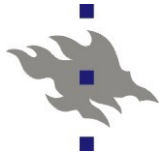




Palvelusuuntautunut ohjelmistotuotanto

Luento 2: Palvelut ja palvelukoosteet
Toni Ruokolainen, 22.1.2010



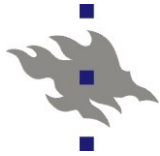
Luennon runko

- Palvelut
 - Palvelun määritelmä
 - Palveluiden ominaisuuksia ja suunnittelukriteerejä
- Palveluiden koostaminen
 - Palvelukoosteiden ominaisuuksia
 - Erilaisia koostamistapoja



Mitä tarkoittaa "palvelu"?

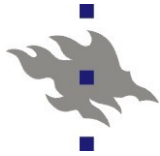




Mitä tarkoittaa "palvelu"?

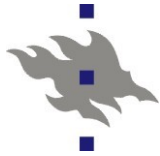
- Interaktiota asiakkaan ja palveluntuottajan välillä?
- Ulkoisesti havaittavaa käyttäytymistä?
- Palveluntarjoajan kyky tuottaa hyötyä asiakkaalle?
- Joukko käytettävissä olevia operaatioita?
- Järjestelmän (lisä-)ominaisuus?
- Yleisesti saatavissa oleva, verkon yli jaettava sovellus?





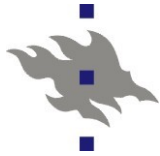
Palvelun määritelmä

- Palvelun käsite sisältää kaikki edelliset *näkökulmat*
- Palvelulla tarkoitamme tällä kurssilla *ohjelmisto(tuotannon)komponenttia*
 - *Joka täyttää jonkin käyttäjätarpeen*
 - *Jolla on hyvin määritelty rajapinta*
 - *Formaali, yksikäsitteinen, looginen*
 - *Joka on itsensä kuvaava*
 - *Palvelukuvaus saatavilla*
 - *On avoimesti saatavilla*
 - *Julkaistut rajapinnat, yleisesti käytössä olevien standardien hyödyntäminen, neuvoteltavissa olevat käyttöpolitiikat,...*
 - *Esimerkiksi Web Services -standardien käyttäminen*
- On palveluperustaisen analyysin, suunnittelun ja koostamisen yksikkö (*unit of (de)composition*)



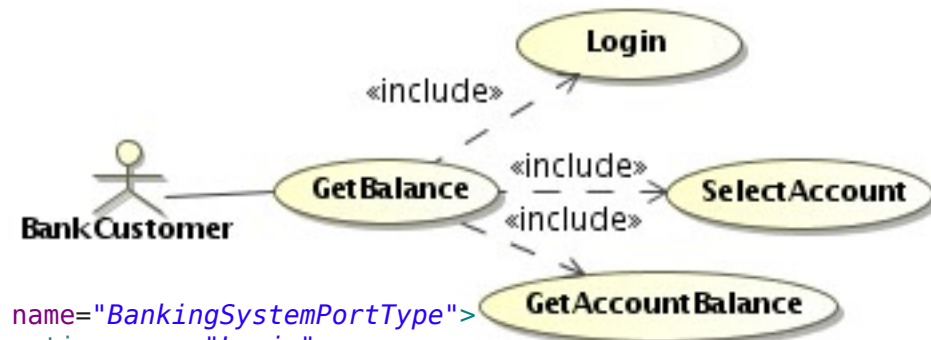
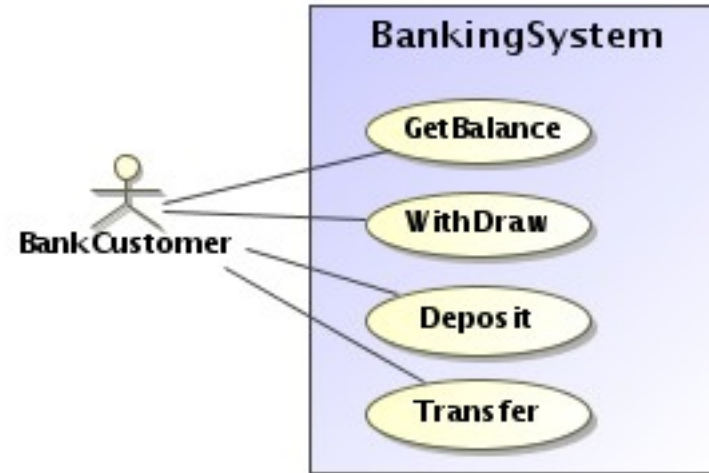
Palvelu ohjelmistokomponenttina

- Tällä hetkellä yleisin teknologinen kehys perustuu Web Services (*WS**) standardeihin
 - SOAP (*Simple Object Access Protocol*) kommunikointiprotokollana
 - WSDL (*Web Service Description Language*) kuvauskielenä
 - UDDI (*Universal Description, Discovery and Integration*) palvelurekisterinä
 - Ja paljon muuta..
- Mutta myös muita palveluperustaisin teknologioita
 - REST (*Representational State Transfer*)
 - Erityisesti ns. Web 2.0 sovelluksissa käytettävä lähestymistapa
 - J2EE (*Java-perustaiset yritysohjelmistot*), CORBA (*väliohjelmistoarkkitehtuuri*), OSGi (*Java-pohjainen plug-in arkkitehtuuri*),...

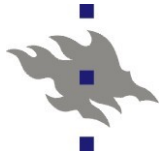


Palvelu täyttää käyttäjätarpeen

- "Käyttäjätarpeen" tulkinta riippuu käytettävästä abstraktiotasosta
- Palveluita esiintyy eri abstraktiotasoilla
 - Liiketoiminnallinen palvelu
 - Täyttää asiakkaan tarpeen
 - Komponenttipalvelu
 - Täyttää liiketoiminnallisen palvelun tarpeet
 - Teknologinen palvelu
 - Toteuttaa komponenttipalvelun toiminnallisuuden

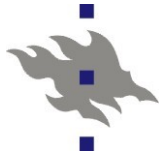


```
<wsdl:portType name="BankingSystemPortType">
  <wsdl:operation name="Login">
    <wsdl:input message="tns:LoginRequest"></wsdl:input>
    <wsdl:output message="tns:LoginResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SelectAccount">
    <wsdl:input message="tns:SelectAccountRequest"></wsdl:input>
    <wsdl:output message="tns:SelectAccountResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetBalance">
    <wsdl:input message="tns:GetBalanceRequest"></wsdl:input>
    <wsdl:output message="tns:GetBalanceResponse"></wsdl:output>
  </wsdl:operation>
</wsdl:portType>
```



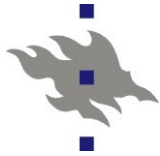
Hyvin määritelty palvelurajapinta

- Mitä palvelurajanpinnasta tulisi ilmetä?
 - Rakenne
 - Operaatioiden ja informaation rakenne
 - Semantiikka
 - Erityisen tärkeää palveluita etsittäessä
 - Palvelun laatu
 - Erityisen tärkeää palveluita valittaessa ja palvelusopimusten osana
- Hyvin määritelty?
 - Formaali
 - Koneen tulkittavissa
 - Yksikäsitteinen
 - Looginen kokonaisuus



Palvelurajapintojen yleisiä suunnitteluperiaatteita

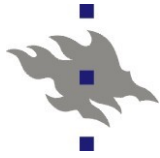
- Ortogonaalisuus (*orthogonality*)
 - Rajapinnan määrittelemät operaatiot eivät ole päällekkäisiä toiminnallisuudeltaan
- Täydellisyys (*completeness*)
 - Rajapinta määrittää toimialueen jonkin toiminnallisuuden kokonaisuudessaan
- Laajennettavuus
 - Esimerkiksi rajapintojen alityypitys
- Evoluution sieto
 - Rajapintojen versiointi
- Uudelleenkäytettävyys
 - Alhainen kytkentä
 - Korkea koheesio



Palvelurajapintojen kytkentä (*coupling*)

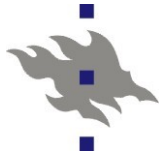
- Tavoitteena alhainen kytkentä eli riippuvuuksien minimointi
- Esimerkkejä riippuvuuksien luonteesta:

Facet	Tight Coupling	Loose Coupling
4.1 Discovery	Registration	Referral
4.2 Identification	Context-based	Global
4.3 Binding	Early	Late
4.4 Platform	Dependent	Independent
4.5 Interaction	Synchronous	Asynchronous
4.6 Interface Orientation	Horizontal	Vertical
4.7 Model	Shared Model	Self-Describing Messages
4.8 Granularity	Fine	Coarse
4.9 State	Shared, Stateful	Stateless
4.10 Evolution	Breaking	Compatible
4.11 Generated Code	Static	None/Dynamic
4.12 Conversation	Explicit	Reflective



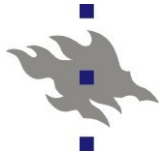
Palvelurajapintojen koheesio (*cohesion*)

- Tavoitteena korkea koheesio operaatioiden välillä
- Koheesioperiaatteita
 - Toiminnallinen koheesio
 - Rajapinnan operaatiot liittyvät tiettyyn tehtävään
 - Esimerkiksi "GetBalance"-käyttötapauksen operaatiot yhdessä rajapinnassa.
 - Kommunikointikoheesio
 - Rajapinnan operaatiot käsittelevät samoja informaatioelementtejä
 - Esimerkiksi käyttäjätietoja käsittelevät operaatiot yhdessä rajapinnassa.
 - Looginen koheesio
 - Rajapinnan operaatiot ovat toisistaan riippumattomia, mutta suorittavat samankaltaisia tehtäviä
 - Esimerkiksi eri maksumuodot samassa rajapinnassa.



Palvelukuvaus

- Palvelukuvaus määrittelee palvelun käyttötavan ja sen käyttöön liittyvät rajoitteet
- Käyttötavan kuvauksesta
 - Syntaksi: miten palvelua kutsutaan?
 - kommunikoitavan informaation struktuuri (esim. XML-kielet)
 - Operaatioiden struktuuri (nimet, syöte ja tuloste)
 - Semantiikka: mitä metodit ”oikeasti” tekevät?
 - Esimerkiksi *IOPE: input, output, precondition, effect*
 - Käyttäytymisen: miten palvelua käytetään?
- Rajoitteista
 - Käyttöpolitiikat: hinta, turvallisuus,...
 - Käyttörajoitukset: saatavuus (esim. ”arkipäivisin klo 8-16), käyttäjäryhmät ja roolit,...



Palvelukuvausesimerkki: WSDL

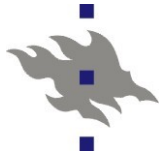
- Web Service Definition Language

- XML-pohjainen palvelurajapintojen kuvauskieli

- Tärkeimmät elementit:

- Porttityypit
- Tyyppi- ja viestimääritelmät
- Operaatiot
- Protokollasidonnat
 - Yleensä SOAP

```
<wsdl:portType name="BankingSystemPortType">
  <wsdl:operation name="Login">
    <wsdl:input message="tns:LoginRequest"></wsdl:input>
    <wsdl:output
message="tns:LoginResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SelectAccount">
    <wsdl:input
message="tns:SelectAccountRequest"></wsdl:input>
    <wsdl:output
message="tns:SelectAccountResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetBalance">
    <wsdl:input
message="tns:GetBalanceRequest"></wsdl:input>
    <wsdl:output
message="tns:GetBalanceResponse"></wsdl:output>
  </wsdl:operation>
</wsdl:portType>
```



Palvelukuvausesimerkki: WS-Policy

- XML-pohjainen kieli palvelun ei-toiminnallisten ominaisuuksien ja tarpeiden kuvaamiseksi

- Tärkeimmät elementit:

- Loogiset operaattorit *Exactly* ja *All*

- Poliittikkakielet

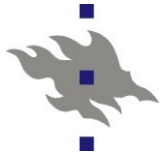
- WS-SecurityPolicy, WS-ReliableMessaging, WS-BusinessActivity,...

- Käytetään yhdessä WSDL:n kanssa

```
<wsp:Policy
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/
    securitypolicy"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/
    policy" >
  <wsp:ExactlyOne>
    <sp:Basic256Rsa15 />
    <sp:TripleDesRsa15 />
  </wsp:ExactlyOne>
</wsp:Policy>
```

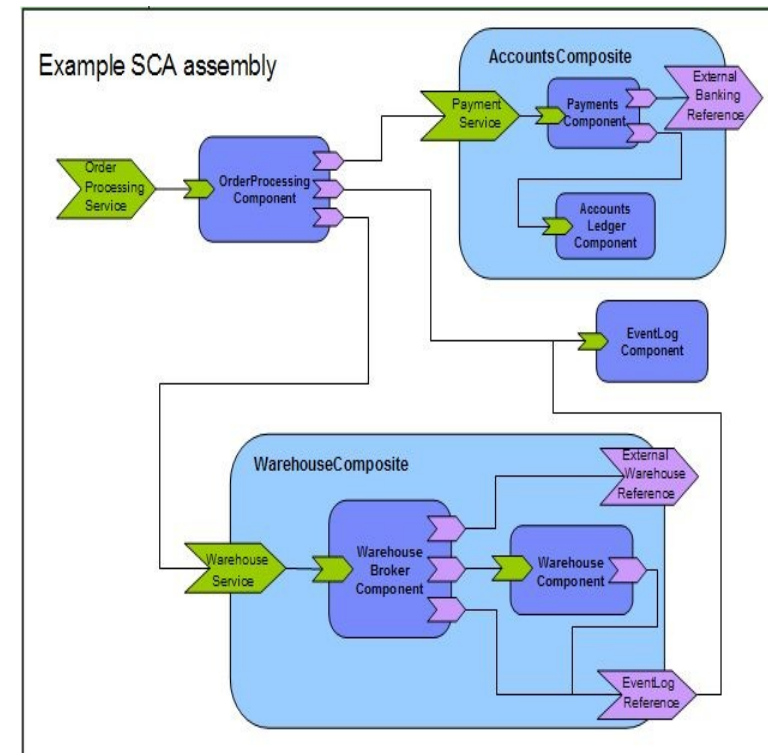


Tauko..



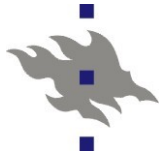
Palveluiden koostaminen (*service composition*)

- Palvelukooste
 - Täyttää jonkin yhteistyötarpeen
 - 2- n palvelun koordinoitua ennalta määriteltyjen ehtojen mukaan
- Koosteen koordinointi voi olla
 - hajautettua, tai
 - keskitettyä
- Koostamistapoja
 - Määrittely: Imperatiivinen vs. deklarativinen
 - Synteesi: Manuaalinen vs. automaattinen
 - Sidonta: Staattinen vs. dynaaminen



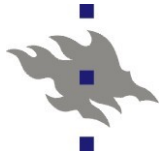
Kuva:

<http://www.osoa.org/display/Main/Relationship+between+SCA+and+BP>



Esimerkkejä koostamistekniikoista

- Prosessikielet
 - WS-CDL, WS-BPEL, Jolie, XPD, BPML,...
- Semanttinen koostaminen
 - Semantic Web Services (SWS), OWL-S
 - Käytetään logiikkaa, päättelyä ja suunnittelua
- Formaalit menetelmät
 - Äärelliset automaatit, prosessialgebrat, Petriverkot,...
 - Perustuu palveluiden käyttäytymisen formaaliin mallintamiseen
- Malliperustaiset menetelmät
 - BPMN, UML diagrammit
 - UML:stä erityisesti aktiviteetti ja sekvenssidiagrammit
 - UML profilointimekanismi
- Ja muutama muu..



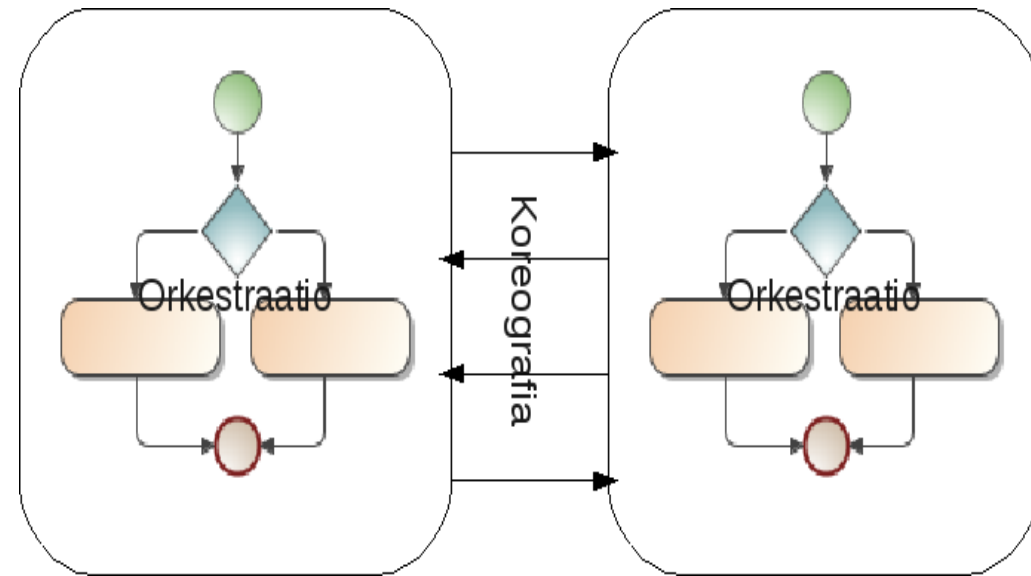
Orkestraatiot ja koreografiat

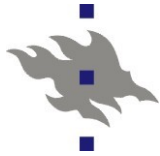
■ Koreografia

- Palvelukooste, jolla hajautettu koordinointi
- Täyttää jonkin yhteistoiminnallisen tarpeen
- Määrittelee yhteistyöverkoston roolit ja niiden väliset interaktiot
 - Globaali lintuperspektiivi

■ Orkestraatio

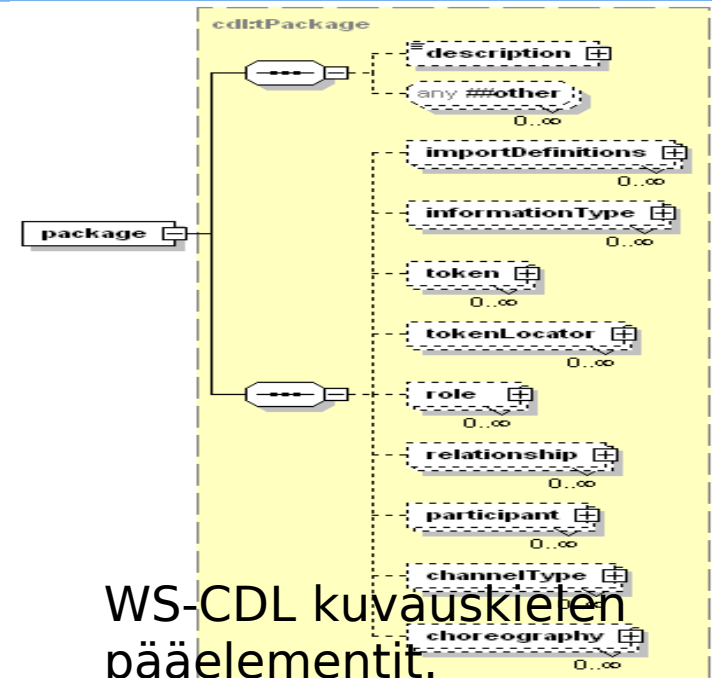
- Palvelukooste, jolla keskitetty koordinointi
- Täyttää jonkin yhteistoimintaverkoston roolin vaatimukset



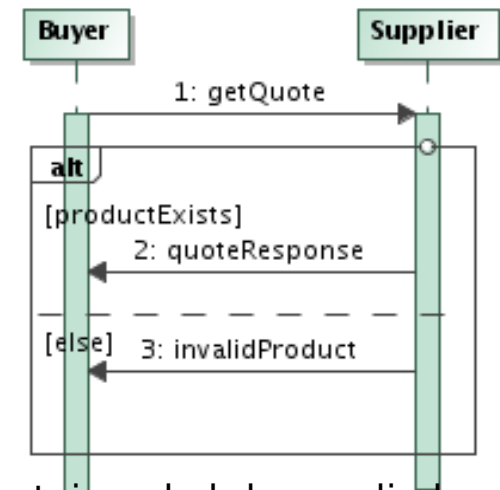


Koreografian kuvaaminen prosessina

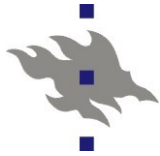
- Koreografiaprosessi kuvaa
 - Joukon rooleja (2..*)
 - Niiden väliset interaktiot
 - Interaktioiden informaation sisällön
 - Mahdollisesti muitakin ominaisuuksia
 - Kommunikaatiokanavien ominaisuuksia, liiketoimintaan liittyviä yksityiskohtia, ...
- Esimerkkejä mallinnustavoista
 - BPMN
 - UML sekvenssidiagrammit
 - WS-CDL
- Käyttökohteita koreografioille
 - Palveluiden suunnittelu
 - Yhteistyösopimuksen osana
 - Orkestrointien tuottaminen



WS-CDL kuvauskielen pääelementit.

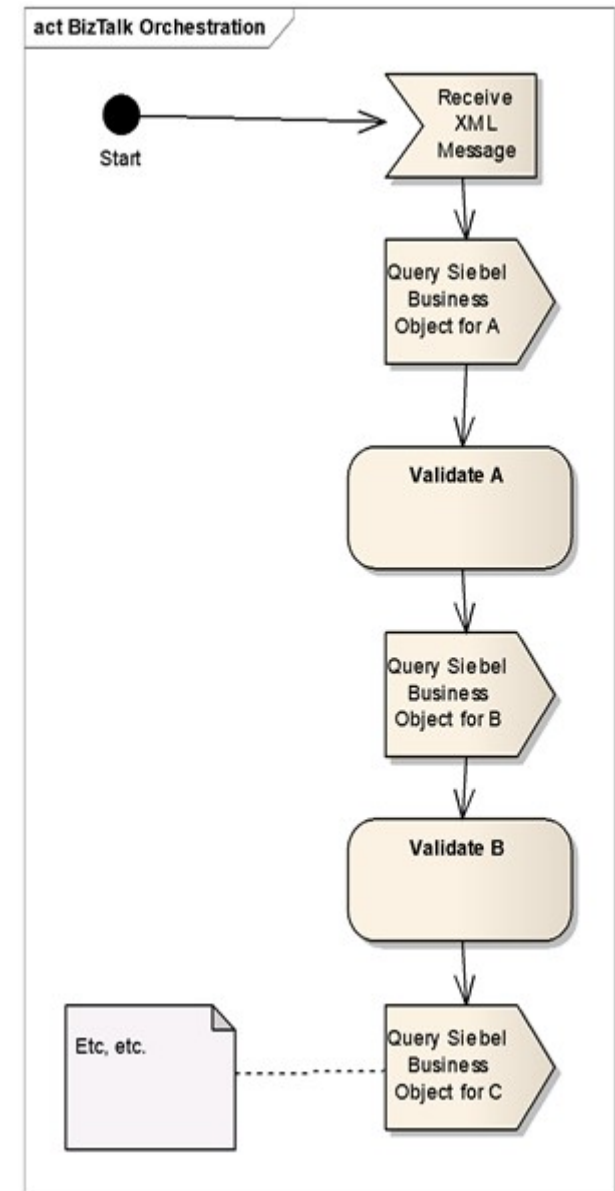


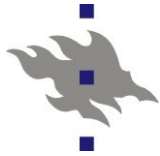
Yksinkertainen kahden roolin koreografia mallinnettuna UML sekvenssidiagrammina



Orkestraatioprosessit

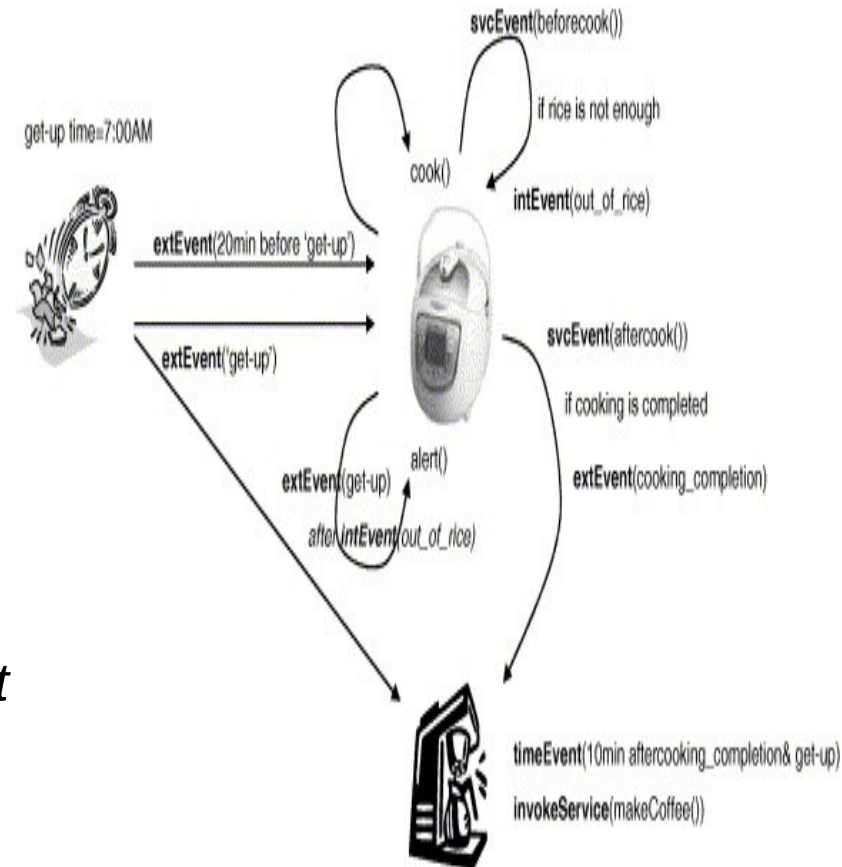
- Yleisin käytetty tapa tehdä palvelukompositioita on *orkestraatioprosessit*
- Keskitetty prosessi joka määrittelee
 - palvelukutsujen keskinäisen järjestyksen,
 - tietovuon orkestraation sisällä ja
 - kontrollirakenteet
- Orkestraation kuvaamiseen voidaan käyttää esimerkiksi UML:n aktiviteettikaatioita
- Toteutetaan esim. WS-BPEL kielellä





Palvelukoosteiden imperatiivinen ja deklaratiiivinen määrittely

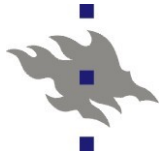
- Imperatiivinen koostaminen
 - Esimerkiksi prosessikielet
 - Määrittelee tiukan järjestyksen operaatioiden välille
- Deklaratiivinen koostaminen
 - Määritellään löyhät riippuvuudet tapahtumien välille
 - Joustavampi kuin imperatiivinen koostaminen
 - Sääntökielet (*rule languages*)
 - Tapahtumapohjaiset arkkitehtuurit (*event driven architectures*)
 - Säännöt tapahtumien koordinoimiseksi
 - (ECA, *event, condition, action*)
 - Tapahtumien välitys (*event bus, publish-subscribe, tuple-space*)



Jung, J., Park, J., Han, S., and Lee, K. 2007. An ECA-based framework for decentralized coordination of ubiquitous web services. *Inf. Softw. Technol.*

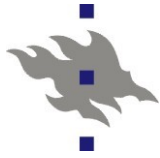
49, 11-12 (Nov. 2007), 1141-1161.

DOI=<http://dx.doi.org/10.1016/j.infsof.2006.11.008>



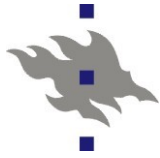
Palvelukoosteiden manuaalinen ja automaattinen syntetisointi

- Ennen palvelukoosteen varsinaista toimintaa voidaan tarvita koosteen syntetisointi
 - Synteesissä muodostetaan suoritettava versio koosteesta
- Manuaalinen syntetisointi
 - Synteesi tehdään tyypillisesti suunnittelu- tai toteutusvaiheessa
 - Esimerkiksi WS-BPEL prosessin määrittely.
- Automaattinen syntetisointi
 - Tyypillisesti muodostetaan deklarativista koostemäärittelyä vastaava konkreettinen prosessi
 - Automaattista syntetisointia käytetään erityisesti yhdessä deklarativisen määrittelyn ja dynaamisen sidonnan kanssa.
 - Edut
 - Löyhempi sidonta: päätepisteitä ei tarvitse sitoa kuin vasta viime hetkellä.
 - Joustavuus: voidaan valita soveltuvat palvelut tarpeen mukaan.



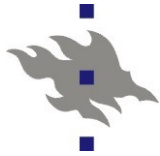
Koosteiden staattinen ja dynaaminen sidonta

- Staattinen koostaminen
 - Koostaminen suoritetaan suunnitteluaikana
 - Ei tue:
 - Palveluekosysteemin evoluutiota (uudet palvelut, palveluiden evoluutio)
 - Joustavaa palveluiden valintaa
- Dynaaminen koostaminen
 - Etuja
 - Joustavampi
 - Käyttäjän erityistarpeet voidaan huomioida ajonaikaisesti
 - Täysin uusia palveluita voidaan koostaa tarpeiden mukaisesti
 - Mahdollistaa palveluiden ajonaikaisen laajentamisen ja evoluution
 - Haittoja
 - Monimutkaisempia toteuttaa
 - Tarvitaan uusia infrastruktuuripalveluita (erityisesti dyn. sidonta)
 - Mahdolliset suorituskykyongelmat



Dynaamisen koostamisen menetelmiä

- Ajonaikainen uudelleenkonfigurointi
 - Esimerkiksi kääreiden (*wrapper*) hyödyntäminen
 - Esimerkiksi kommunikointisalauksen tuominen jo toteutettuun palveluun.
 - Myös muita menetelmiä (superimpositio, delegointi)
- Koostamiskielten käyttäminen
 - Hyödynnetään skriptaus- ja koordinoitikieliä sekä eksplisiittisiä kuvauksia palvelun komponenteista.
- Työvuokuvauksiin perustuvat menetelmät
 - Määritellään abstrakti työvuo, joka instantioidaan ajonaikaisesti.
- Ontologioihin perustuvat menetelmät
 - Hyödynnetään palvelua kuvaavien käsitteiden ja niiden välisiä suhteita kuvaavia ontologioita sekä koosteen loogista kuvausta.
- Deklaratiivisiin kuvauksiin perustuvat menetelmät
 - Käytetään koostamissääntöjä ja -rajoitteita palveluiden välillä.



Lähteet

- Dick A. Quartel, Maarten W. Steen, Stanislav Pokraev, and Marten J. Sinderen. Cosmo: A conceptual framework for service modelling and refinement. *Information Systems Frontiers*, 9(2-3):225-244, 2007.
- Michael P. Papazoglou and Willem-Jan Van Den Heuvel. Service-oriented design and development methodology. *Int. J. Web Eng. Technol.*, 2(4):412-442, 2006.
- Dustdar, S. and Schreiner, W. 2005. A survey on web services composition. *Int. J. Web Grid Serv.* 1, 1 (Aug. 2005), 1-30. DOI=<http://dx.doi.org/10.1504/IJWGS.2005.007545>