



Olioperustainen ohjelmistokehitys

- Olioperustaisuus (object oriented)
 - **järjestelmä (system)** muodostuu joukosta olioita (object), jotka yhteistyössä toimien tuottavat järjestelmän palvelut.
- Olio on kokonaisuus (entity), joka
 - kykenee suorittamaan omaan tietämykseensä (information contents) perustuvia palveluita (service / operation / method)

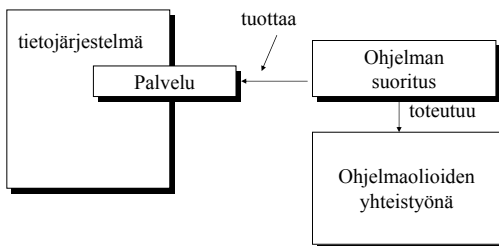


Olioperustainen ohjelmistokehitys

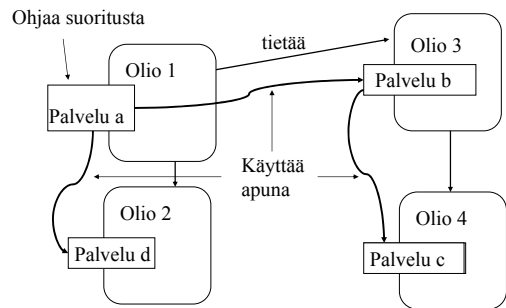
- Olio voidaan nähdä abstraktiona, jota voidaan soveltaa eri tilanteissa eri tavoin:
- Olio-ohjelmoinnissa:
 - ohjelman suoritus etenee tietokoneen sisuksissa toimivien tietojenkäsittelyolioiden yhteistyön tuloksena
 - *olioiden palvelut ovat suorituksen askelia*
- Tietojärjestelmän palvelut saadaan aikaan suorittamalla ohjelmia



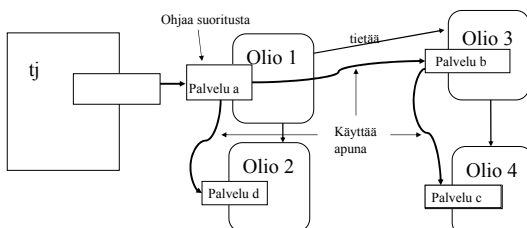
Tietojärjestelmän palvelut ja olio-ohjelmointi



Olioiden yhteistyö ohjelmassa



Tietojärjestelmän palvelun suoritus ohjelmassa

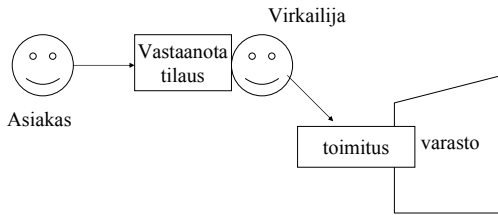


Liiketoiminta olioiden yhteistyönä

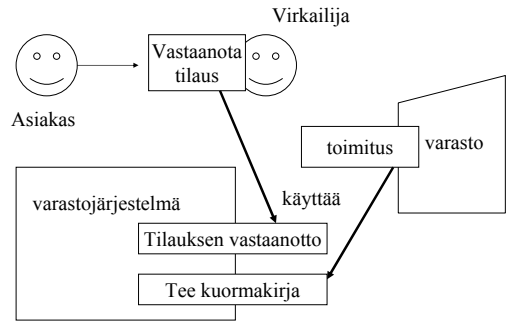
- Toisena ääripäänä
 - liiketoiminta voidaan myös hahmottaa liiketoiminnan osapuolten (jälleen olioita) välisenä yhteistyönä - yleensä liiketoiminnan osapuolina ovat organisaatioyksiköt ja henkilöt



Liiketoiminnan osapuolten yhteistyötä



Tietojärjestelmä avustavana oliona



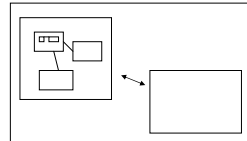
Olioperustainen kuvaaminen

- Tietojärjestelmä ja sen ympäristö voidaan siis hahmottaa erilaisten olioiden yhteistyön avulla.
- Miten tämä sitten pitäisi kuvata?
 - Mikä on oleellista ja välttämätöntä
 - Eri menetelmät korostavat eri asioita, tärkeänä voidaan pitää:
 - oliorakennetta, yhteistyötä, olioiden elinkaarta, järjestelmän käyttöä, tiedon kulkua,...
 - Kuvausjärjestelmät enimmäkseen graafisia:
 - yksi kuva kertoo enemmän kuin 1000 sanaa



Osajärjestelmät olioina

- Järjestelmät voidaan jakaa osajärjestelmiin.
- Yleensä kunkin osajärjestelmän vastuulla on jokin toiminnallinen kokonaisuus.
- Osajärjestelmät voidaan nähdä olioina
- Osajärjestelmä voidaan edelleen jakaa pienempiin osajärjestelmiin, jne...



UML

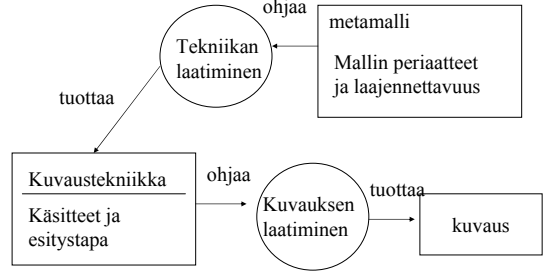
- UML (unified modeling language) on kokoelma käsitteitä ja kaaviotekniikoita oliokeskeiseen järjestelmäkehitykseen
- Yhdistelmä 'hyviksi havaituista' (of best practices) tekniikoista
 - OMT+Booch+OOSE
 - Rumbaugh, Booch, Jacobson
 - <http://www.rational.com/>



UML

- Kehitetty Rational Software Corp. toimesta 1996 ->
- OMG:n (Object Management Group) valinta kuvaustekniikaksi
 - OMG on olioteknologian kehittämiseen tähtäävä yritysten perustama organisaatio - tuotoksia mm. CORBA-teknologia
- Useat CASE- välineet (Computer Aided Software Engineering) tukevat tai ovat siirtymässä tukemaan UML-tekniikkoja

- UML sisältää:
 - metamallin, jolla määritellään varsinainen kuvausjärjestelmä, UML-mallit
 - *peruskäsitteet, käsitteiden väliset yhteydet, säännöt, laajennettavuus*
 - kokoelman graafisia kuvaustekniikoita eri tarkoituksiin



- UML-kuvaustekniikat
 - **Käyttötapamalli** (use case model)
 - *kuvaa mitä järjestelmällä tehdään*
 - **Luokkakaavio ja oliokaavio** (class/object diagram)
 - *tiedot ja mitä niille/niillä tehdään*
 - **Tilakaavio** (statechart diagram)
 - *olioiden elinkaaret tilakoneina*

- **Olioiden yhteistyökaaviot**
 - *palveluketjut (sequence diagram)*
 - *yhteistyörakenteet (collaboration diagram)*
 - *Kummatkin kuvaavat sitä kuinka toiminta hoidetaan olioiden välisenä yhteistyönä. Toimintokejuissa korostuu olioiden palvelujen hyväksikäyttö. Yhteistyörakenteissa olioiden välisten kytkentöjen hyödyntäminen yhteistyön perustana.*
- **Aktiviteettikaavio** (activity diagram)
 - *kontrollin kulku prosessissa tai ohjelman operaatiossa*
 - *haarautumat, workflow*

- **Komponenttikaavio** (component diagram)
 - *ohjelmiston koostuminen komponenteista*
- **Sijoittelukaavio** (deployment diagram)
 - *ohjelmiston osien sijoittuminen tietoverkkoon*
- Molemmat voidaan ajatella erikoistuneisiin olioihin perustuvina oliokaavioina