

Suomen Kuntaliitto
Sairaalapalvelut

TERVEYDENHUOLLON XXVI ATK-PÄIVÄT
29. - 30.5.2000

**Komponenttipohjainen ohjelmistotuotanto
terveydenhuollossa:**

**Tavoitearkkitehtuuri vuonna 2005 ja
siirtymäpolku Musti-tyyppisistä
erillisjärjestelmistä**

**tutkija Juha Mykkänen,
Kuopion yliopisto**

**Oikean Rytmän
Pöytä**

Komponenttipohjainen

sovellustuotanto terveydenhuollossa

***-tavoitearkkitehtuuri 2005 ja
siirtymä Musti-tyyppisistä
järjestelmistä***

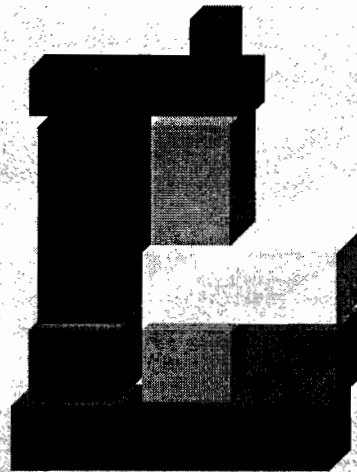
Terveydenhuollon XXVI atk-päivät

Pori 29.5.2000

Juha Mykkänen, tutkija

Kuopion yliopisto, atk-keskus

<http://www.uku.fi/atkk/fixit/comp/thatk2000.pdf>



Sisältö

- **Komponenttipohjainen sovelluskehitys
komponentit, niiden mahdollisuudet ja keinot**
- **Viitearkkitehtuuri, siirtymäpolku ja
tavoitearkkitehtuuri**
- **Sovellusintegraatio ja teknisiä kysymyksiä**

Ohjelmistotuotannon kehitys

1. keskuskesovellukset

tietokantaintegraatio,
päätekäyttö, keskitetty hallinta

2. työasemasovellukset

asiakas-palvelin-arkkitehtuuri
pc-sovellukset

3. www-sovellukset

kolmitasoarkkitehtuuri, thin client

4. komponenttisovellukset

e-business, "e-something",
monitasoarkkitehtuurit

I rakenteinen ohjelmointi

II oliomenetelmät

III hajautetut järjestelmät

IV komponenttijärjestelmät

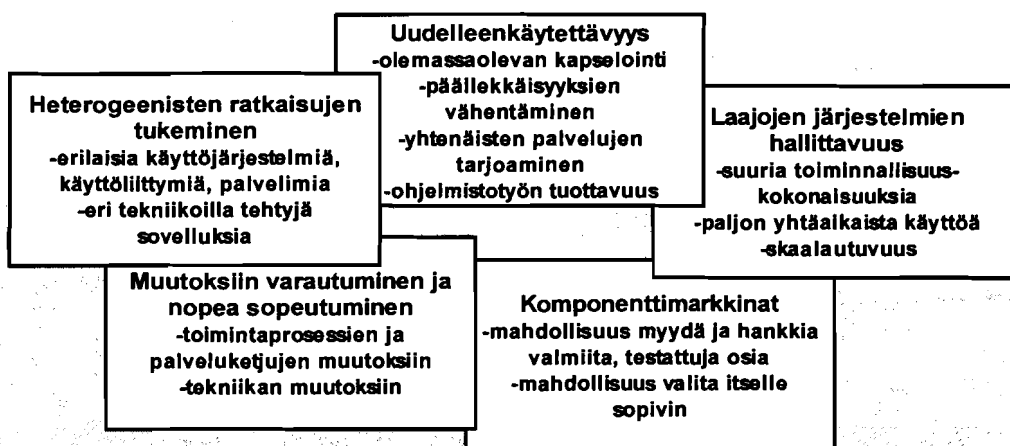
Komponentti

- itsenäinen ohjelmistorakenne "rakennuspalikka"
- suorituksen aikaiset liittymät (interface) tarjoavat palveluita ja piilottavat sisäisen toteutuksen käyttäjältä
- itsenäisesti jaeltavissa
- rakennetaan tuntien suoritusaluista etukäteen
- rakennetaan yhteistoimivaksi muiden komponenttien kanssa

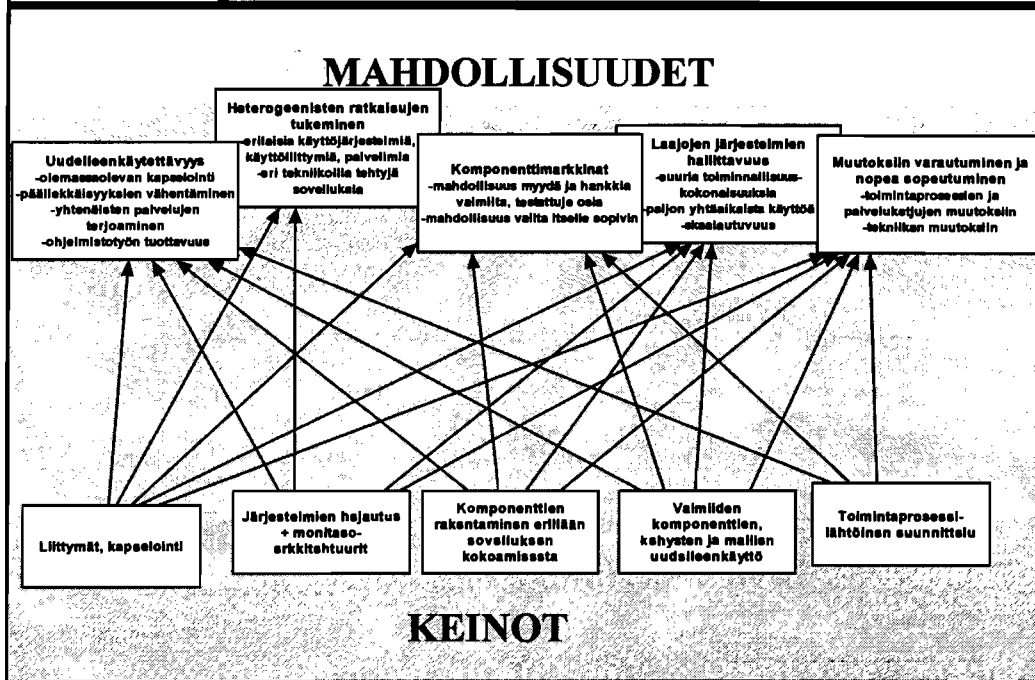
Komponenttipohjainen ohjelmistotuotanto

- Component-Based Development (CBD)
- järjestelmien rakentaminen **kokoamalla** olemassa olevista komponenteista
- ohjelmiston arkkitehtuuri, suunnittelu, toteutus, jakelu, tekninen infrastruktuuri jne. perustuvat uudelleenkäytettäviin komponentteihin
- olemassaolevien järjestelmien, infrastruktuurin, mallien ja kehysten käyttö
- tekninen ja toiminnallinen standardointi

Komponenttitekniikan mahdollisuudet



Komponenttitekniikan keinot



Kuinka mahdollisuuksista tehdään saavutuksia??

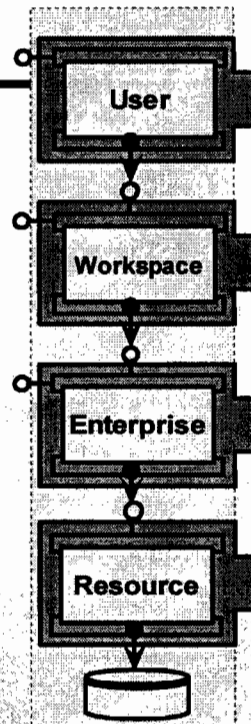
- arkkitehtuuri ja uudelleenkäytettävyys keskeisiä
 - toimialan jakaminen ”itsenäisiin” osiin
 - komponentti käytettävissä erillään alkuperäisestä tarkoituksesta
- tietojärjestelmän luonne muuttuu
 - uusi tapa myydä ja ostaa järjestelmäratkaisuja
- osien itsenäinen kehittäminen ja jakelu
 - muutosten vaikutusten rajaaminen, määritellyt riippuvuudet
 - asteittainen kehittäminen, siirtyä uusien tekniikoiden käyttöön
- ohjelmistotuotantoprosessi muuttuu
 - toimintaprosessit sovellusprojektien lähtökohdaksi
 - palveluarkkitehtuurin käyttö tietoarkkitehtuurin sijaan
 - komponenttivarastojen tarve
 - erillinen komponenttien hankinta- ja hallintaprosessi
- tarvitaan sekä teknisiä että toiminnallisia standardeja ja sopimista
 - ei samaan muottiin pakottamista: samaa palvelua voi käyttää eri tavoin

Komponentti-FixIT -projekti

- mukana ohjelmistoyrityksiä, yliopistosairaloita ja tietokanta- ja infrastruktuuritoimittajia
- tavoitteita:
 - tuoteriippumaton puitearkkitehtuuri (sovelluksen tavoitearkkitehtuuri 2005)
 - siirtymäpolku Musti-pohjaisista järjestelmistä
 - välineet ja teknologia siirtymäpolun alkuun
 - sovellusten perustuminen uudelleenkäytettäviin toimialakomponentteihin
 - tuki oliopohjaiselle sovellusintegraatiolle (HL7v.3)
 - tuki aluearkkitehtuurille (Makropilotti)
 - välineistön koulutus- ja käyttöönottoprosessi
- 1. vaihe 2000-2001, tavoitearkkitehtuurin määrittely ja siirtymäpolun ensimmäisten vaiheiden toteutus

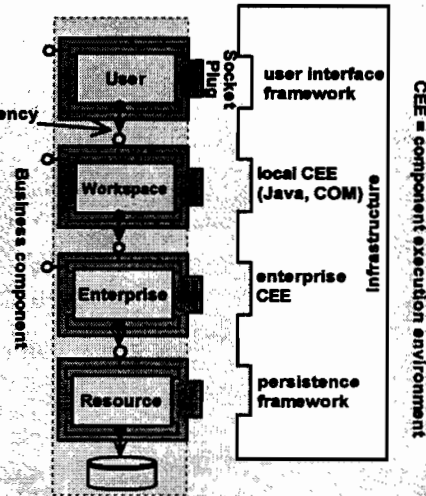
Komponenttipohjainen viitearkkitehtuuri

- Nelitasoarkkitehtuuri - kerrosten vastuut
 - käyttäjäkerros (user tier)
 - käyttöliittymä
 - edustajakerros (workspace tier)
 - yhden käyttäjän työtila, paikallinen logiikka
 - toimintakerros (enterprise tier)
 - yleinen toimintalogiikka, transaktiot jne.
 - resurssikerros (resource tier)
 - tietovarastot, yleiset palvelut
- Tekninen näkymä: kerrokset toteutetaan komponenttien avulla
- Toiminnallinen näkymä: komponentti on kerrosten välinen yhdistävä tekijä (toimialakomponentti)

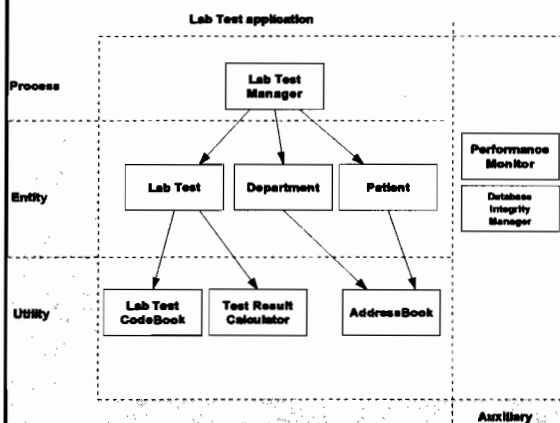


Toimialakomponentti (Business component)

- korkean tason komponentti:
 - edustaa todellista käsitettä tai prosessia (potilas, lähetteen kirjaus)
- sisältää yhden tai useamman kerroksen
 - kerrosten välinen yhdistävä tekijä
 - kerrokset toteutettu teknisten komponenttien avulla
 - tekniset komponentit voivat muodostua esim. ohjelmointikielten luokista tai kapseloida vanhan järjestelmän
- itsenäinen yksikkö sekä suunnittelussa, toteuttamisessa että jakelussa
- riippuvuudet muista komponenteista ja suoritusalustasta määritelty



Toimialakomponenttijärjestelmä

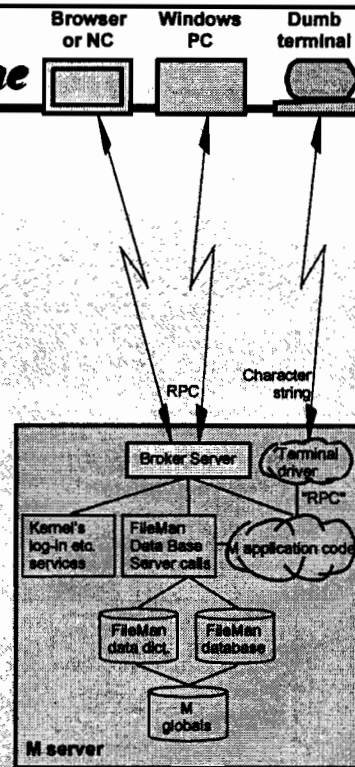


- toiminnallinen näkymä: kukin (toimiala)komponentti voi sisältää useita teknisiä komponentteja ja kerroksia
- ylimmän tason "komponentti" voi olla esim. eri komponentteja käyttävä (käyttöliittymä)-sovellus

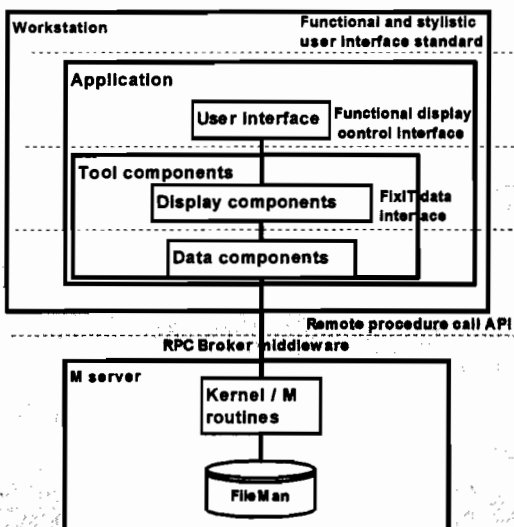
- **Roolit:**
 - process: koordinaatio, muiden komponenttien ohjaus
 - entity: pysyvä tieto ja sen kiinteä toimintalogiikka
 - utility: yleiskäyttöinen, esim. eri sovelluksissa tarvittava palvelu

Musti-järjestelmien nykytilanne

- M-pohjainen arkkitehtuuri
- FileMan-tietokanta
- päätekäyttö
- käyttöliittymien uusiminen mahdollista FixIT-välineillä
 - työasema/palvelin-arkkitehtuuri
- Windows- tai Java- käyttöliittymät
- järjestelmien integrointi toteutettu yhteisen tietokannan avulla tai sovelluskohtaisesti

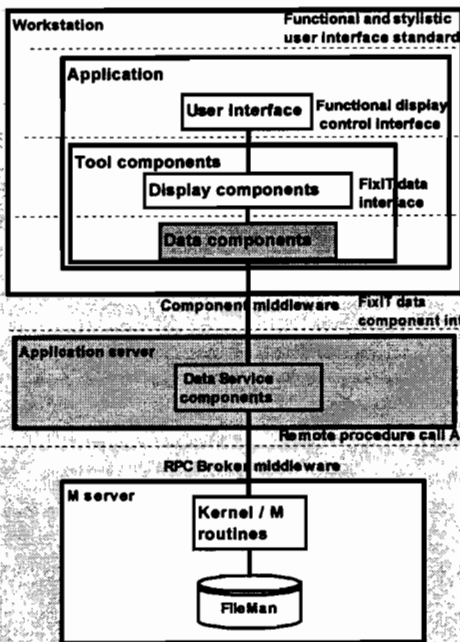


FixIT-nykyarkkitehtuuri



- välineistöt graafisten käyttöliittymien rakentamiseen
- 2-tasoarkkitehtuuri
- liittymätasot
- näyttökomponentit
- tietokomponentit
- M-palvelimen kutsuminen RPC Broker - väliohjelmiston avulla
- siirtymän aikana käytetään samoja välineitä soveltuvin osin kuin tähän asti

Siirtymäpolku vaihe 1: 2000-

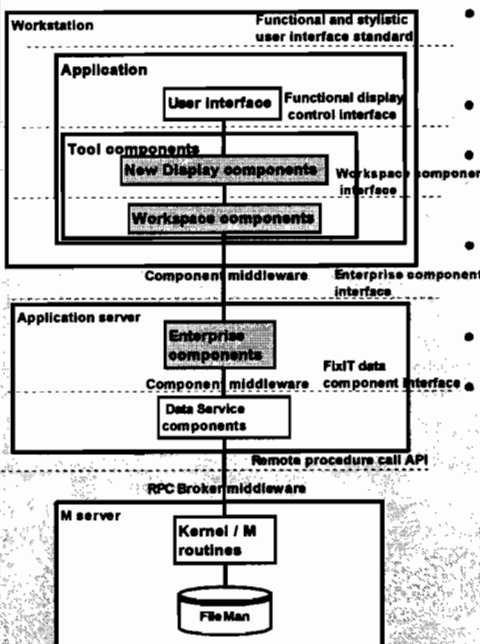


- siirtyminen monitasoarkkitehtuuriin (sovelluspalvelin)
- uudet tietokomponentit
- Broker-väliohjelmisto siirtyy työasemalta sovelluspalvelimelle

Hyödyt:

- keskitetty tietokantayhteyksien hallinta
- sisäänkirjoittautumismekanismien hallinta
- parempi toimivuus esim. palomuurin ulkopuolelta
- perusta jatkotoimenpiteille - alhaalta ylös rakentaminen

Siirtymäpolku vaihe 2: 2001-

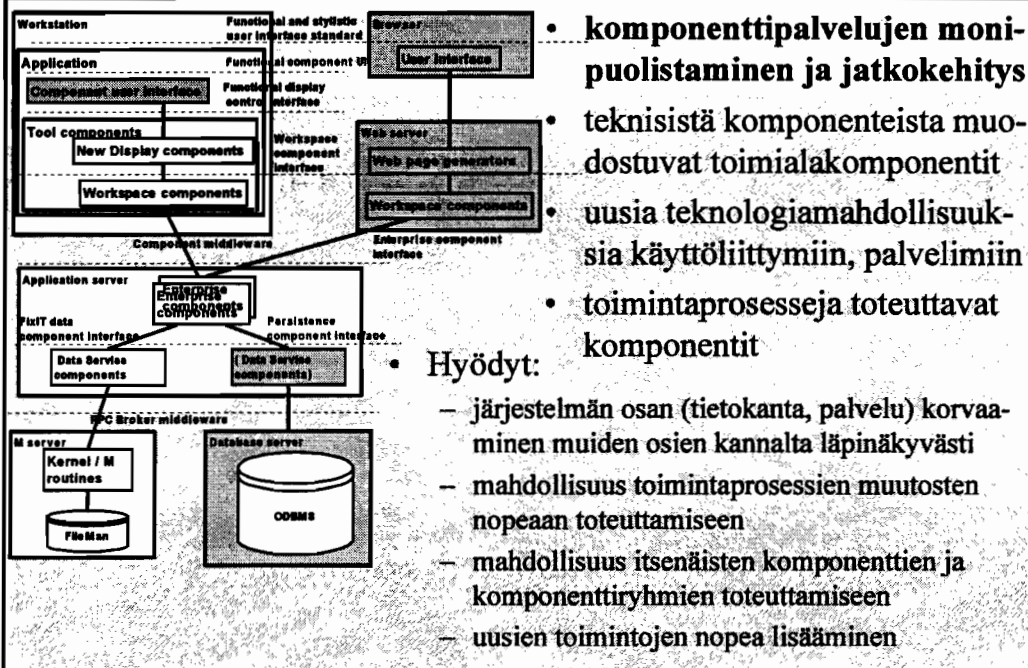


- siirtyminen komponentti-palvelujen käyttöön
- tekniset komponentit
- logiikan ja tiedon säilytystekniikan kapselointi sovelluspalvelimelle
- työasemalle palvelimen komponentteja käyttävät komponentit
- peruskehys entity-komponenteille

Hyödyt:

- uudelleenkäytettävyyden ja skaalautuvuuden paraneminen
- mahdollisuus vaihtaa tietokantatekniikkaa tai toimintalogiikan toteutusta vaikuttamatta käyttöliittymään
- mahdollisuus järjestelmien päällekkäisyyksien karsimiseen

Siirtymäpolku vaihe 3: 2002-2005

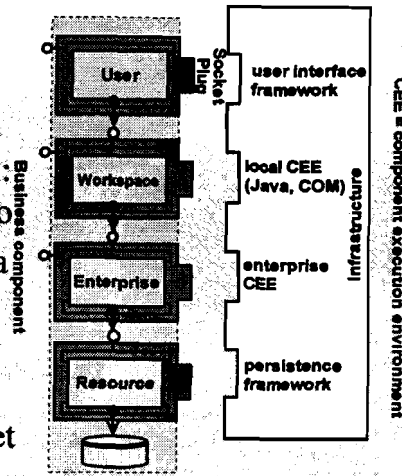


Sovellusintegraation näkökulma

- integraatiotyötä tehdään eri tasoilla:
 - sanomaintegraatio
 - HL7, XML
 - palvelupohjainen integraatio
 - CORBAMed
 - työpöytäintegraatio
 - CCOW / HL7, yhteinen kontrolleri
- komponentti voi tarjota liittymiään muihin järjestelmiin
- saman komponentin käyttö eri järjestelmissä vaatii yhteistä infrastruktuuria tai siltatoiminnallisuuden hankkimista
- vaihtoehtoisesti komponentti voi kapseloida myös olemassa olevan integraation toteutuksen tai esim. uuden sanomaintegraation, esim. XML-liittymä osana toimialakomponenttia
- korkean tason integraatiossa kuvattava myös semantiikka ja toiminnallinen malli

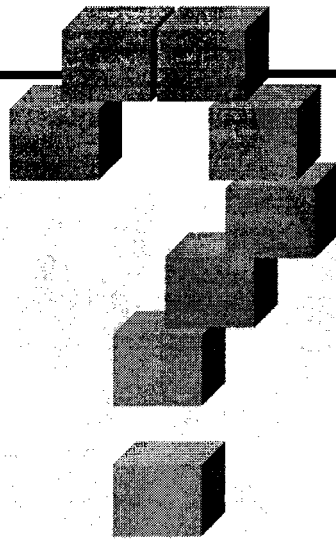
Teknisiä seikkoja

- CORBA, EJB ja COM+ tarjoavat vaihtoehtoiset mahdollisuudet komponenttien toteutukseen
- komponenttistandardit kehittyvät nopeasti: tärkeää valita toteutuksessa käytettävä taso
- komponenttistandardeja käyttäviä tuotteita ja palveluita runsaasti eri tarkoituksiin ja kehitys standardien kehitystä nopeampaa
- tärkeää määritellä ensin sovellusarkkitehtuurin vaatimukset ja valita sopivat tuotteet sen toteuttamiseen
- hyvä suoritusympäristö (CEE) tarjoaa ratkaisuja kommunikointiin, hajautukseen, siirrettävyyteen, transaktioiden käsittelyyn ja turvallisuuteen



Yhteenveto

- komponenttipohjaisen ohjelmistotuotannon etuja:
 - uudelleenkäyttö, järjestelmän jakaminen hallittaviin, loogisiin ja itsenäisiin osiin
 - suorituskyvyn kohdentaminen, skaalautuvuus
 - eri tyyppisten palveluiden nopea integrointi sovelluksiin
- arkkitehtuuri ja uudelleenkäytettävyys keskeisiä
- liittymien avoimuus yhteentoimivuuden perusta
- vanhat järjestelmät tulevat olemaan pitkään käytössä
 - asteittainen siirtymä uuden tekniikan käyttöön pala palalta, vanhat sovellukset käytössä siirtymän aikana kuten ennenkin
 - vanhan "kapselointi" mahdollistaa jo tehdyn työn hyväksikäytön
 - kaikkea ei korvata komponenteilla; hyödyt päällekkäisyyksien vähentämisestä, uusista ja tehostuneista toiminnoista ja uudelleenkäytöstä
- pelkät tekniikat ja välineet eivät riitä
 - ohjelmistojen tuottaminen, hankinta ja hallinta muuttuvat
- valmiina on paljon tarvittavaa, haasteena valinta ja yhdistely



Komponentti-FixIT -projekti <http://www.uku.fi/atkk/fixit/comp/>
Juha.Mykkanen@uku.fi