

## Ohjelmistoarkkitehtuurit

---

### Ohjelmistokehykset



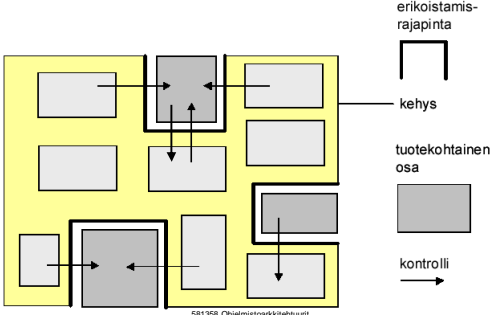
6.10.2015 581358 Ohjelmistoarkkitehtuurit 1

### Ohjelmistokehykset (software frameworks)

- Osittain abstraktiksi jätettyjä **ohjelmistorunkoja**, joita eri tavoin täydentämällä saadaan rakennettua kokonaisuuksia uusia sovelluksia tai sovelluksen osia
- Tavoitteena laajamittainen ja systemaattinen ohjelmistojen (sekä koodin että yleisrakenteen eli arkkitehtuurin) **uudelleenkäyttö**
  - Suosittu tekniikka tuoterunkoarkkitehtuurien toteuttamiseksi
- Olioperustaisissa kehyksissä **ohjelmistorunko** = kokoelma luokkia, komponentteja ja rajapintoja
  - Ohjelmistokehykset sisältävät ohjelmakoodia, joten ne ovat sidoksissa ohjelmointikielen
  - Kehys on usein osa laajempaa *ohjelmistoalustaa* (platform) ja kehykseen voi kuulua omia työkaluja ja apuvälineitä

6.10.2015 581358 Ohjelmistoarkkitehtuurit 2

### Ohjelmistokehykset



erikoistamis-  
rajapinta

kehys

tuotekohtainen  
osa

kontrolli

6.10. 581358 Ohjelmistoarkkitehtuurit 3

### Ohjelmistokehykset

- Kehykset eroavat (luokka-, funktio-) *kirjastoista* (library)<sup>1</sup>
  - *Inversion-of-control*: kehyksen sisäänrakennettu koodi ja sen logiikka ohjaa sovelluksen suoritusta (kontrollinkulkua), ei sovelluskohtainen koodi kuten luokkakirjastoja käytettäessä
  - *Oletustoiminnallisuus*: kehyks tarjoaa käyttökelpoista oletustoiminnallisuutta, ei pelkästään tynkätoteutuksia (no-ops, stubs)
  - *Laajennettavuus*: kehyksen käyttäjä voi valikoiden syrjäyttää, erikoistaa tai lisätä kehyksen tarjoamaa toiminnallisuutta oman sovelluksena tarpeisiin kehyksen tekijän etukäteen määrittämällä tavoilla
  - *Kehyksen muuttumaton koodi*: kehyksen koodia ei ole (yleensä) tarkoitettu käyttäjän muutettavaksi - paitsi laajentamalla sitä tietyillä tavoilla erikseen määrättyissä kohdissa (kts. edellinen kohta)

[1] [http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework)

6.10.2015 581358 Ohjelmistoarkkitehtuurit 4

### Ohjelmistokehykset

- Ohjelmoijan näkökulmasta usein: Kehys = API
  - Teknisesti ajatellen kehys yleensä sisältää ja tarjoaa useita API:ja (rajapintoja) eri tarkoituksiin
  - Rajanveto ei aina käytännössä ole kovin selvä, käytännössä monia kehyksiä kutsutaan yksinkertaisesti API:ksi
- Ensimmäinen laajalti käytetty ohjelmistokehyks: Smalltalk-80-ympäristön *Model-View-Controller*
  - Alkuperäisen Model-View-Controller-kehyyksen arkkitehtuurin pohjalta määritelty MVC-arkkitehtuurityyli
- Erityisesti käyttöliittymätoteutukseen on kehitetty useita kehyksiä työasemasovelluksiin, web-sovelluksiin, useille eri ohjelmointikielille, kaupallisia – ei kaupallisia

6.10.2015 581358 Ohjelmistoarkkitehtuurit 5

### Ohjelmistokehykset

- Kehys voi olla kokonaisvaltainen koko sovellusta hallitseva tai osittaisongelmaan tarkoitettu tukikehyks
  - esim. käyttöliittymäkehyykset rajautuvat käyttöliittymän toteutukseen, Java Enterprise Edition tarjoaa puitteet ja tuen kokonaisten yritysjärjestelmien toteuttamiseen
  - Ns. *sovelluskehyykset* on tarkoitettu tietyntyyppisten sovellusten toteuttamiseen
  - Androidissa Java-sovelluksen rakenteen määrää ohjelmistokehyks, joka on osa laajempaa käyttöjärjestelmän sisään rakennettua sovellusarkkitehtuuria
- Ohjelmistokehyks ei ole valmis ohjelmistototeutus, vaan toteutus saadaan aikaan kehystä täydentämällä tai mukauttamalla

6.10.2015 581358 Ohjelmistoarkkitehtuurit 6

## Ohjelmistokehykset

- Ohjelmistokehyksen **erikoistaminen** (framework specialization, framework adaptation) = ohjelmiston (osan) toteuttaminen täydentämällä kehyyksen tarjoamaa ohjelmarunkoa
  - Abstraktien luokkien erikoistaminen,
  - Toiminnallisuuden koostaminen kehyyksen konkreettisista luokista
- **Sovelluskehys** (application framework) = kehys, josta voidaan erikoistaa kokonainen sovellus
- **Komponenttikehys** (framelet) = "minikehys", jonka erikoistamisen tuloksena syntyy yksittäinen komponentti

6.10.2015

581358 Ohjelmistoarkkitehtuurit

7

## Ohjelmistokehykset

- **Laajennoskohta, variaatiopiste** (hot spot, variation point) = kehyyksen "aukko kohta", jota täydentämällä voidaan sovelluksen puolella varioida ja/tai ottaa käyttöön tietty kehyyksen toiminnallisuus/ominaisuus
- **Erikoistamisrajapinta** (specialization interface) = laajennoskohtien ja niihin liittyvien vaatimusten kokoelma
  - Kehyyksen erikoistamisrajapinta on tyypillisesti paljon monimutkaisempi kuin yksittäisen komponentin rajapinta
  - Erikoistamisrajapinnan laajennoskohtien välillä on riippuvuuksia, joiden ymmärtäminen on edellytys kehyyksen oikealle käytölle

6.10.2015

581358 Ohjelmistoarkkitehtuurit

8

## Ohjelmistokehykset

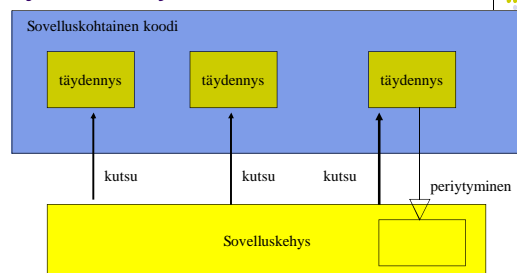
- Erikoistamiseen voidaan käyttää esim.
  - rajapintojen toteutusta
  - periytmistä ja syrjäyttämistä (jos kieli tarjoaa)
  - olioiden luontia ja kompositiota
  - geneerisiä rakenteita
  - myöhäistä sidontaa ja sitä tukevia rakenteita
- Sovelluskehyyksissä **päälogiikasta vastaa kehys** ja sovelluskohtaiset erityistoimet toteutetaan täydennyksinä

6.10.2015

581358 Ohjelmistoarkkitehtuurit

9

## Ohjelmistokehykset



Kutsuissa käytössä ns. käänteinen kutsurakenne = kehys kutsuu dynaamista sidontaa hyväksikäyttäen sovelluskohtaisia täydentäviä moduleja (Hollywood-periaate: don't call us we call you)

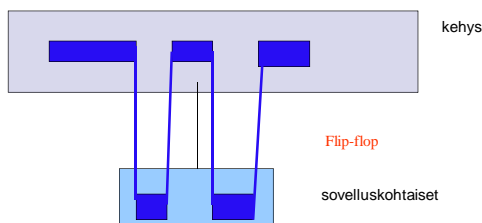
6.10.2015

581358 Ohjelmistoarkkitehtuurit

10

## Ohjelmistokehykset

Kehyksissä ohjelman kontrollia ohjaavat vuoroin kehys vuoroin sovelluskohtaiset täydennykset



6.10.2015

581358 Ohjelmistoarkkitehtuurit

11

## Ohjelmistokehykset

- Kehyksissä sovelletaan tyypillisesti suunnittelumalleja (design patterns) erilaisten hyvinä pidettyjen asioiden aikaansaamiseksi
  - Joidenkin mallien soveltaminen on välttämätöntä kehyyksen erikoistamismahdollisuuksien kannalta
- Kehyyksen koodi sisältää tyypillisesti useiden suunnittelumallien ilmentymiä
  - "Alkuperäiset" suunnittelumallit on "löydetty" analysoimalla olemassa olevia kehyyksiä (esimerkiksi Smalltalk, HotDraw)
  - Suunnittelumallit sopivat usein kehysten dokumentointiin [Johnson, 1992]
- Suunnittelumalleissa on tyypillisesti kehyyksen kuuluva osa (esim. abstraktit luokat) ja tuotekohtainen osa (esim. vaihtuvat konkreettiset luokat)

6.10.2015

581358 Ohjelmistoarkkitehtuurit

12

**(Olio-)Ohjelmistokehykset**

- Eri tyyppisiä kehyksiä:
  - Abstrakti kehys
  - Muunneltava kehys (white box framework)
  - Pistokekehys (plugin framework)
  - Koottava kehys (black box framework)

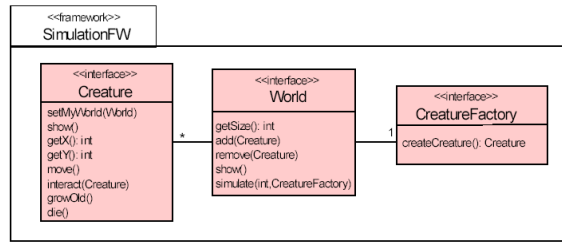
6.10.2015

581358 Ohjelmistoarkkitehtuurit

13

**(Olio-)Ohjelmistokehykset**

- Abstrakti kehys: vain abstrakteja luokkia (tai rajapintoja); ei toiminnallisuutta
- vuorovaikutus on toteutettava sovelluskohtaisesti



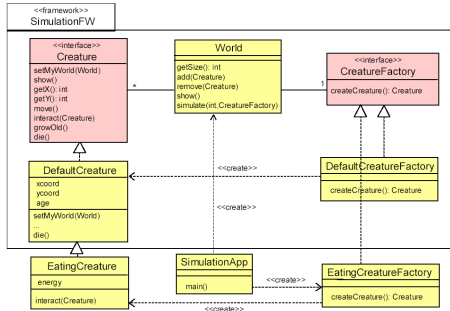
6.10.2015

581358 Ohjelmistoarkkitehtuurit

14

**(Olio-)Ohjelmistokehykset**

- Muunneltava kehys: Tarjoaa rajapinnat ja ohjelman päälogiikan
- Myös periminen ja syrjäyttäminen erikoistamistekniikoina



6.10.2015

581358 Ohjelmistoarkkitehtuurit

15

**(Olio-)Ohjelmistokehykset**

- Muunneltavan kehyksen yhteydessä kehyksen käyttäjän on oltava hyvin perillä kehyksen toiminnasta kyetäkseen käyttämään sitä
- Metodien väliset riippuvuudet voivat aiheuttaa ongelmia. Yhden syrjäyttäminen voi edellyttää jonkin toisenkin syrjäyttämistä, jolloin luokkien määrän kasvaa nopeasti.

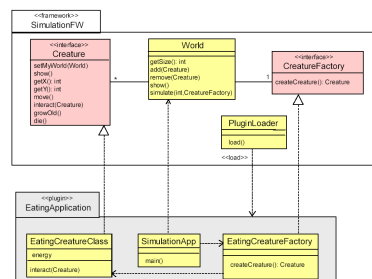
6.10.2015

581358 Ohjelmistoarkkitehtuurit

16

**(Olio-)Ohjelmistokehykset**

- Pistokekehys: erikoistetaan (miltei yksinomaan) rajapintoja; erikoistus tarjotaan yhtenäisenä plugin-komponenttina, esim Eclipse plugins



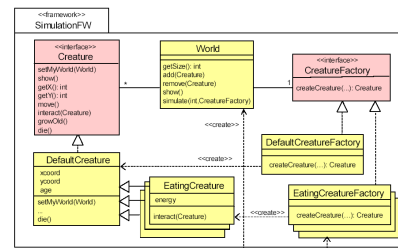
6.10.2015

581358 Ohjelmistoarkkitehtuurit

17

**(Olio-)Ohjelmistokehykset**

- Koottava kehys: Koostetaan sovellus kehyksen tarjoamista valmiista komponenteista; ei takaisinkutsuja sovelluskoodiin



6.10.2015

581358 Ohjelmistoarkkitehtuurit

18

## (Olio-)Ohjelmistokehykset

- Koottavat kehykset
  - Uutta toiminnallisuutta saadaan yhdistelemällä tarjolla olevia olioita uusin tavoin ja säätämällä toimintaa parametrein
  - Perintä korvataan toimintojen delegoinnilla ja parametroinnilla

6.10.2015

581358 Ohjelmistoarkkitehtuurit

19

## (Olio-)Ohjelmistokehykset

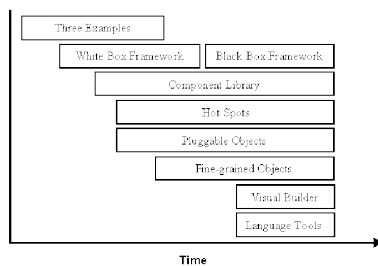
- Koottavat kehykset ...
  - Käyttäjän ei tarvitse tuntea kehyksen sisäisiä yksityiskohtia, mutta on tiedettävä millaisia olioita voi luoda ja miten niitä voi kytketään toisiinsa.
  - Luotavien olioiden luokat ovat valmiiksi määriteltyjä. Luotavaan oloon voi vaikuttaa parametreilla.
  - Koottavat kehykset (black-box) ovat yleensä helpompikäyttöisiä kuin läpinäkyvät kehykset. Toisaalta niitä on vaikeampi kehittää koska kehittämisessä pitää varautua monipuolisemmin eri käyttötilanteisiin.
  - Koska joustavuudella on ennakkomäärittelyn asettamat rajat, eivät koottavat kehykset voi soveltamisalueeltaan olla kovin laaja-alaisia

6.10.2015

581358 Ohjelmistoarkkitehtuurit

20

## Ohjelmistokehyksen evoluutio



Kehyksen elinkaaren aikana tyypillisesti esiintyviä "patteerjeja"  
 Roberts: <http://st-www.cs.illinois.edu/users/droberts/evolve.html>

6.10.2015

581358 Ohjelmistoarkkitehtuurit

21

## Ohjelmistokehysten riskejä

- Tekninen vaativuus ja monimutkaisuus
  - Arkkitehdin tunnettava hyvin sovellusalue ja joustavuuteen liittyvät oliotekniikat (esim. suunnittelumallit)
  - Abstraktius voi tehdä kehyksestä erikoistajille hankalasti ymmärrettävän
- Kehysten yhdistely voi olla ongelma
  - Mitä tehdään, jos halutaan käyttää useampaa kehystä, joista kukin määrittelee pääkontrollisilmukan?

6.10.2015

581358 Ohjelmistoarkkitehtuurit

22

## Ohjelmistokehysten riskejä

- Monoliittisuus
  - Kehys saattaa kasvaa suureksi ja hallitsemattomaksi
- Laadullinen varianssi
  - Variaatiopisteet liittyvät useimmiten toiminnallisuuden muokkaamiseen
  - Laatuominaisuuksia variointi sen sijaan yleensä hankalaa (esim. tietoturvan lisääminen, suorituskyvyn optimointi, ...)
- Dokumentointi
  - Olennaista on erikoistamisrajapinnan kuvaus
  - Tälle ei kuitenkaan vakiintunutta kuvaustapaa

6.10.2015

581358 Ohjelmistoarkkitehtuurit

23

## Ohjelmistokehysten riskejä

- Toteutuksen ongelmia
  - Joustavien ratkaisujen toteuttaminen edellyttää asiantuntemusta (esim. suunnittelumallit auttavat)
  - Joustavuus → monimutkaisuus, abstraktit käsitteet
    - Ylläpito vaativaa
    - Testaus? (vrt. tuoterunkojen testaus)
- Käytön ongelmia
  - Miten kehysten käyttö pitäisi dokumentoida?
    - Keittokirjat (cookbooks)
    - Suunnittelumallipohjainen dokumentointi
  - Riittääkö tavanomainen dokumentaatio erikoistamisrajapinnan kuvaukseen?
  - Työkalutuki sovellusten rakentamiseen

6.10.2015

581358 Ohjelmistoarkkitehtuurit

24