerkiksi reaaliaikaisen videokuvan katseluun. Valitettavasti 3G-teknologioiden mahdollistama kaistaleveys on rajallinen ja usein varsinkin pääkaupunkiseudulla käytettävissä oleva kaista loppuu kesken. Silti parhaimmillaan 3G-yhteydet mahdollistavat jopa reaaliaikaisen videon, esimerkiksi YouTube, katselun. Aiemmin NMT- ja GSM-käytössä olleen 900 megahertsin taajuusalueen siirryttyä 3G-käyttöön, nopeat mobiiliyhteydet ovat käytettävissä jopa haja-asutusalueilla.

Neljännen sukupolven mobiiliviestintä (4G)

Käyttäjämäärän kasvun ja mobiilien päätelaitteiden (älypuhelimet) yleistymisen myötä nykyisen 3G teknologian rajat alkavat tulla vastaan. Niinpä uusimpia neljännen sukupolven (4G) teknologioita, kuten LTE Long Term Evolution) ollaan ottamassa käyttöön. Tällä hetkellä suomalaisilla teleoperaattoreilla on jo rajoitetussa kokeilukäytössä olevia verkkoja muutamilla paikkakunnilla. Toistaiseksi sopivien päätelaitteiden saatavuuskin rajoittaa palvelun yleistymistä. Ensimmäiset markkinoilla olevat laitteet ovat matkamikroon kytkettäviä ”mokkuloita”. Nykyisten laitteiden virrankulutus on sen verran korkea, että se rajoittaa kokopäiväisen työskentelyn lähinnä paikkoihin, joissa laite voidaan kytkeä verkkovirtaan. Verkkojen rakentamiseen ja laiteteknologian kehittymiseen menee todennäköisesti joitakin vuosia ja vasta tulevaisuus näyttää, kuinka laajamittaiseen käyttöön niistä on.

Tällä hetkellä LTE on teknologia, jota eurooppalaiset operaattorit ja laitevalmistajat markkinoivat ensimmäisenä neljännen sukupolven mobiiliteknologiana. Kuitenkin, tarkkaan ottaen, se on vain 3.9G. Alkuperäisen ITU:n määritelmän mukaisen 4G teknologian olisi pystyttävä 1Gbit/s nopeuteen paikallaan oltaessa ja 100 Mbit/sek liikkeessä. LTE:n nopeus kuitenkin jää, koeolosuhteissa, vain alle 100 Mbit/s lataussuuntaan ja alle 20 Mbit/s lähetyssuuntaan. Lisäksi yksittäisen tukiaseman käytettävissä oleva kaistanleveys jaetaan kaikkien sen alueella olevien laitteiden kesken. Jotta päästäisiin edes lähelle teoreettista siirtonopeutta, on tukiasemia oltava riittävän tiheässä. Lisäksi LTE käyttää aiempaa korkeampia radiotaajuuksia, jotka läpäisevät heikommin rakennuksien seiniä ja muita esteitä. Siksi LTE:n suuri nopeus saavutetaan vain, jos tukiasemia on melkein joka korttelissa tai kauppakeskusten ja kahviloiden sisätiloissa.

Viidennen sukupolven verkot (5G)

Japanissa ja Yhdysvalloissa ollaan jo ottamassa käyttöön LTE Advanced -teknologiaa. Alun perin sen piti olla ensimmäinen varsinainen 4G -verkkoratkaisu, mutta sitä markkinoidaan nykyisin 5G -verkkona. Ensimmäisiä, laajan mittakaavan, verkkoja ovat avanneet NTT DoCoMo Japanissa ja Sprint Yhdysvalloissa. Jopa hieman ennen näitä ehdittiin avata verkot myös Ruotsissa ja Norjassa. LTE Advanced on teknisesti melko samanlainen, kuin LTE. Teknologialla pystytään teoriassa 1 Gbit/s siirtonopeuteen.

Mitä tulevaisuudessa?

Teknologia jatkaa todennäköisesti tulevaisuudessakin kehittymistä suurin piirtein nykyisellä vauhdilla. Tosin kehityksen suunnan ennustaminen on jo paljon vaikeampaa. Kaikille teknologian kehittymisen mahdollistamille palveluille ei loppujen lopuksi löydy kysyntää. Niinpä hyödyllisempi lähestymistapa on miettiä, mitä me kuluttajina oikeasti tarvitsemme. Alun perin merkityksettömältä tuntunut innovaatio, kuten tekstiviestit, saattaa osoittautua todelliseksi tappajasovellukseksi. Niinpä tuotekehittäjien kannattaisi panostaa nykyistä enemmän hyvien ongelmien etsimiseen sen sijaan että yritetään, teknologiavetoisesti, kehittää ratkaisuja ongelmiin, joita kenelläkään ei ole.

Q-FACTORY
www.q-factory.fi

Ilpo Kojo
Kirjoittaja on akateemiselta taustaltaan filosofian tohtori ja psykologian dosentti Helsingin yliopistossa. Hän toimii tieteellisenä johtajana Fluid Interaction Oy:ssä.
[ilpo@fluid.fi] www.fluid.fi

Tietämyksen hallintaa ihmisen ehdoin: Front-end intelligence

Digitoidun tietämyksen määrä lisääntyy ja muodot monipuolistuvat valtavalla vauhdilla. Pullonkaulaksi on muodostunut käyttäjän kyky omaksua ja ymmärtää tietojärjestelmien syytämää informaatiota. Eräänä ratkaisuna on muokata tietojärjestelmän tuotokset mahdollisimman hyvin ihmisen ymmärtämään muotoon. Siihen tarvitaan tutkimustietoa ihmisen tiedonkäsittelytavoista.

Biologiset ja piihin perustuvat tietojärjestelmät

Tietolaitteet käsittelevät ja tallentavat tietoa bitteinä ja niiden erilaisina kombinaatioina. Bitit tulevat laitteisiin monista eri lähteistä, näppäimistöltä, hiireltä, kameroista, mikrofoneista jne. Bitit pysyvät bitteinä, mutta niiden kombinaatiot ylemmillä tasoilla riippuvat sen tiedon luonteesta, jota bitit edustavat.

Ihmisen hermojärjestelmässä tilanne on samankaltainen, eri aistipiirit muuttavat ulkomaailman fysikaalisia ja kemiallisia tapahtumia hermoimpulsseiksi, joita aivot sitten jatkojalostavat ja tuottavat koettuja aistimuksia, havaintoja ja muistijälkiä. Ihmiselle on ominaista pyrkiä yhdistämään eri aistijärjestelmistä saatua tietoa ja ulkomaailman käsittelyssä (motoriikassa) tarvittavaa osaamista monipuoliseksi kokonaisuudeksi. Kasvavat ja kehittyvät lapset käyttävät vuosia elämästään näiden integraatiotoimintojen osien synkronointiin. Tarpeen vaatiessa myös aikuiset voivat kehittää itselleen uusia integroituja toimintoja.

Ihmisille ovat luontaisia kohteita sellaiset, joita voi katsella, kuunnella, haistaa, maistaa ja kosketella. Tietolaitteet voivat puolestaan prosessoida asioita, joita voi mallintaa ja joiden ominaisuuksia voi selvittää laskemalla. Ihmisille ominaiset taidot ovat monipuolisia, mutta usein hitaita. Tietojärjestelmät ovat puolestaan monissa asioissa käsittämättömän nopeita. Toivottuun tulokseen voidaan päästä yhdistämällä molempien vahvuudet, tietokoneiden nopeuden ja ihmisen ajattelun monipuolisuuden ja joustavuuden.

Esimerkkinä Twitter ja 250 miljoonaa viestiä päivässä

Twitterin kautta kulkeva viestimassa on kuin valtava bittien kaatopaikka, josta kuitenkin löytyy helmiä ja kultahippuja melkein kaikille tietämyksen alueille. Ne vain tulee osata kaivaa esiin. Merkityksellisiä viestejä voi luokitella esimerkiksi aiheiden tai viestin lähettäjäksi tai vastaanottajaksi merkityn henkilön (tai nimimerkin) avulla. Yleisesti loppukäyttäjä antaa Twitterille ohjeet siitä, mitä hän haluaa ja Twitter- järjestelmä hoitaa viestien luokittelun. Oikeiden ohjeiden antaminen on tärkeää, koska hatara määritelmä tuottaa valtavasti viestejä ja tiukka saattaa estää kultajyvienkin löytymisen.

Tässä yhteydessä tarvitaan käsitettä ”Front-end intelligence”. Sillä tarkoitan sitä, että käyttöliittymä (UI) esittää välitetyt tiedot käyttäjän kannalta optimaalisella tavalla. Tässä esiteltävässä Different Tack –nimisessä twitter –käyttöliittymässä viestit on järjestetty ympyrän sektoreihin. Sektoria koskemalla käyttäjä saa näkyviinsä viestin sisällön omaan ruutuunsa, joka on näytöllä sektoroidun ympyrän vieressä (ylhäällä, alhaalla, oikealla tai vasemmalla, käyttäjän halun mukaan). Käyt-

täjän koskettama sektori ja sen vieressä olevat sektorit levenevät helpottaen sektorin valitsemista jatkokäsittelyyn tai sen hylkäämistä ja seuraavaan sektoriin siirtymistä. Kun joku viesti valitaan, viestivalikon tilalle avautuu sektoroitu toimintovalikko, joka esittää, mitä kyseiselle viestille voi tehdä, tietenkin kyseistä sektoria käyttämällä.

Different Tack –käyttöliittymässä ja sen taustalla olevassa käyttöliittymäkonseptissa sisäänrakeennettu "Front-end intelligence" arvioi viestin sisällön perusteella (kontekstisensitiivisesti), mitä viestille voi tehdä ja tarjoaa vain ne mahdollisuudet. Käyttäjän toiminnan perusteella (peukalon tai sormen liike kosketusherkällä näytöllä) UI kasvattaa käyttäjän tarkoittamia sektoreita ja pienentää muita. Tässä yhteydessä on sovellettu tietoa keskushermoston toiminnasta. Tieto silmästä aivojen päätöksentekomekanismeihin kestää muutaman sataa millisekuntia. UI tekee ennakointinsa (proaktiivisuus) heti käyttäjän sormen liikkeen alussa ja muokkaa näyttöä sen mukaisesti jo ennen kuin käyttäjä ehtii oivaltaa mitä on tekemässä. Niinpä käyttäjän tarvitsema näyttösisältö on jo valmiina odottamassa ennen kuin hän on ehtinyt suorittaa valintaansa loppuun. Kaikki tapahtuu pienellä peukalon tai sormen liikkeellä. Käyttäjän kokemus on ollut usein hämmennyksen ja ihastuksen kombinaatio.

Perustutkimuksesta poikinut sovellus

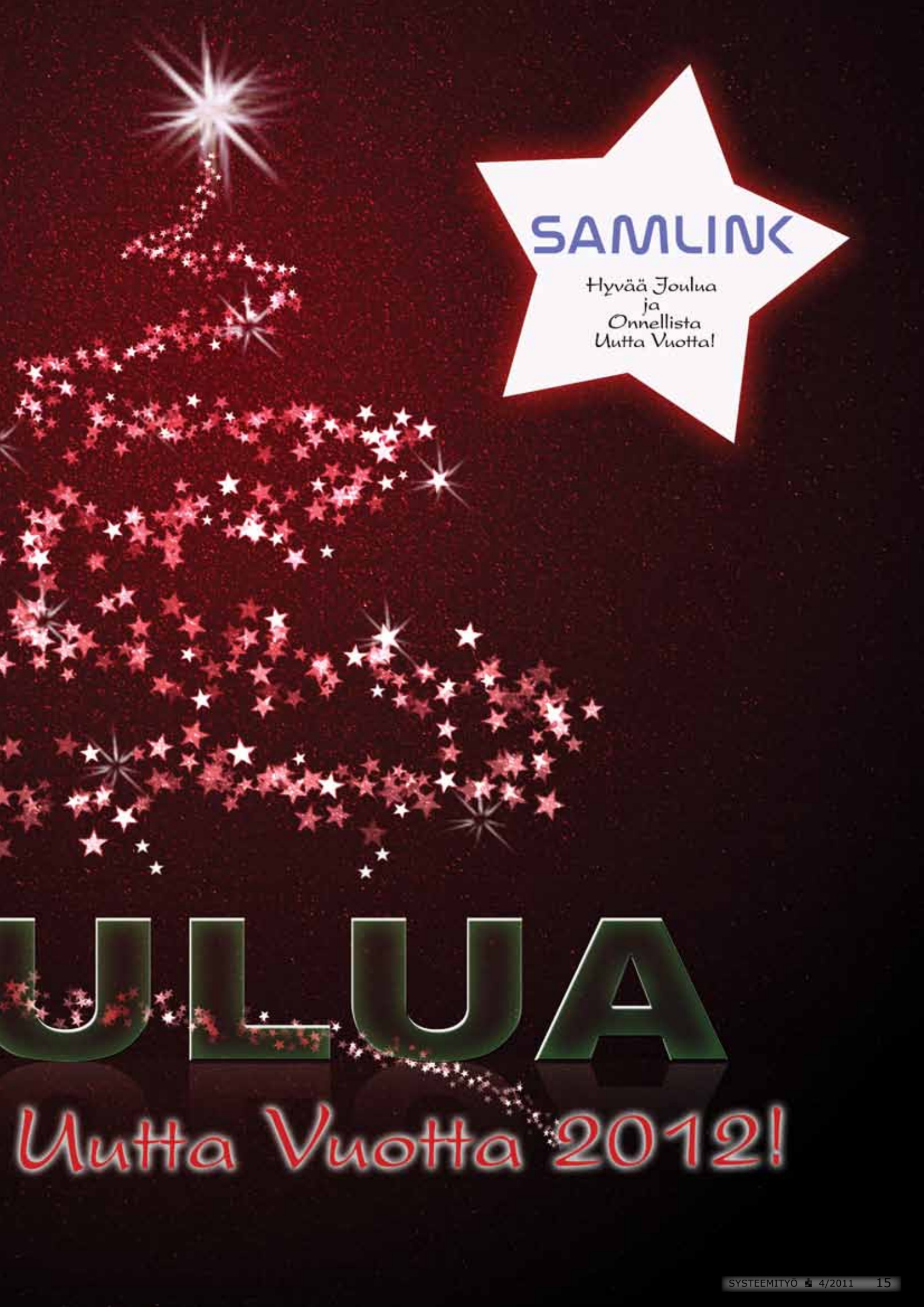
Tässä kirjoituksessa esitetty UI-konsepti on kehittynyt paljolti perustutkimuksen sivutuotteena. Tutkimukseen on kulunut paljon tutkijoiden resursseja ja rahaa (Suomen Akatemialta ja Tekesiltä) ja tuloksena on tullut tieteellisiä julkaisuja ja joitain väitöskirjoja. Teknologian nopea kehittyminen ja halpeneminen ovat mahdollistaneet kokeilut, joita ei vielä parikymmentä vuotta sitten voinut kuvitella toteutettaviksi. Tärkein yksittäinen innovaatio on Applen laajoille joukoille lanseeraamat kosketusnäyttöiset iPad ja iPhone sekä tietysti muiden toimijoiden vastaavat tuotteet. Pelkkä patentoitu UI-konseptin idea ei toki riitä. Ideasta tuotteeksi on pitkä matka ja valmiiseen tuotteeseen on tarvittu sekä ohjelmointi- että designosaamista.

Mitä Twitterillä voi tehdä?

Twitter on viestiohjelma, jonka kautta voi lähettää lyhyitä viestejä, esim. omia ajatuksia tai linkkejä muualle. Sen kautta kulkee päivittäin n. 250 miljoona viestiä (tweet). Twiittien virtaa voi seurata esim. asiasanojen tai henkilöden perusteella. Sitä käyttävät presidentit, ministerit ICT-julkkikset ja tavalliset kansalaiset. Nykyisin monet tärkeät tiedot lähetetään ensimmäisenä Twitterin kautta. Sitä voi käyttää myös ryhmissä ideoiden, kysymysten ja neuvojen välittämiseen. Jotkut pitävät sitä erinomaisena innovointityövälineenä sen nopeuden ja helppokäyttöisyyden takia. Huono puoli voi olla se, että käyttäjä sortuu informaatioahneuteen. Tällöin syntyy infoähky (information overflow). Different Tack on suunniteltu infoähkyn hallitsemiseen.

Lisää aiheesta: Berg ja Kojo (2011): Informaation visualisointi. Kirjassa: Oulasvirta (toim.) Ihmisen ja tietokoneen vuorovaikutus, Gaudeamus.

Tietoalan toimihenkilöt
Tietoalan toimihenkilöt toivottaa kaikille Rauhallista Joulua!
AVAREA
Hyvää joulua Sytykkeen lukijoille toivottaa Avarean väki.
Enjoy VG Life
Viking Line toivottaa asiakkailleen Hyvää Joulua!
Hyvää
JOU
&
Sytyttävää


SAMLINK
Hyvää Joulua
ja
Onnellista
Uutta Vuotta!
JOULUA
Uutta Vuotta 2012!

Ilkka Pirttimaa

Head of Technology, Stockmann. Kirjoittaja on Stockmannin pääarkkitehti ja kehittää mobiilisovelluksia harrastuksenaan.
[ilkka.pirttimaa@stockmann.com]

Kuinka aloittaa mobiilikehitys iOSille?

iOS-kehittäjälisenssi maksaa 79 euroa vuodessa. Ilman lisenssiä voi toki kehittää, mutta sovelluksia ei voi silloin siirtää lainkaan laitteisiin: Sovelluskehitys laitteeseen vaatii aina kehittäjäsertifikaattilla allekirjoittamisen - Tämä malli on osa iOS-laitteiden tietoturvaa.

Kehittäjälisenssiin sisältyy myös pääsy kattaviin sovelluskehitysohjeisiin ja esimerkkisovelluksiin. Vuosittaisesta WWDC-kehittäjäkonferensista julkaistaan myös satoja tunteja videoituja esityksiä esityskalvoineen kehittäjille. Toisaalta kehittäjälisenssiin kuuluu tekninen tuki, tosin vain yksi ilmainen tekninen kysymys / vuosi. Itse en ole tätä palvelua koskaan tarvinnut, sillä iOS-kehittäjäyhteisö on suuri ja keskustelupalstoilta löytyy vastaus lähes kaikkeen eteentulevaan.

Kehitysympäristö XCode

Sovelluskehitykseen tarvitaan kehityskone, jossa on MacOS-käyttöjärjestelmä. Mikä tahansa kone käy, itselläni on Macbook 13”, jossa on 8 GB keskusmuistia. Näyttöresoluutiona 1024 x 768 on vain liian pieni, jos toteuttaa paljon käyttöliittymiä iPadille.
Sovelluskehitysympäristö XCode on ilmainen. Se sisältää kaiken tarpeellisen, mm.

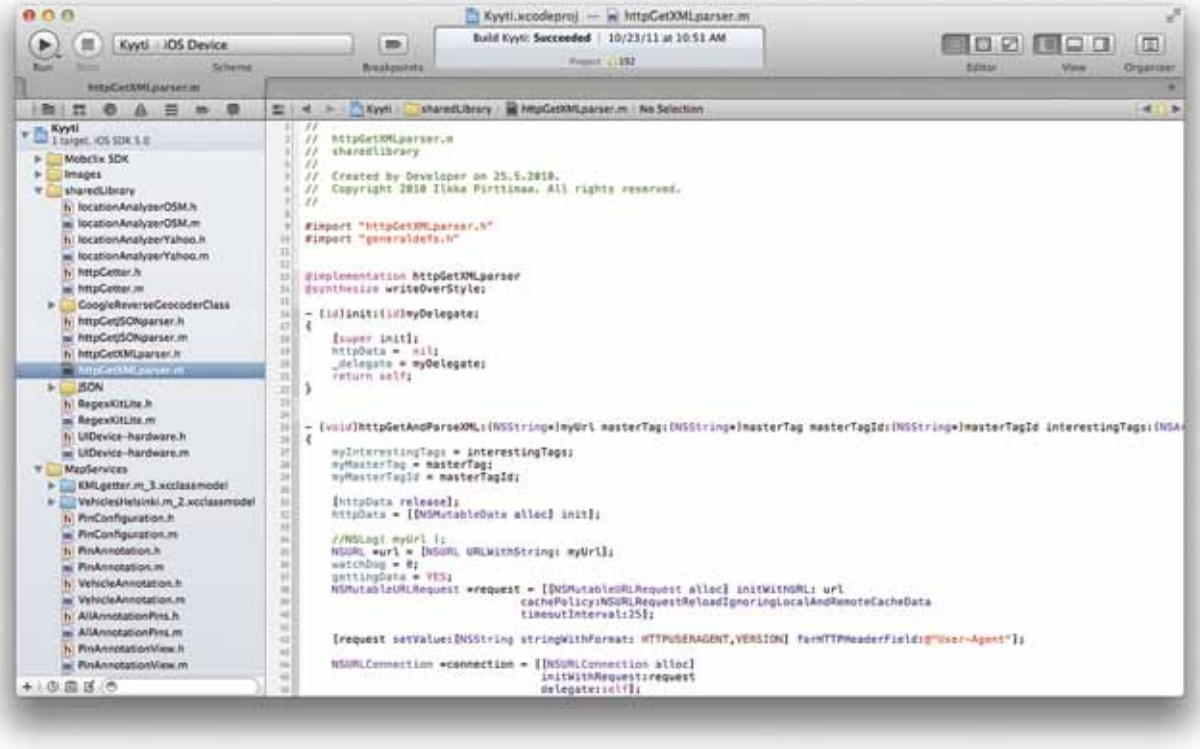
- Editori
- Projektipohjat ja wizardit erityyppisille sovelluksille
- Paikallinen git-versiohallinta
- Simulaattorit iPhonelle ja iPadille, useille iOS-käyttöjärjestelmäversioille
- Debuggeri
- Käyttöliittymäeditori
- Tietokannan ER-mallinnin ja siihen liittyvä objektimallin generointi
- Kattavat analysaattorit mm. suorituskyvyn, muistivuotojen jne. mittaukseen
- Välineet sovelluksen paketointiin
- Organizer laitehallintaan ja sovelluspakettien adhoc-julkaisuun, yrityssovellusjulkaisuun tai App Store-julkaisuun
- Yksikkötestausympäristö, jossa yksikkötestien ajo voidaan automatisoida suoritettavaksi käännösten yhteydessä

Ohjelmointia tehdään pääasiallisesti objective-C-kielellä, mutta myös C ja C++:n käyttö on mahdollista.

Natiivisovellus vai HTML5-pohjainen webbipalvelu?

Ennenkuin aloittaa mobiilisovelluksen teon, kannattaa ensin pohtia, tarvitaanko ns. natiivi iOS-sovellus, vai riittäisikö mobiilioptimoidun www-palvelun tekeminen. Natiivisovelluksen etuja verrattuna webbipalveluun on mm.

- Sovellus saadaan erittäin responsiivikseksi – Näytöllä asiat tapahtuvat heti eivätkä “sivun latauduttua”.
- Voidaan hyödyntää kameraa, sijaintia ja kiihtyvyysantureita
- Voidaan tehdä integraatioita puhelimen muihin sovelluksiin, kuten esim. yhteystiedot, kalenteri ja valokuvat
- On helpompaa toteuttaa käyttöliittymiä, jotka noudattavat laitteen käyttöliittymästandardeja
- Mahdollisuus myydä sovellusta App Storessa
- Mahdollisuus jakaa sovellusta ilmaiseksi, mutta hyödyntää App Storen suosittujen sovellusten listan tuomaa näkyvyyttä



Osaa näistä asioista voidaan toki toteuttaa myös tekemällä mobiilioptimoitu webbipalvelu. Toisaalta voidaan tehdä ns. hybridisovellus: Toteutetaan itse palvelu webbipalveluna ja toteutetaan iOS:lle (ja mahdollisesti muille mobiiliympäristöille) sovellus, jossa on sisällä webbiselain, joka avaa webbisovelluksen verkosta. Mikäli sovelluksessa tulee tarvetta hyödyntää esim. kameraa tai kiihtyvyysantureita, voidaan nämä osiot toteuttaa sovellukseen natiivikoodilla ja välittää tieto palvelimelle esim. javascirptillä tai erillisillä http-kutsuilla siten, että tiedot yhdistyvät näytöllä olevaan sessioon. Hybridisovellusten tekoon löytyy myös valmiita kehikoita, mm. Adoben PhoneGap (http://phonegap.com/).

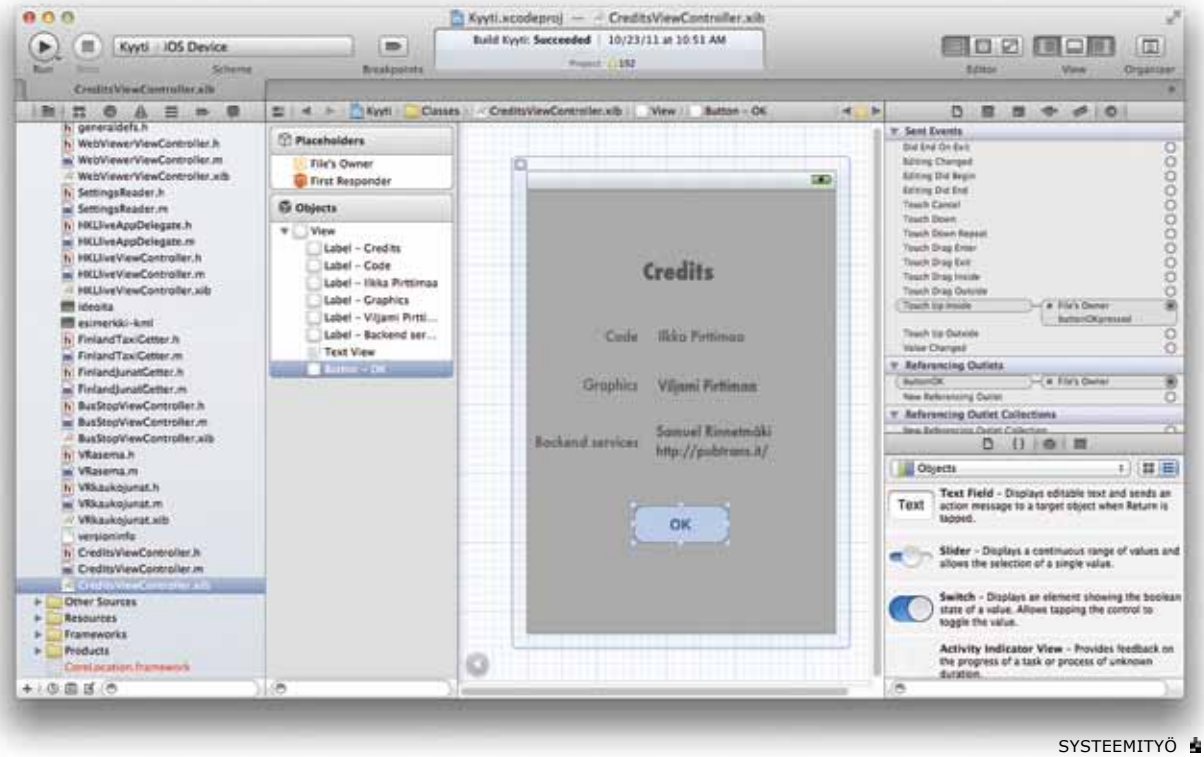
Natiiviohjelmointi Objective-C:llä

iOS-laitteille sovelluskehitys tehdään Objective-C -kielellä: Samalla, jolla myös MacOS on tehty ja myös monet suositut MacOS-sovellukset, kuten esim. Adobe Photoshop.

Sovelluskehittäjän on syytä varata aikaa kielen opiskeluun: Se muistuttaa C:tä ja Javaa, mutta on perinyt objektisyntaksin Smalltalkista. Ehkäpä suurin eroavaisuus on metodikutsut: Ne tehdään lähettämällä viestejä instansseille:

```
- (int) laskeSumma:(int)luku1 toinen:(int)luku2
{
    return luku1+luku2;
}

- (void) tulosta2plus2
{
    NSLog(@"2 + 2 on %d",[self laskeSumma:2 toinen:2]);
}
```



Tässä esimerkissä on toteutettu metodi nimeltä "laskeSumma", joka saa kaksi int-tyyppistä parametria. Metodi palauttaa näiden summan.

Toinen metodi "tulosta2plus2" käyttää NSLog-funktiota (huom! Tämä on C-tyylinen kutsu) logitulostukseen. Parametrikseen se ottaa tuloksen, joka saadaan lähettämällä viesti self-instanssille. Viestin käsittelyn lopputulos sijoitetaan NSLog:in formatointilauseeseen. Virheet näin tehdyissä metodikutsuissa havaitaan vasta ajoaikaisena, minkä takia yksikkötestaukseen ja testausten kattavuuteen kannattaa kiinnittää erityishuomiota.
Toinen merkittävä ero muihin ohjelmointikieliin liittyy muistinhallintaan. Kaikki objektit periytyvät NSObject-luokasta, joka sisältää muistinhallintaan liittyvät perusmekanismit. Muistinhallinta perustuu referenssimäärien laskentaan. Esimerkiksi seuraava koodi saattaa alkuun hämmästyttää:

```
Kirja *munKirja = [[Kirja
alloc] initWithTitle:
@"Aapiskirja"]];
[kaikkiKirjat addObject:
munKirja];
[munKirja release];
```

Ensimmäisellä rivillä luodaan Kirja-tyyppinen objekti, johon varataan muistia (alloc) ja se alustetaan kirjan nimellä (huom. []-syntaksi: nämäkin operaatiot tehdään lähettämällä viestejä!). Toiselle rivillä tämä kirja lisätään taulukkoon, jossa on kaikki kirjat.

Viimeisellä rivillä äsken luomamme kirja vapautetaan. Itseäni tämä ihmetytti alkuun: Miksi objekti vapautetaan, vaikka se on listalla jossa siihen viitataan?

Selityksenä on NSObject, josta myös Kirja-objekti on peritty: Se sisältää referenssilaskurin. Aina, kun objektia käytetään (esim. lisätään listalle), sen referenssilaskuria lisätään. Yllä olevan koodin toinen rivi kasvattaa siis referenssilaskuria 2:een. Viimeinen rivi kertoo, että kutsuva koodi ei ole enää kiinnostunut objektista, jolloin referenssilaskuri laskee 1:een. Koska referenssilaskuri ei mennyt nollaan, objekti jää edelleen muistiin ja vapautuu vasta, kun se poistuu listalta.

Eri laitemallien huomioiminen

Apple on tehnyt eri laitemallien ja käyttöjärjestelmäversioiden tukemisen varsin helpoksi. Tässä on yksi suljetun järjestelmän eduista: Eri laitemallien määrä on rajallinen ja uusien ominaisuuksien lisäys on mietitty niin, että vanha sovelluskoodi harvoin menee rikki. Esimerkiksi kun iPhone 4 ilmestyi, sen retina-näyttö tuplasi näytön pikselimäärän. Sovelluskoodissa käytetään kuitenkin ehkä hieman yllättäen alkuperäisen iPhonen koordinaattijärjestelmää. Tuplatarkkuusgrafiikat saadaan käyttöön lisäämällä sovellukseen kuvat tuplaresoluutiolla ja nimeämällä kuvatiedostot lisäämällä nimen loppuun "@2x.png". Sovelluskoodiin ei tarvitse koskea vaan iOS-käyttöjärjestelmä lataa automaattisesti tuplatarkkuuden kuvan, mikäli sovellusta ajetaan laitteella jossa on retina-näyttö. Tästä on helppo arvata, että kun seuraava iPad julkaistaan, sen resoluutio on mitä ilmeisimmin 2048 x 1536, mikä on tasan tuplat nykyiseen iPadiin.

Sovellustestausta voi tehdä iPhone- ja iPad-simulaattoreilla, jotka on integroitu XCodeen. Simulaattorien lisäksi sovellus on kuitenkin hyvä testata myös oikeilla laitteilla, jolloin esim. integroinnit osoitekirjaan yms. voidaan todentaa.

Kun toteutetaan iOS-sovellusta, pitäisi pyrkiä välttämään laitteen tyypin tunnistusta (iPhone, iPad, iPod Touch ja näiden kaikki variaatiot). Sen sijaan pitäisi käyttää iOS:n palveluita, joilta voi kysyä, mitä laite osaa (esim. resoluutio, kamera, kiihtyvyysanturit jne.). Näin taataan yhteensopivuus tuleville laitteille.

iCloud - Dataa pilveen

iOS 5:n yksi suurimmista muutoksista on iCloud. Moni puhelimen perustoiminto tukee nyt Applen pilvipalveluita: Kalenteri, sähköpostit, osoitetiedot ja jopa valokuvat tallentuvat haluttaessa Applen palvelimille ja sitä kautta muille iOS-laitteille, webbiin ja työasemiin. Vihdoin esim. kontaktitietoja ei tarvitse enää siirtää puhelimesta toiseen puhelinta vaihdettaessa.

iCloud tarjoaa myös APIt sovelluskehittäjille: Sovellus voi tallentaa ja lukea dokumentteja tai avain-arvopareja pilvestä. Esimerkiksi iCloud APIa hyödyntävä ostoslistasovellus toimisi kaikilla laitteilla synkronoituna siten, ettei tarvita omaa palvelinpään sovellusta.

Sovelluksen julkaiseminen

Kun sovellus on valmis ja testattu, voi sen julkaista. Julkaisuun on kolme pääasiallista tapaa:

1) Julkaisu App Storessa (tulee ladattavaksi kaikissa maailman iOS-laitteissa)
2) Adhoc-jullkaisu määrätyille laitteille (voit listata 100 laitetta, jossa voidaan ajaa tekemääsi sovellusta). Tämä julkaisutapa sopii betatestausvaiheeseen
3) Enterprise-julkaisu mahdollistaa yrityksen sisäisen sovellusjakelun omiin laitteisiin esim. yritys-wifi-verkon yli

Mikäli julkaisu tehdään App Storeen, käydään sovelluksen perustiedot kirjaamassa itunesconnect-palveluun, joka on App Storen "takahuone". Sitä

kautta kerrotaan sovelluksen kuvaukset, hintatiedot ja muut App Storessa näkyvät tiedot.

Sen lisäksi Applen kehittäjäportaalissa käydään generoimassa julkaisuavain, joka asennetaan kehityskoneelle. Sovellus käännetään valitsemalla tämä julkaisuavain käyttöön. Tämän jälkeen XCodesta valitaan arkistointitoiminto, joka paketoi sovelluksen julkaisua varten. Julkaisun lähetystoiminto tunnistaa automaattisesti itunesconnectista aktivoidun sovellusjulkaisun. XCode suorittaa ensin paikallisia tarkistuksia ja osaa kertoa, mikäli paketista puuttuu ikoneita, käynnistyskuva tms. perusvaatimuksia. Mikäli virheitä ei havaita, sovellus lähetetään App Storeen tarkastettavaksi.

Seuraavan vaiheen suorittaa Apple: Muutaman päivän kuluessa (vaihtelee käytännössä 4 - 10 päivän välillä) sovellus otetaan tarkastukseen. Sen toiminnallisuus tarkistetaan ja sovellusta testataan useilla eri laitteilla. Mikäli kaikki on kunnossa, sovellus julkaistaan App Storeen vuorokauden kuluessa tarkastukseen otosta.

Mikäli sovellus ei mene läpi tarkastuksesta, Apple tiedottaa sähköpostitse ongelman syystä. Kerran sovellukseni ei toiminut oikein iPhone 4:ssa: Sain heiltä hyvän bugiraportin kuvaruutukaappauksineen!

Kun sovellukseen lähetetään päivitysversioita, käydään iTunesconnectissa lisäämässä sovellukselle uusi versio jossa kuvataan, mikä on muuttunut. Tämän jälkeen XCodesta voidaan tehdä uusi arkistopaketti joka lähetetään AppStoreen. Päivitykset menevät usein nopeammin tarkistukseen.

Ennätykseni on tarkastus 3 minuuttia lähetyksen jälkeen ja hyväksyntä 20 minuuttia myöhemmin.

iTunesConnect sisältää myös myynninseurannan. Se osaa piirtää trendigraafit sekä ilmaisten että maksullisten sovellusten latausmääristä maittain. Myös sovellusten arvosteluja voi lukea, samoin crash logeja eri käyttöjärjestelmäversioittain lajiteltuna.

Mobiilisovellus ja taustapalvelut

Hyvin usein mobiilipalvelut hakevat tietoja internetpalveluista. Mikäli datan lähteet eivät ole optimoitu mobiilikäyttöön, saattaa tietomassat kasvaa liian isoiksi: Mobiilidataa liikkuu liikaa ja sovelluksesta tulee hidas. Tällöin kannattaa rakentaa oma taustapalvelu. Se voi olla esimerkiksi proxypalvelin, joka noutaa tarvittavat tiedot netissä olevilta palvelimilta, tallentaa omaan välimuistiin ja antaa ne kompaktissa muodossa mobiiliclientille.

Hyvä esimerkki tällaisesta tarpeesta on tekemässäni Kyyti-sovelluksessa: Se on sovellus, joka näyttää mm. ratikoiden ja junien reaaliaikasijainnit aikataulut ja kulkuneuvojen poikkeustiedotteet. Ilman proxypalvelinta mobiiliclientin pitäisi lukea vähintään 5 eri palvelun tietoja, joista osa on XML:ää, osa JSONia, osassa kutsutapana webservice/soap, osassa REST jne. Usein palvelut palauttavat myös enemmän tietoa kuin mobiiliclientissa tarvitaan.

Nyt sovelluksesta lähtee vain yksi kutsu proxypalvelimelle, joka koostaa vain tarvittavat tiedot ko. käyttäjälle.

Viemällä muutakin toiminnallisuutta taustapalveluihin, saadaan kevennettyä mobiilisovellustoteutusta. Näin on helpompi toteuttaa sama sovellus monelle eri käyttöjärjestelmälle, kun varsinainen logiikka voidaan toteuttaa vain kerran taustapalvelimelle.

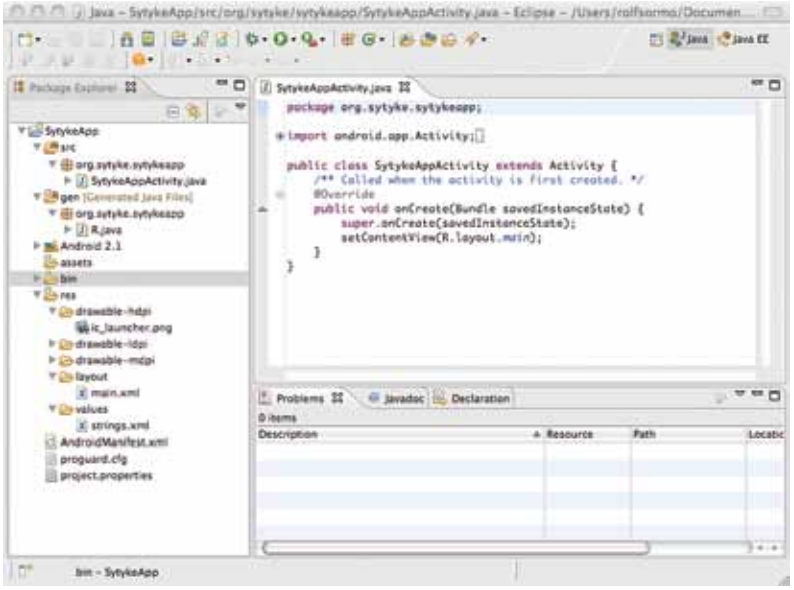
Taustapalvelimen toteutukseen Microsoftilla on tarjota Azure Toolkit for iPhone: Samainen palvelu löytyy myös Androidille ja Windows 7:lle. Se mahdollistaa mm. datan tallennuksen, liiketoimintalogiikan toteutuksen ja pushnotifikaatiot toteutettuna niin, että sama taustatoteutus palvelee montaa eri mobiilialustaa.

iOS-kehityslinkkivinkit:

Päiväkirja iPhone-koodausopiskelusta	http://www.qaiku.com/go/46i3/
Diigoon kerätty linkkilista iPhone-koodauksesta	http://groups.diigo.com/group/iphonecoding
Microsoftin Bing-kartta API iOSille	http://www.bing.com/community/site_blogs/b/maps/archive/2011/05/05/new-bing-maps-ios-sdk.aspx
Windows Azure toolkit for iPhone	http://watwp.codeplex.com/ https://github.com/microsoft-dpe/wa-toolkit-ios
Ilen blogissa lisää mm. iOS-koodauksesta	http://softaajarautaa.blogspot.com

Rolf Sormo
Kirjoittaja on toiminut teknologia-johtajana ja teknisenä arkkitehtina yrityksessä, joka tekee ketterää Android-kehitystä ja myy Android-laitteilla tehtäviä mobiilitiedonkeruurat-kaisuja.
[rolf.sormo@aphex.fi]

Kehitystyötä Androidilla



Android on vuonna 2007 julkistettu älypuhelimille ja tablet-laitteille tarkoitettu vapaan lähdekoodin käyttöjärjestelmä, laaja standardikirjasto, kokoelma mobiiliapplikaatioita sekä standardikehitysmalli.

Android-applikaatiot ohjelmoidaan Javalla. Java-luokat käännetään ensin .class -tiedostoiksi, joista ne applikaatiopaketin luonnin yhteydessä muunnetaan Androidin dx-työkalulla .dex -muotoon. Android-projekteissa voidaan käyttää suurinta osaa saatavilla olevista Javalle tehdyistä kirjastoista

Applikaatioihin on myös mahdollista liittää C:llä tai C++:lla ohjelmoituja osia, jos halutaan uudelleenkäyttää jo olemassa olevaa C/C++-koodipohjaa.

Kehitysympäristö

Android-kehitys on helpoimmillaan kun käytetään Eclipseä ja siihen sopivaa Android-pluginia.

Silloin ohjelmaa kehitetään nykypäivän kehitystyökalulla, ja käännön jälkeen ohjelma lähtee suoraan käyntiin joko Android-emulaattorissa tai suoraan tietokoneeseen USB-johdolla liitetyssä puhelimessa.

Android-kehitystä voi myös tehdä omilla tutuilla työkaluilla ja kutsua Android SDK:n mukana tulevia työkaluja komentoriviltä (tai oman IDE:n integraatioiden kautta), jos Eclipsellä kehittäminen ei ole vaihtoehto.

Projektin tärkeimmät osat

Android-projektissa on muutama tärkeä tiedosto joita joutuu projektin edetessä käsittelemään uudestaan ja uudestaan. Alla lista tärkeimmistä. Pienessä projektissa ei välttämättä muihin tarvitse koskea ollenkaan.

- **src/myapp/MyAppActivityjava**
  Jokaisessa applikaatiossa on vähintään yksi Activity-luokan toteutus, joka toteuttaa käsittelijämetodeja applikaation elinkaaren vaiheisiin. Yksinkertaisimmillaan luokka toteuttaa onCreate-metodin, luo siinä käyttöliittymän ja rekisteröi käsittelijät eri käyttöliittymäelementtien vuorovaikutuksille (esim. käyttäjä painaa nappia tai skrollaa listaa).
- **AndroidManifest.xml**
  Applikaatiossa on mukana manifesti, jossa listataan ja konfiguroidaan kaikki Activityt, listataan kaikki tarvittavat oikeudet, kerrotaan minimiversio Androidista, jolla applikaatio pyörii, sekä muita teknisiä tietoja järjestelmälle.
- **res/layout/main.xml**
  Androidin käyttöliittymät määritellään niin sanotuissa layout-tiedostoissa, joissa kuvaillaan käyttöliittymän rakenne ja eri elementtien attribuutit (mm. värit, suhteelliset koot, otsikot, ym.).
- **res/values/strings.xm**
  Androidin kehityskäytännön mukaan kaikki käyttöjärjestelmän tulostamat tekstit määritellään erikseen niin sanotussa strings-tiedostossa, ja niihin vain viitataan muualta. Tämä mahdollistaa samojen stringien uudelleenkäytön, sekä helpon tavan kääntää koko applikaatio eri kielille vain kirjoittamalla uusi strings.xml -tiedosto uudella kiellellä (esim. res/values-fi/strings.xml).

Android-kehityksen aloittaminen

1. Asenna Eclipse 3.5 tai uudempi, suositellaan Eclipse IDE for Java Developers-versiota
2. Asenna Android SDK
3. Asenna Android-plugin Eclipseen (EDT, tarvitsee polun SDK:hon)
4. Luo uusi Android-projekti (generoi automaattisesti Hello Worldin)
5. Koodaa

Android logo on Googlen luoma ja jakama ja sitä käytetään Creative Commons 3.0 Attribution lisenssin mukaisesti. Android on Googlen rekisteröimä tavaramerkki.

Käyttöliittymän skaalautuvuus

Android tukee erilaisia näyttöjä siten, että applikaatiota kehitettäessä grafiikoista ja muista käyttöliittymäelementeistä voidaan antaa erilaisia versioita. Ruudun pikselitiheyteen perustuvat vaihtoehdot ovat ldp eli low dpi, mdpi eli medium, hdpi eli high ja xhdpi eli extra high dpi.

Android versioon 3.2 asti käytetään ruudun pikselimäärään perustuvia määritelmiä, jotka ovat vastaavasti small, normal, large ja xlarge. Versiosta 3.2 lähtien määritelmiä voidaan antaa myös ruudun tiheyden mukaan korjattujen tiheysriippumattomien pikseleiden (density-independent pixel, eli dp) perusteella. Niillä saadaan tarkemmin määriteltyä rajat, joiden mukaan eri grafiikkoja ja käyttöiittymäelementtejä käytetään.

Oikeudet

Jokaisen Androidiin asennettavan paketin metatiedoissa kerrotaan tarvittavien käynnistystietojen lisäksi mitä oikeuksia (permissions) kyseinen applikaatio tarvitsee. Kun käyttäjä lataa applikaatiota Android Marketista, käyttäjälle näytetään selkokielisesti, mitä oikeuksia applikaatio on ilmoittanut tarvitsevansa. Tämän perusteella käyttäjä voi päättää olla asentamatta applikaatiota.

Jotta metatiedoissa olisi oikeaa dataa, Androidin tiettyjä tietoturvan kannalta oleellisia palveluita ei voi kutsua jos niitä ei ole määritelty metadataan. Jos applikaatio ei ole erikseen ilmoittanut tarvitsevansa INTERNET-oikeutta, ja yrittää lähettää tai vastaanottaa verkosta tavaraa, kyseinen kirjastokutsu palaa virheilmoituksella. Sama koskee lähes kaikkia muitakin palveluita - paitsi applikaation oman käyttöliittymän käsittelyä. Käyttäjän yhteystietojen käsittely, puhelimen konfiguraation muuttaminen, puheluiden soittaminen, tekstiviestien lähettäminen ja kaikkien sensoreiden tilan lukeminen ovat esimerkkejä näistä.

Android versiot

Android-projektia luodessa kehittäjän on valittava mille Android-versiolle applikaatio tehdään. Suositeltavaa on valita mahdollisimman pieni versio, jossa käytettäväksi aiotut rajapinnat ovat tuettuina. Androidin API:ssa on aina merkitty Androidin API:n versio, josta lähtien funktio on tuettu.

Joissain tapauksissa on myös mahdollista kääntää paketti uudempaa API:n versiota vasten, mutta silti merkitä manifestiin vielä vanhempi API:n versio. Tällöin koodissa pitää ennen uudemman API:n käyttämistä tarkistaa mikä Androidin versio laitteessa on, ja tarvittaessa olla kutsumatta uudempaa API:a. Tällöin voidaan tehdä Applikaatio niin, että se toimii uudemmissa laitteissa paremmin kun se voi käyttää uudempaa toiminnallisuutta.

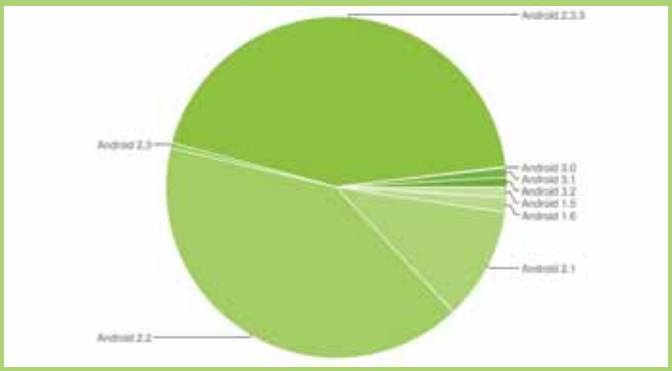
Tällä hetkellä 87% laitteista sisältää vähintään Android 2.2:n ("Froyo") tai uudemman, jolloin applikaation kannattaa tukea vähintään tätä versiota. Tukemalla Android 2.1:tä ("Eclair") applikaatio on asennettavissa jo 97.7%:iin laitteista.

Tiedot turvaan

Vaikka Android onkin alustana suhteellisen turvallinen, eikä applikaatiotuottajan tarvitse yleensä miettiä haittaohjelmia tai muuta tietoturvaa, niin kannattaa pitää mielessä, että noin yleisesti mobiililaitteisiin ei pidä luottaa. Huolimatta järjestelmien tarkasta suunnittelusta tietojen pitämiseksi salassa, järjestelmissä on aina heikkouksia, joita hyväksikäyttämällä tietoihin voi päästä käsiksi, ja salaiseksi tarkoitettu tieto voi päätyä vääriin käsiin. Ja koska kyse on mobiililaitteista, niihin on helpompi päästä käsiksi kuin vaikka yrityskannettaviin.

Mobiililaitteisiin ei siis nyrkkisääntönä pitäisi tallettaa mitään salaista tietoa, ainakaan luettavassa muodossa. Jos salaista tietoa halutaan pitää puhelimen välimuistissa, se pitäisi aina salata. Androidissa on tähän valmis palvelu standardikirjastossa, ja kyse on lähinnä asian tärkeyden ymmärtämisestä.

Android-versiot



Lähde:
http://developer.android.com/resources/dashboard/platform-versions.html

Mika Olkkola
Kirjoittajalla on vuosien kokemus tietotekniikka-alan johtotehtävistä. Tällä hetkellä hän työskentelee Microsoftilla sovelluskehittäjäyhteisöjen parissa.
[mika.okkola@microsoft.com]

Esitetyt mielipiteet ovat tekijän omia, eivätkä välttämättä edusta kirjoittajan työnantajan mielipiteitä.

Tulevaisuudessa menestyvän softan pitää myös näyttää ja tuntua hyvältä

Kuluttajistuminen on ilmiönä huomattavasti laajempi ja vaikuttavampi kuin useimmat olettavat. Kapeimmillaan se nähdään haluna tuoda oma taulutietokone työpaikalle. Vaikutus on kuitenkin paljon syvempi. Todellisuudessa kuluttajatottumukset tulevat mullistamaan tapamme käyttää tietotekniikkaa.

Kuluttajistuminen vaikuttaa ennen muuta työskentelytapoihin ja sitä kautta työvälineisiimme. Sosiaalisen median käytännöt ovat jo tulossa työpaikoille. Useissa yrityksissä on jo alettu rakentaa sisäisiä facebook-tyyppisiä ratkaisuja. Väitän, että tulemme näkemään myös apps-tyyppisten pääteohjelmien ja jopa pelillisten elementtien (gamification) lisääntymisen yritysohjelmistoissa.

Vaatimukset kasvavat

Päätelaitteiden kehittyminen tulee osaltaan vaikuttamaan tapaan, jolla jatkossa käytämme kaikkea teknologiaa. Kosketusnäytöt ja eri tarpeisiin soveltuvat kannettavat päätelaitteet ovat jo tätä päivää. Oletko huomannut, että kauppojen elektroniikkaosastoilla jokainen esillä oleva näyttölaite on täynnä sormenjälkiä, vaikka kyseessä ei olisikaan kosketusnäyttö? Kohteen koskettaminen on meille ihmisille luontevampaa kuin näppäimistön, hiiren tai vaikka kameran ohjausnäppäinten käyttö. Järjestelmäkamerassani ei ole kosketusnäyttöä mutta räplään sen näyttöä jatkuvasti – mutta turhaan. Olen vain niin tottunut kosketusnäytöllisten laitteiden käyttöön. Kosketusnäytöt ovat vasta alkua. Äänen ja liikkeen tunnistus ovat jo yleisiä pelisovelluksissa mutta tulevat ennen pitkää myös osaksi yrityssovelluksia.

Yritysten tietohallinnon suurin lähivuosien haaste on tuoda uusia vapausasteita tietotyöläisille. Vanhanaikainen ajattelu lukitusta ja hallitusta työasemasta ei toimi, kun työntekijät rikkovat rajoja tuomalla työpaikalle omia päätelaitteitaan. Ihmiset eivät koe pystyvänsä tekemään töitään riittävän tehokkaasti tietohallinnon tarjoamilla välineillä. Tilanne johtaa nopeasti hallitsemattomaan kaaokseen, jollei käyttäjien huutoon pystytä vastaamaan.

Kuluttajatottumukset ohjaavat myös työkäyttöä

Tietotekniikan kehityksen kärki on jo kauan sitten siirtynyt yrityksistä kotitalouksiin. Uudet teknologiainnovaatiot tehdään poikkeuksetta kuluttajille – ei yrityksille. Ilmiö ei ole uusi. Esimerkiksi CD-ROMit ja DVDt suunnattiin ensin kotitalouksille. Samoin pikaviestintä, IP-puhelut, blogit ja statuspäivitykset ovat olleet osa kuluttajien arkea jo pitkään. Nyt pikaviestintä on käytössä jo useissa organisaatioissa.

Monet muutkin sosiaalisen median käytännöt ovat hyödyllisiä työelämässä. Kuulin pari vuotta sitten väitteen, että Facebook on työelämän kannalta mer-

kittävintä, mitä on tapahtunut pitkään aikaan. Ei Facebook itsessään vaan ilmiö, jota se edustaa. En silloin täysin ymmärtänyt väitettä mutta nyt ymmärrän. Useat suomalaiset ovat omaksuneet Facebookin käytön ja tekevät sen avulla oman vapaa-aikansa läpinäkyväksi omalle ystäväpiirilleen. Facebookin kautta jaetaan tunnelmia, ajatuksia, kokemuksia ja sisältöjä. Miten tämä liittyy työelämään?

Työ muuttuu liikkuvammaksi ja olemme yhä harvemmin samassa paikassa samaan aikaan työtovereidemme kanssa. Miten pysymme perillä toistemme tekemisistä? Sähköposti ei ole vastaus haasteeseen.

Tietotyön seuraava kehitysvaihe, läsnätyö, perustuu suurelta osin oman työn tekemiseen läpinäkyväksi omalle yhteisölle. Meidän tulee tehdä oma työmme läpinäkyväksi samalla tavoin kuin teemme vapaa-aikamme läpinäkyväksi käyttäen tilapäivityksiä, tageja, ja jakamalla sisältöjä. Tämän päivän teknologia mahdollistaa sisäisen facebookin rakentamisen varsin helposti. Ihmisten hakeminen onnistuu muutenkin kuin nimellä – hakukriteerinä voi käyttää mm. kompetensseja, asiakassuhteita ja monia muita asioita. Tämä ominaisuus yhdistettynä ajantasaiseen läsnäolotietoon voi lyhentää asioiden käsittelyaikaa dramaattisesti. Samoin statuspäivityksillä voidaan kertoa ajankohtaisista työtehtävistä tai haasteista yhtälailla kuin sosiaalisessa mediassa kerrotaan viikonlopun aktiviteeteista. Foursquare on hyvä esimerkki nopeasti levinneistä sosiaalisen median välineistä. Voisiko sen edustamaa sosiaalista navigointia myös soveltaa työelämään?

Pelinomaiset tekniikat parantavat tuottavuutta

Pelillistäminen (gamification) tarkoittaa pelinomaisten tekniikoiden ja toimintojen tuomista ongelmanratkaisun tueksi. Tyypillisesti sitä käytettäisiin muiden sovellusten kuin varsinaisten pelien kehittämiseksi. Pelillistämisen tavoite on tehdä teknologian käyttö houkuttelevaksi, tukea haluttuja käytösmalleja ja käyttää hyväksi ihmisten luontaista mielenkiintoa pelaamiseen. Pelillistäminen mahdollistaa käyttökokemuksen tekemisen houkuttelevammaksi palkitsemalla osaamista ja kehittymistä. Teknologia voi auttaa esimerkiksi ikävien, toistuvien työtehtävien suorittamisessa. Tetris ja Angry Birds nousivat suosioon siksi, että sitä on helppo pelata ja siihen jää koukkuun. Pelillistäminen ja yhteisöllisyys voisivat tuoda näitä samoja elementtejä myös liiketoimintasovelluksiin – toki ensisijaisesti nuoremmille sukupolville. Jonkin ajan kuluttua yrityssovellustenkin on pakko koukuttaa käyttäjänsä positiivisella tavalla.

Hyvä design ja käytettävyys ovat välttämättömiä myös yritysohjelmistoissa. Miten tämä vaikuttaa yritysohjelmistoja tekevän sovelluskehittäjän työhön? Minua on pariinkin otteeseen haastettu siitä, että esimerkkisi mobiililaitteiden appsit eivät tuo mitään lisäarvoa yritysohjelmistojen käyttöön. Eivätkö todella? Muistan kuinka minua haastettiin samasta asiasta, kun aikanaan intoilin Windows 95 –version markkinoille tuloa. Olihan se ensimmäinen kunnollinen graafinen käyttöliittymä Intel-laitteille. Epäilevät Tuomaat eivät nähneet graafisella käyttöliittymällä mitään lisäarvoa yritysohjelmistojen käytössä. Niissä ohjelmissahan vain syötetään lukuja kenttiin. Muuta ei tarvitse kun tabin ja enterin. Kuinkahan moni käyttäjä olisi valmis vaihtamaan nykyisen pääteohjelmansa merkkipohjaiseen versioon?

On ymmärrettävää, että loppuvuonna 1995 oli haastavaa nähdä se dramaattinen vaikutus, joka uudella Windows-versiolla tulisi olemaan PC-laitteiden käyttöön. Samalla tavalla nyt on vielä vaikea nähdä, mikä vaikutus uusien käyttöliittymä- ja päätelaitetekniikoiden kehittymisellä on esimerkiksi yrityssovellusten käyttöön.

Design ja käytettävyys tulevat jatkossa olemaan keskeisiä kilpailutekijöitä ohjelmistosuunnittelussa. Tietotyön tuottavuus on yksi suurimpia läntisten maiden haasteita kansainvälisessä kilpailussa. Tietotekniikan käytettävyys on yksi keskeisiä tekijöitä tuottavuuden parantamiseksi. Ohjelmistoteollisuudessa sähköisten markkinapaikkojen kautta jaeltavat mikro-ohjelmistot ovat uusi ja kiinnostava mahdollisuus. Näillä markkinoilla design on suurin myyntivaltti ja käytettävyys kaikki kaikessa asiakasuskollisuuden saavuttamiseksi. Tästä eteenpäin menestyvän softan pitää myös näyttää ja tuntua hyvältä.

Timo Jokela

KäytettävyysOSY uusiutuu

Käytettävyyttä halutaan mutta ei osata vaatia.

KäytettävyysOSY:n toiminta keskittyy käytettävyyteen järjestelmäkehityksessä.

KäytettävyysOSY:n toiminta on tarkoitus profiloitua erityisesti tietojärjestelmien käytettävyys- ja käyttäjäkokemusasioihin (vs. tuotteiden käytettävyys). Käytettävyyssuunnittelun periaatteet ja menetelmät ovat geneerisiä, ja sitä kautta KäytettävyysOSY:n fokus kattaa kyllä yleisemminkin kiinnostavat käytettävyysasiat.

Mutta tietojärjestelmien käytettävyyteen liittyy piirteitä, jotka näyttävät tekevän siitä haastavamman kuin mitä käytettävyys on tuotekehitysympäristössä. Tämä näkyy siinä, että tietojärjestelmien käytettävyys on monesti heikommalla tasolla kuin tuotteiden.

KäytettävyysOSY järjesti seminaarin maailman käytettävyyspäivänä 10.11. (joka vuonna marraskuun toinen torstai). Seminaarin ingressissä käytettävyyden haaste tietojärjestelmäkehityksessä ilmaistiin seuraavasti : Nykyiset älypuhelimet, jotka on kehitetty miljoonille erilaisille käyttäjillä ovat radikaalisti helpompia käyttää kuin tietojärjestelmät, jodien käyttäjäkunta on yleensä suppea ja hyvin tunnistettu. Miten tämä on mahdollista? Tilannehan pitäisi olla päinvastoin - käytettävyyden suunnittelun keskeinen lähtökohta kun on ”tunne käyttäjäsi”.

Kun tarkastelee ongelman syytä, niin yhdeksi keskeiseksi nousee se, että järjestelmäkehityksessä on kaksi eri organisaatiota edustavaa osapuolta, hankkija ja ohjelmistoyritys. Tämä osapuolten ero vielä korostuu, kun hankkijana on julkinen organisaatio. Tällöin hankkijan on valittava ohjelmistoyritys julkisella kilpailutuksella.

Kilpailutuksessa keskeisessä asemassa on tarjouspyyntö – senhän perusteella ohjelmistoyritys pyrkii laatimaan mahdollisimman kilpailukykyisen tarjouksen. Vuonna 2010 julkaistussa tutkimuksessa (Lehtonen& 2010) ei löytynyt yhtään sellaista tarjouspyyntöä, jossa käytettävyys olisi ollut aito vaatimus tai valintakriteeri. Tätä kautta katsottuna ei ole ihme, että järjestelmäkehityksessä käytettävyys laahaa perässä.

Toisaalta kyseisessä tutkimuksessa kyllä havaittiin useimmissa tarjouspyynnöissä pyrkimystä käytettävyyden vaatimiseen – mutta asiassa siis ei onnistuttu aidosti. Siten hankkijat ehkä uskovat ja toivovat, että ohjelmistoyrityksillä olisi vastuu käytettävyydestä, vaikka tarjouspyynnöissä sitä ei käytännössä siis edellytetä. Käytettävyysasioiden kehittäminen tarjouspyyntöihin onkin yksi keskeinen kehittämiskohde.

Seminaari 10.11.2011

KäytettävyysOSYn Haaga-Helian tiloissa Pasilassa 10.11. pidetyssä seminaarissa osallistujia oli 50 hengen tila kutakuinkin täysi. Aluksi KäytettävyysOSYn nykyinen puheenjohtaja Timo Jokela (Joticon) kävi lyhyesti läpi juuri yllämainittua tarjouspyyntöjen ongelmaa. Anu Ylä-Pietilä (Trafi) edusti tilaajan näkökulmaa, kertoi tilaajan kohtaamista haasteista ja siitä, miten tärkeää olisi saada tunnistetuksi käytettävyyden vaikuttavuuden ja sitä kuvaavat mittarit. Mihael Cankar (IBM) ja Jouni Linkola (Tieto) olivat toimittajien edustajia. Mihael kertoi, miten ketterän kehittämisen voi mukauttaa niin, että käyttäjien tarpeet tulevat paremmin huomoiduksi. Jouni väitti, että tulisi korostaa, mitä mitattavia liiketoiminnallisia hyötyjä hyvä käyttökokemus mahdollistaa. Esimerkkinä Jouni esitti Tiedon sisäisen ERP järjestelmän, jonka kehitys aidosti käyttäjäystävälliseksi järjestelmäksi mahdollistui konkreettisen business casen kautta.

Seminaaria sävytti keskustelu ja ajatusten vaihto, niin esitysten lomassa kuin loppukeskustelun aikana.

Osaamisyhteisöt 2011

Sytykkeessä toimitaan niin yhdistystasolla kuin aihepiireittäin erikoistuneissa osaamisyhteisöissä. Monipuolisessa tarjonnassamme löytyy jokaiselle jotakin. Vaihtoehtona on myös perustaa omalle kiinnostukselleen uusi osaamisyhteisö – Sytyke-hallitus toivottaa toimintaehdotukset tervetulleeksi. Osaamisyhteisön toimintaan pääset mukaan laittamalla postia vetäjälle.

BisOSY
BisOSY on ohjelmistoliiketoiminnan osaamisyhteisö, jonka tavoitteena on parantaa suomalaista ohjelmistoliiketoimintaosaamista ja kiihdyttää alan kasvua ja kansainvälistymistä.
Vetäjänä Jyrki Kontio. [jyrki.kontio@iki.fi]

KäytettävyysOSY
KäytettävyysOSY tarjoaa jäsenilleen mahdollisuuden verkostoitua ja vaihtaa kokemuksia käytännössä toimiviksi osoittautuneista käytettävyyden kehittämiskeinoista sekä kehittää omaa käytettävyysosaamistaan. Vetäjänä ja yhteyshenkilönä Matti Vuori.
[xmvuori@kolumbus.fi]

MallinnusOSY
MallinnusOSYn tehtävä on yhdistää ohjelmistokehitykseen liittyvästä mallinnuksesta kiinnostuneet henkilöt keskustelemaan ja vaihtamaan ajatuksia sekä välittää tietoa aiheesta. DAMA (www.damafinland.org) on yhdistetty MallinnusOSYyn.
Vetäjänä toimii erikoistutkija Juha Pärssinen VTT:ltä.
[juha.parssinen@vtt.fi] (puh. 020 722 5612)

PrOSY
Projektitoiminnan osaamisyhteisö yhdistää systeemityöprojektien kehittämisestä kiinnostuneet. PrOSYyn on yhdistetty myös Ketterän kehityksen OSYn näkökulma. Vetäjänä Petteri Puurunen.
[petteri.puurunen@tieto.com]

RELA
on vapaamuotoinen, relaatiotietokantoihin keskittynyt osaamisyhteisö, joka kokoontuu 1-2 kuukauden välein. Tapaamisissa on aina pari alustajaa tai esittäjää, jotka ovat yleensä kerhon jäseniä.
Vetäjänä Lauri Pietarinen.
[lauri.pietarinen@relational-consulting.com]

SK SIG - Suomen suorituskykytestaajat
SK SIG Suomen suorituskykytestaajat on tietojärjestelmien suorituskykyyn keskittyvä TestausOsyn ja RELA:n rinnakkaisryhmä.
Vetäjänä toimii Ilkka Myllylä. [ilkka.myllyla@reaaliprosessi.com]

SOA SIG
SOA SIG (Special Interest Group) on palveluarkkitehtuuriin (Service-Oriented Architecture, SOA) keskittynyt osaamisyhteisö, jonka tarkoituksena on tarjota jäsenilleen mahdollisuuksia vaihtaa kokemuksia ja tietoa palvelulähtöisten liiketoiminta-arkkitehtuurien periaatteista, menetelmistä, välineistä ja sovelluksista.
Vetäjänä Janne J. Korhonen. [janne.korhonen@jannekorhonen.fi]

TestausOSY - FAST
TestausOSY on testauksen parissa työskentelevien yhteinen verkosto. Testauksen osaamisyhteisön tarkoituksena on edistää suomalaisen ohjelmistotestauksen ja ohjelmistotestaajien asemaa ja auttaa kaikkia testauksen parissa painiskelevia. Paikallistoimintaa on pääkaupunkiseudulla, Tampereella sekä Oulussa. Vetäjänä toimii Maaret Pyhäjärvi. [maaret.pyhajarvi@iki.fi]

KAOS
Kokonaisarkkitehtuurin osaamisyhteisö, joka pyrkii edistämään liiketoimintalähtöistä kokonaisarkkitehtuurinäkemystä suomalaisissa yrityksissä ja yhteisöissä, muodostamaan aktiivisen arkkitehtiverkoston sekä synnyttämään kokonaisarkkitehdin ammatti-identiteetin.
Vetäjänä toimii Anna Aaltonen. [anna.aaltonen@fortum.com]

Sytyttääkö? - Liity jäseneksi

Huomaa: *Liittyminen Sytykkeen jäseneksi maksaa vain 14€/vuosi, jos olet jo jonkin toisen TTL ry:n jäsenyhdistyksen jäsen! Jäsenenä saat muiden etujen lisäksi Systeemityö-lehden kotiin kannettuna neljästi vuodessa!*

Sytykkeen jäseneksi liitytään Tietotekniikan liiton kautta valitsemalla jäsenyhdistykseksi Systeemityöyhdistys ry. Nykyinen Tietotekniikan liiton jäsen voi liittyä joko vaihtamalla jäsenyhdistystä tai liittymällä lisäjäseneksi.

Tietotekniikan liiton henkilöjäsenmaksu vuonna 2012 on alkaen 55€ ilman lehtiä, vain Tietokone-lehti 74€. Erityisryhmien hinnoittelusta lisätietoja Tietotekniikan liitosta. Lisäjäsenyys maksaa 14€/yhdistys.

Sytyke ry on vuodesta 1979 toiminut valtakunnallinen systeemityöntekijöiden ammatillinen yhdistys, joka kehittää alan ammattilaisten välistä yhteistyötä ja tutkimustoimintaa.

Teemayhdistyksen jäseneksi voivat liittyä kaikki systeemityöstä kiinnostuneet yksityiset henkilöt, yhdistykset ja yritykset. Sytyke ry:n toiminta-alueena on koko Suomi. Sytyke on Tietotekniikan liitto ry:n jäsenyhdistys.

Lisätietoja Sytyke ry:stä: www.sytyke.org

TOIMISTO
Sytyke ry
Tietotekniikan liitto ry
Lars Sonckin kaari 12
02600 Espoo
vaihde: 020 741 9898
fax 020 741 9889

www.ttlry.fi
jasenasiat@ttlry.fi

KUUTAMOLLA

Ekosysteemityöläinen

Tätä kirjoitettaessa mobiiliekosysteemi on hypesanojen kärkipäässä. Sanon, että tätä kirjoitettaessa, koska lehden ilmestymiseen mennessä meillä voi hyvinkin olla jotain muuta hypeskeltävää. Ne meistä, jotka laskevat itsensä trendisettereiksi, nostavat tässä vaiheessa toista ylähuulen reunaansa ja mutisevat, että kyllä minä puhuin mobiiliekosysteemeistä jo niin ja niin monta viikkoa sitten, vai vielä hypetystä! No, minä kun olen enemmänkin trendipetturi, niin joudun toteamaan, kuten niin monet muutkin, että nyt kun sana on enempi tuttu, niin se tuntuu tulevan vastaan joka paikassa. Tai sitten hypetyksen kriittinen massa on saavutettu.

Ensimmäisenä sanasta 'ekosysteemi' mieleeni assosioituu viidakko. Mielleyhtymien syntymistä ovat monet tutkijat syvästikin pohtineet, mutta minun tiedossani ei ole, miksi juuri viidakko yhdistyy aivoissani ekosysteemiin. Miksei yhtä hyvin savanni, tundra, lehtometsä, meri tai suo? Suo voisikin olla oiva assosiaatiovaihtoehto mobiiliekosysteemille. Mutta ei siitä nyt sen enempää, pysytään viidakossa.

Kuten kaikki Tarzan-kirjoja lukeneet tietävät, viidakoissa on paljon puita, puskia, liaaneja, apinoita ja norsuja. Lisäksi siellä on hämähäkkejä ja matoja ja krokotiilejä. Tämän lyhyen kuvauksen jälkeen voimmekin miettiä viidakon ja mobiiliekosysteemin yhteneväisyyksiä. En sano, ketkä ovat apinoita, koska en halua koodareiden vihoja päälleni.

Jokaisessa viidakossa on tietysti omanlaisensa ravintoketju. Ylimpänä keikkuu joko Tarzan tai sitten erinäinen joukko kiiluvasilmäisiä kissapetoja. Mobiiliekosysteemissä tunkion kukkona ainakin mobiilipäätelaitevalmistajien markkinointiviestinnän mukaan kiekuu asiakas, sinä. Joidenkin asiakkaiden mielestä ravintoketjun huipulla on laitevalmistajayrityksen johtohahmo, levätköön tai eläköot rauhassa, kuka sitten liekin. Joka tapauksessa, viidakossa vallitsee kissanviiksenherkkä tasapaino saalistajien ja saalistettavien välillä. Mikä sinänsä on aika ihmeellistä, kun siellä kuitenkin kaikki melkein ovat niin saaliita kuin saalistajiakin, ainakin jossain mielessä. Orkideat kasvavat puunkoloissa loisina, kertoo Wikipedia. Nekin päätyvät jossain vaiheessa jonkun suuhun. Samoin kuin leoparditkin, jotka siis nekin saattavat kasvaa puunkoloissa tai ainakin yöpyä tai vetää päivänokosia raikkaan tapiiriaterian päälle. Leopardeja aika harva otus saalistaa, mutta eiköhän niilläkin jotkut loisensa ja viruksensa ole. Eikä kannata unohtaa uutta lumileopardiakaan (snow leopard), heh heh. Mobiiliekosysteemissä saalistaminen on harvinaisempaa. Joskus joku ostaa jonkun.

Ennen vanhaan – silloin hyvinä aikoina, jolloin nostalgiakin oli parempaa – it-maailma voitiin jakaa softaa ja rautaan. Nyt kun puhutaan mobiiliekosysteemeistä, niin sekaan sotketaan käyttäjät ja jos jonkinlaista palveluntuottajaa, operaattoria ja muuta hiipparia. Kun puhutaan ekosysteemien sodasta muistuu mieleen tietysti ensimmäisenä Skynet, mutta myös seuraavia termejä: OS/2, CP/M, MS-DOS, PC-DOS, GEOS, VMS, ULTRIX, RSX, RSTS, TOPS-10, TOPS-20, Multics, Guardian. Monet näistä käyttöjärjestelmistä on saalistettu sukupuuttoon, matkapuhelinten käyttöjärjestelmien osalta homma on vielä vähän kesken. Uusia tulee, vanhoja poistuu, yleismaailmallinen ilmiö.

Jaa niin otsikko. Hmm. Jos katsotaan uudelleen tuota edellisen kappaleen kirjainlitaniaa. Tai mietitään muutamia mobiilikehitysalustoja. Kaikki nuo kirjainhässäkät ja aikoinaan seksikkäät nimet ovat vaatineet kovan luokan osaamista monissa eri tehtävissä. Sitten, kun kirjainhässäkkä on kuopattu joko saalistuksen tai nääntymisen vuoksi, niin osaajilla onkin pääomanaan ex-osaamista ja exien kanssa on tunnetusti vaikea tulla toimeen. Toimeentulon varmistamiseksi pitääkin sitten hankkia uutta osaamista. Eikä ole ollenkaan moraalitonta hankkia sitä uutta osaamista jo silloin, kun se tuleva ex on vielä ihan voimissaan. Arjen ekosysteemissä kannattaa olla joustava ja varmistaa selusta. Aina kannattaa varautua pahimpaan ja yritä pysyä saalistajien puolella. Viidakossakin täytyy varautua esimerkiksi vihaisten lintujen hyökkäykseen. Viimeistään näin jouluna.

Oikein Hyvää Joulua ja Menestyksellistä Uutta Vuotta Kuutamolta!

Sytyke Ry:n Hallitus 2012

Hallitukseen kuuluu puheenjohtajan lisäksi 6 varsinaista jäsentä ja 2 varajäsentä.

Sytyke ry yhdistää suomalaiset ohjelmistokehityksen asiantuntijat. Yhdistyksen muodostavat erikoisalueisiin keskittyvät osaamisyhteisöt, joiden jäsenet välittävät toisilleen omien alojensa syvällisintä tietoa ja tuoreimpia trendejä.
Sytyke ry:n toiminta tapahtuu pääosin osaamisyhteisöissä, joiden tavoitteena on auttaa jäseniään kehittymään yhä vahvemmiksi omien alojensa ammattilaisiksi.
Sytyke ry on Tietotekniikan liiton (TTL ry) suurin valtakunnallinen teemayhdistys.

Hallitus 2012

Mikko Holmberg, puheenjohtaja
CodeBakers Oy
Metsänneidonkuja 6, 02130 ESPOO
Puh. 040 741 3991
mikko.holmberg@codebakers.fi

Anne Isotalo, hallituksen sihteeri
Fennia
Televisiokatu 1, 00240 Helsinki
Puh. 050 320 3899
anne.isotalo@gmail.com

Tarmo Toikkanen, varapuheenjohtaja
NP-Ratkaisut Oy, Aalto-yliopisto,
T:mi Tarmo Toikkanen
PL31000, 00076 AALTO
tarmo@iki.fi

Marianne Malila
Senior Consultant
Somea Oy
malila68@gmail.com

Minna Oksanen, lehden päätoimittaja
Senior Consultant Avarea Oy
minna.oksanen@gmail.com

Juho Kauppi, seminaarivastaava
Somea Oy
Puh. 040 846 4610
juho.kauppi@iki.fi

Sytyke ry:n hallituksen sähköpostilista: hallitus@sytyke.ttlry.fi

Eija Kalliala
HAAGA-HELIA Ammattikorkeakoulu
eija.kalliala@haaga-helia.fi

Varajäsenet

Altti Lagstedt, varajäsen
HAAGA-HELIA ammattikorkeakoulu
Ratapihantie 13, 00520 Helsinki
Puh. 040 488 7272
altti.lagstedt@haaga-helia.fi

Juha Jääskinen, varajäsen
Tuotepäällikkö Digia Oyj
Hiomotie 19, 00380 Helsinki
Puh. 050 4200 822
juha.jaaskinen@digia.com

Liittokokousedustajat 2012

Minna Oksanen
minna.oksanen@gmail.com

Mikko Holmberg
mikko.holmberg@codebakers.fi

Lauri Laitinen
lauri.laitinen@nokia.com

Seuraava lehti ilmestyy 7.3.2012!

Tehosta liiketoimintaa!

Haluatko tehostaa liiketoimintaasi tietoteknisten ratkaisujen avulla?

CodeBakers Oy tarjoaa palveluita erityyppisten järjestelmien Integraatioiden toteuttamiseen sekä sähköisten palvelujen ratkaisujen luomiseen.

Kun haluat laadukkaan ratkaisun, jonka avulla parannat yrityksesi tuottavuutta ja toiminnan laatua, ota meihin yhteyttä.

Codebakers Oy:

- **96% asiakkaistamme suosittelee meitä**
- **asiakaskuntaamme kuuluu joukko tunnetuimpia suomalaisia yrityksiä**
- **olemme Suomen nopeimmin kasvavien ohjelmistoyritysten joukossa**
- **liikevaihtomme 2010 3,7 M€, 47 paakaria**
- **johdon ja työntekijöiden omistama**
- **sijaitsemme Espoon Tapiolassa**

Lisätietoja:

Seppo Takanen
CodeBakers Oy
050-5810140
seppo.takanen@codebakers.fi

CodeBakers