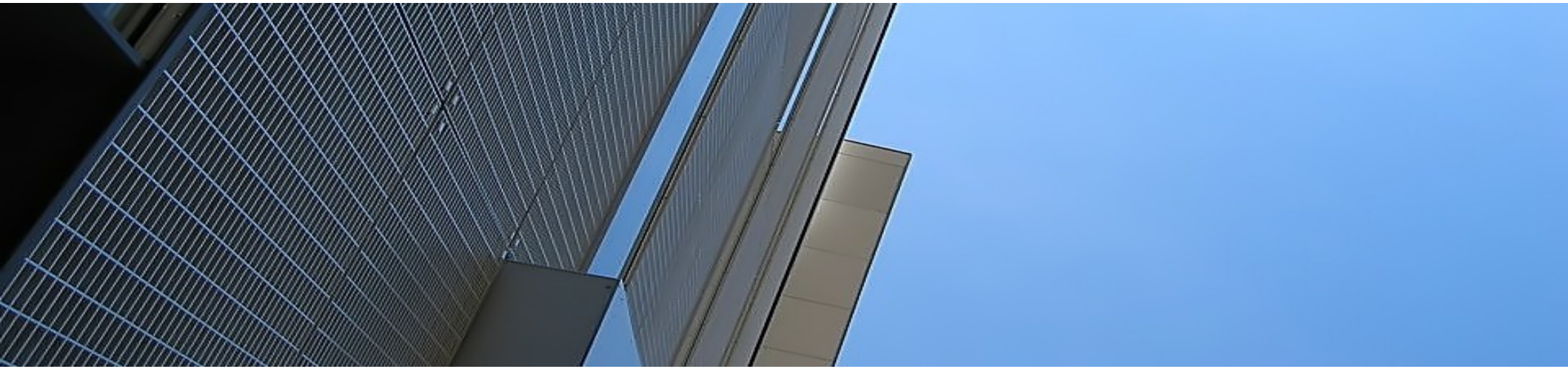
CINCO

Collaborative and interoperable computing



Palvelusuuntautunut ohjelmistotuotanto

Luento 8: Näkökulmien mallinnus ja soveltaminen; Kurssin yhteenveto

Toni Ruokolainen

Luennon runko

- Näkökulmien mallinnuksesta
  - Näkökulmat, näkymät ja näkökulmien väliset vastaavuudet
  - Web-sovellusten toiminnallisten piirteiden mallinnus
    - UML-pohjainen Web-sovellusten mallinnusmetodologia
    - Näkökulmamallien synkronisointi
  - Ei-toiminnallisten piirteiden mallinnus
    - Aspektien mallinnus
    - Aspektien sitominen malleihin
- Kurssin luento-osuuden yhteenveto

Näkökulmien mallinnuksesta

- Suurten järjestelmien mallinnuksessa sovelletaan yleensä erilaisia näkökulmia monimutkaisuuden hallitsemiseksi
- **Näkökulma** (*viewpoint)* määrittelee esitys- ja lähestymistavat järjestelmän näkymän kuvaamiseksi ja käsittelemiseksi
  - Määrittelee erityisesti näkökulmakielen sekä käsitteet näkökulman mallintamiseksi
  - Esimerkiksi järjestelmän käyttäytymisnäkökulma voi määritellä mitä tulee käyttäytymisestä määritellä ja minkälaista mallinnuskieltä käyttäen se kuvataan
- **Näkymä** (*view)* määrittelee osittaiskuvauksen järjestelmästä toisistaan riippuvien piirteiden joukon näkökulmasta
  - Esimerkiksi UML aktiviteettidiagrammina annettu kuvaus tietyn järjestelmän käyttäytymisestä
  - Näkymä mallinnetaan *näkökulmamallissa*
- Näkökulmat eivät ole toisistaan riippumattomia
  - Näkymät kuvaavat osittain samoja elementtejä ja hyödyntävät toisten näkymien sisältöä hyväkseen

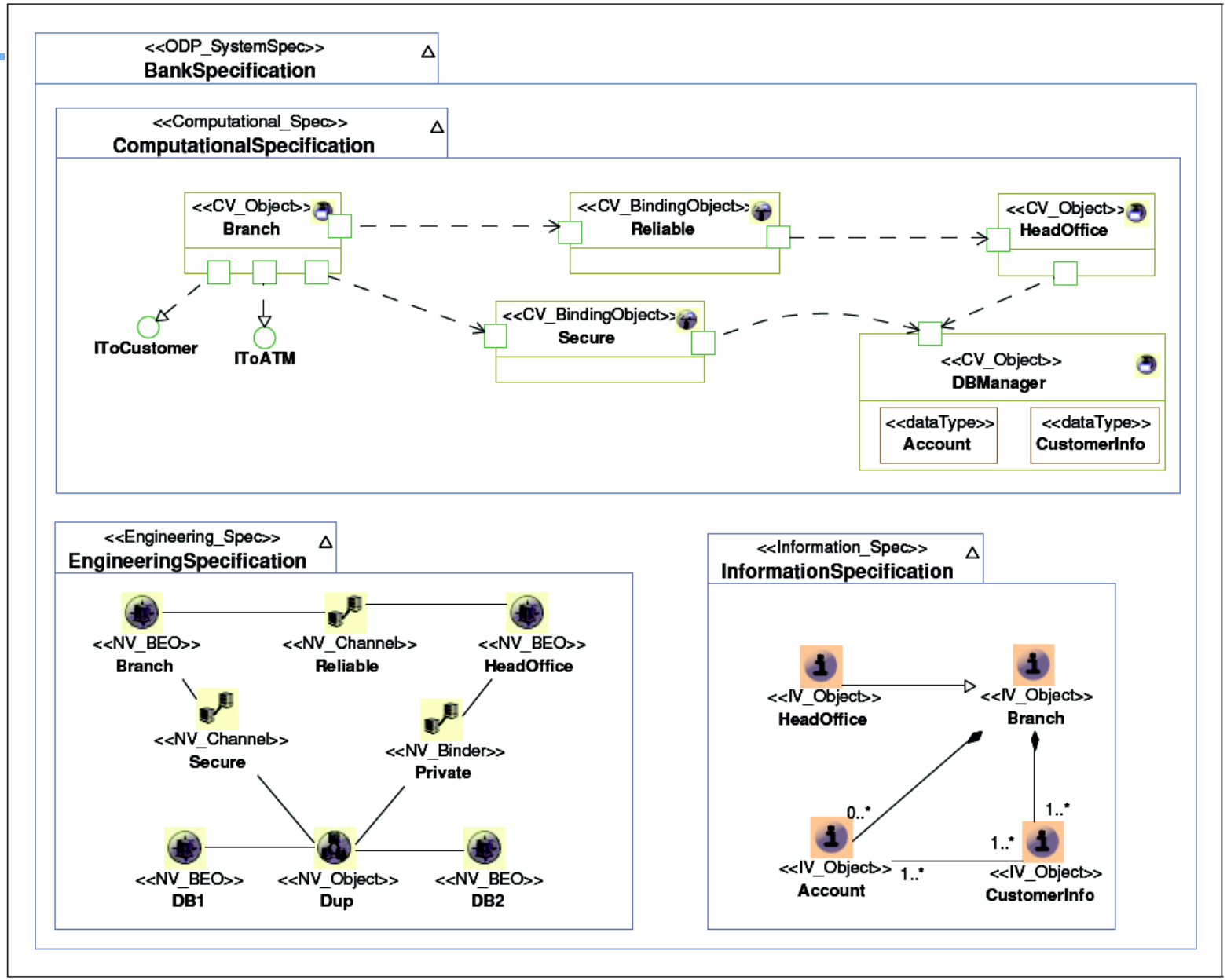
Näkökulmien ongelmia

- Miten voidaan varmistaa, että näkökulmien soveltaminen tuottaa kuvauksen täsmälleen yhdestä järjestelmästä?
  - Näkökulmien integrointiongelma: kuvaavatko näkymät todellakin samaa järjestelmää?
- Miten voidaan varmistaa, että näkökulmat eivät ole keskenään ristiriitaisia?
  - Näkökulmien konsistenssiongelma
- Näiden ongelmien ratkaisemiseksi tulee näkökulmien väliset vastaavuudet (*correspondence*) olla selvillä!
- Useimmat näkökulmiin perustuvat mallinnuskehikot eivät määrittele eksplisiittisesti näkökulmien välisiä vastaavuuksia
  - Käytetään sen sijaan esimerkiksi nimivastaavuutta
    - Saman nimen käyttö eri näkökulmamalleissa tarkoittaa että vastaavat mallielementit käsittelevät järjestelmän samaa komponenttia eri näkökulmista
  - Vastaavuuksien implisiittinen määrittely voi aiheuttaa ongelmia
    - Erityisesti jos näkökulmat suunnitellaan hajautetusti autonomisten toimijoiden toimesta; Tulkintavirheet vastaavuuksista

Esimerkki näkökulmien määrittelystä: ODP standardi

- ODP (*Open Distributed Processing)* on joukko ISO standardeja avoimien hajautettujen järjestelmien kuvaamiseen
- Käytössä viisi näkökulmaa
  - Yritysnäkökulma (*enterprise viewpoint)*: toiminnan tavoitteet
  - Informaationäkökulma (*information viewpoint*): informaation semantiikka
  - Toiminnallinen näkökulma (*computational viewpoint)*: järjestelmän käyttäytyminen
  - Toteutusnäkökulma (*engineering viewpoint)*: toteuttamiseen vaadittava infrastruktuuri
  - Teknologinen näkökulma (*technological viewpoint):* laitteistojen ja ohjelmistojen vaatimukset
- Näkökulmamallien määrittelyyn voidaan käyttää UML-pohjaista mallinnuskieltä
  - UML4ODP
  - Jokaiselle näkökulmalle on määritelty oma UML-profiilinsa

Esimerkki ODP näkökulmamalleista

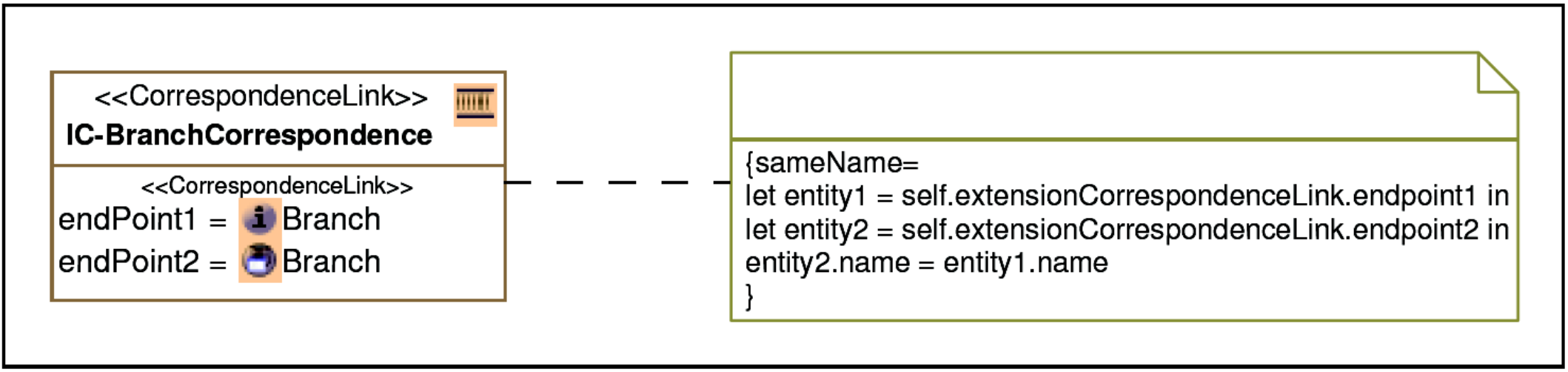


Kuva: Romero, J. R., Jaén, J. I., and Vallecillo, A. 2009. Realizing correspondences in multi-viewpoint specifications. In Proceedings of the 13th IEEE international Conference on Enterprise Distributed Object Computing (Auckland, New Zealand, September 01 - 04, 2009). IEEE Press, Piscataway, NJ, 138-147.

Näkökulmien väliset vastaavuudet ODP standardissa

- Näkökulmien väliset vastaavuuden määritellään eksplisiittisesti
  - Vastaavuusmäärittelyt eivät kuulu mihinkään näkökulmaan, vaan määrittelevät toteamuksia kahden eri näkökulman käsitteiden välisistä suhteista
- Näkökulmavastaavuus *(viewpoint correspondence)*
  - Määrittelee näkökulmassa käytetyn termin tai konstruktin suhteen toisen näkökulman termin tai konstruktin kanssa
- Järjestelmän kuvaus siis koostuu
  - Joukosta näkymiä
  - Joukosta näkymien välisiä vastaavuuksia
- UML4ODP:ssä näkökulmien vastaavuuden voidaan määritellä
  - Intentionaalisesti (metamallitasolla) QVT:tä ja vastaavuusluokkia kuvaavaa UML-profiilia käyttäen
  - Ekstensionaalisesti (terminaalimallitasolla) erityistä UML-profiilia hyödyntäen

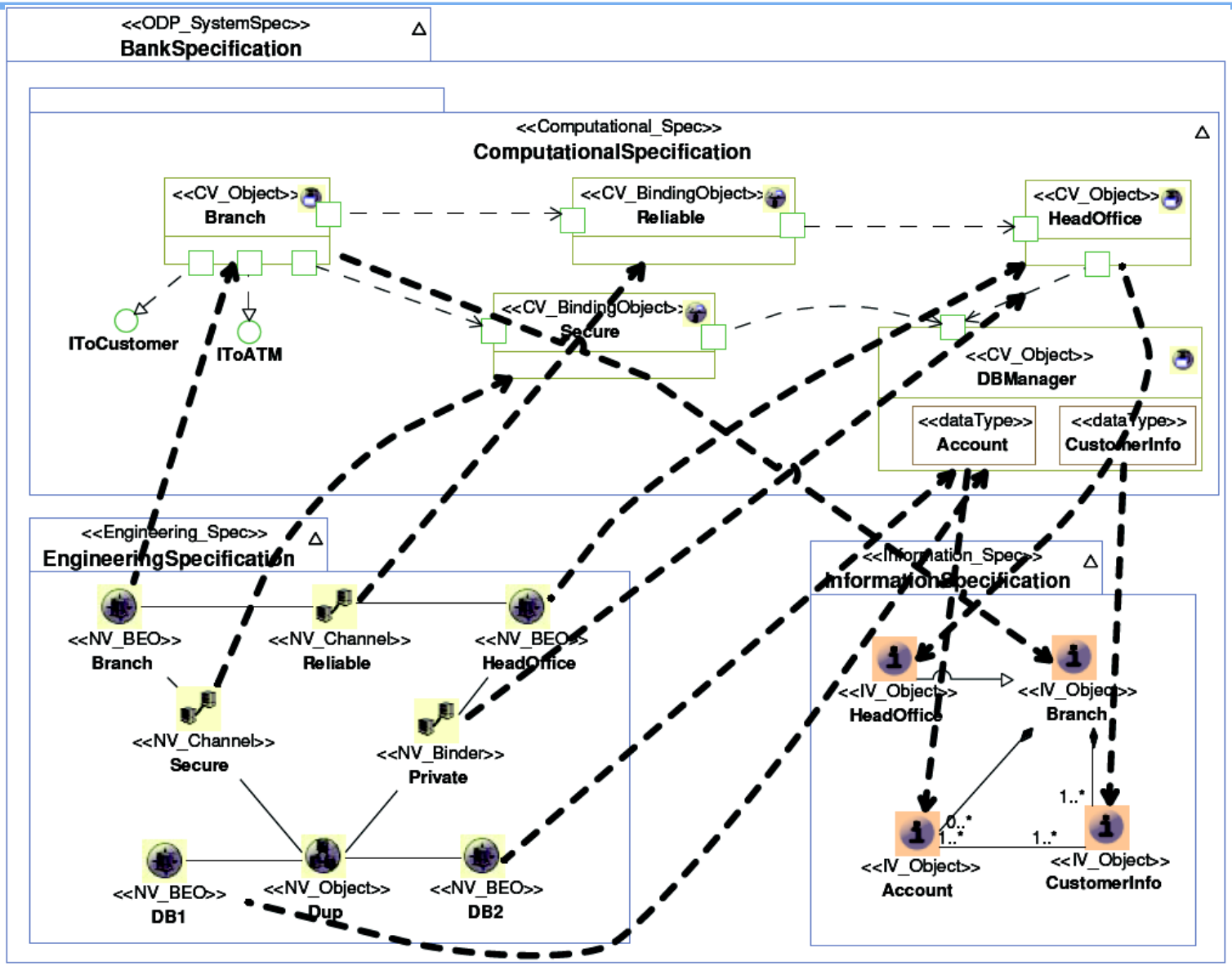
Näkökulmien vastaavuuksien intentionaalinen määrittely ODP:ssä




```
relation IC-BranchCorrespondence {
   domain iv iBranch:Class {name="Branch"}
   domain cv cBranch:Component {name="Branch"}
   when {
      iBranch.stereotypedBy("IV_Object") and
      cBranch.stereotypedBy("CV_Object")
   }
}
```

Kuva: Romero, J. R., Jaén, J. I., and Vallecillo, A. 2009. Realizing correspondences in multi-viewpoint specifications. In Proceedings of the 13th IEEE international Conference on Enterprise Distributed Object Computing (Auckland, New Zealand, September 01 - 04, 2009). IEEE Press, Piscataway, NJ, 138-147.

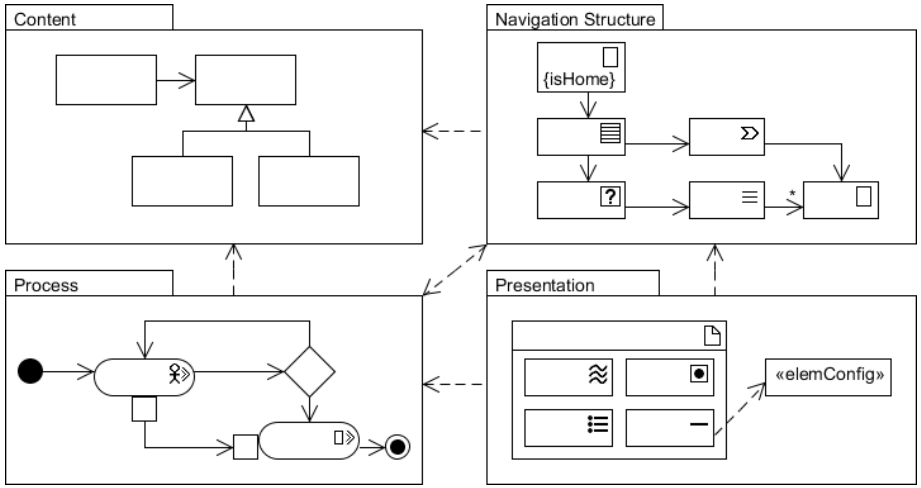
Esimerkki näkökulmien vastaavuuksien ekstensionaalisesta määrittelystä



Kuva: Romero, J. R., Jaén, J. I., and Vallecillo, A. 2009. Realizing correspondences in multi-viewpoint specifications. In Proceedings of the 13th IEEE international Conference on Enterprise Distributed Object Computing (Auckland, New Zealand, September 01 - 04, 2009). IEEE Press, Piscataway, NJ, 138-147.

Esimerkki: Web-sovellusten näkökulmien mallinnus

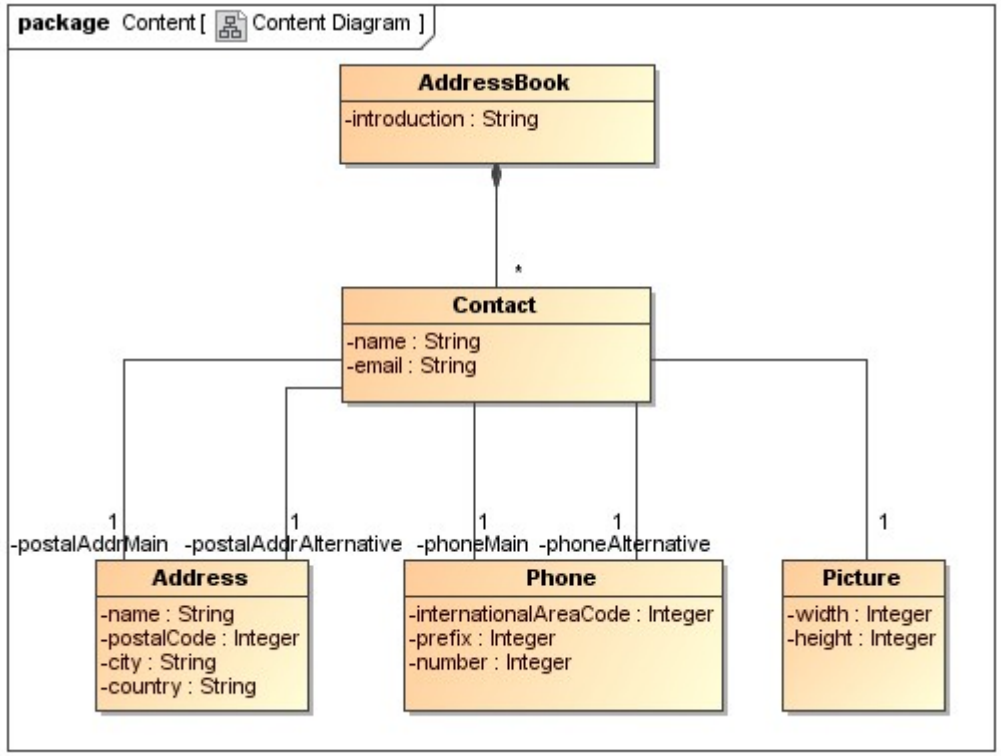
- UWE: UML-based Web Engineering
  - Münchenin LMU-yliopistossa kehitetty Web-sovellusten kehitysmetodologia
  - Hyödyntää UML-profiileja ja mallimuunnoksia Web-sovellusten mallinnukseen ja ohjelmistotuotantoprosessin toteuttamiseen
  - Web-sovellus: verkkoselaimella loppukäyttäjän hyödynnettävissä oleva palvelu
- Näkökulmamallit
  - Content model: määrittelee sovelluksen informaatioelementit
  - Navigation model: määrittelee hypertekstisivujen väliset linkit
  - Presentation model: määrittelee yksittäisten sivujen rakenteen
  - Process model: määrittelee käyttäjän ja Web-sovelluksen välisen interaktioprosessin



Kuva: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Content Model

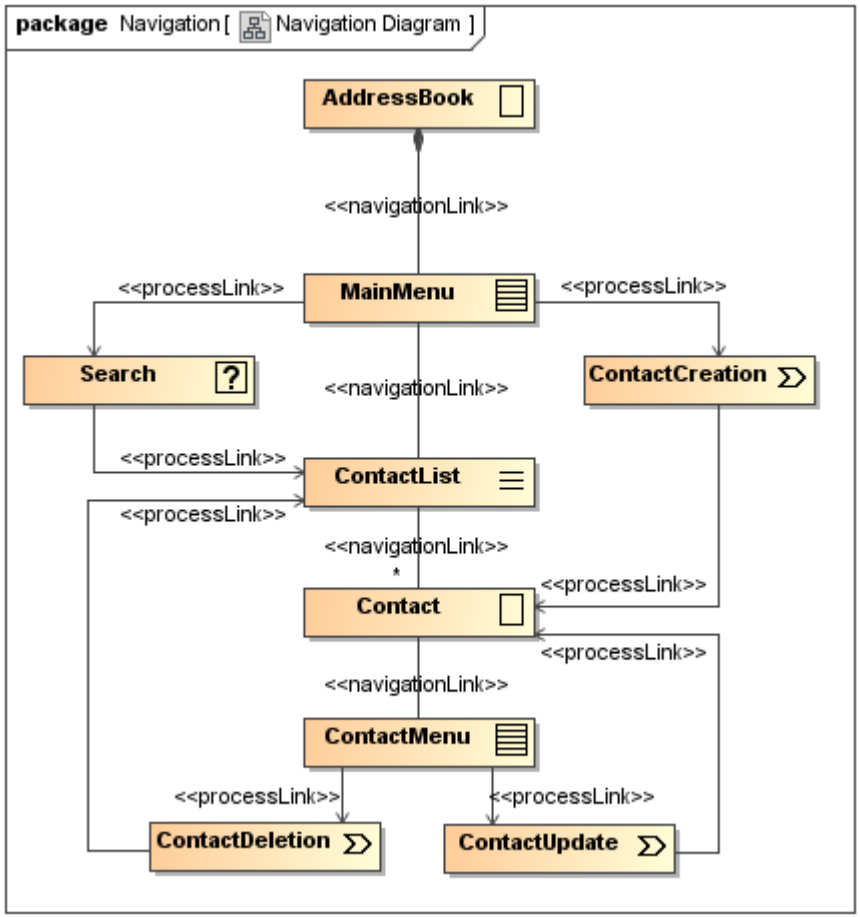
- Content Model määrittelee sovelluksen toimialuekohtaisen tiedon
  - Informaatioelementit
  - Toimialuekohtaiset käyttäytymiskuvaukset
- Informaatioelementit määritellään UML-luokkakaavioiden avulla
- Käyttäytymiskuvaukset voivat määritellä esimerkiksi informaatioelementtien elinkaaria



Kuva: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Navigation Model

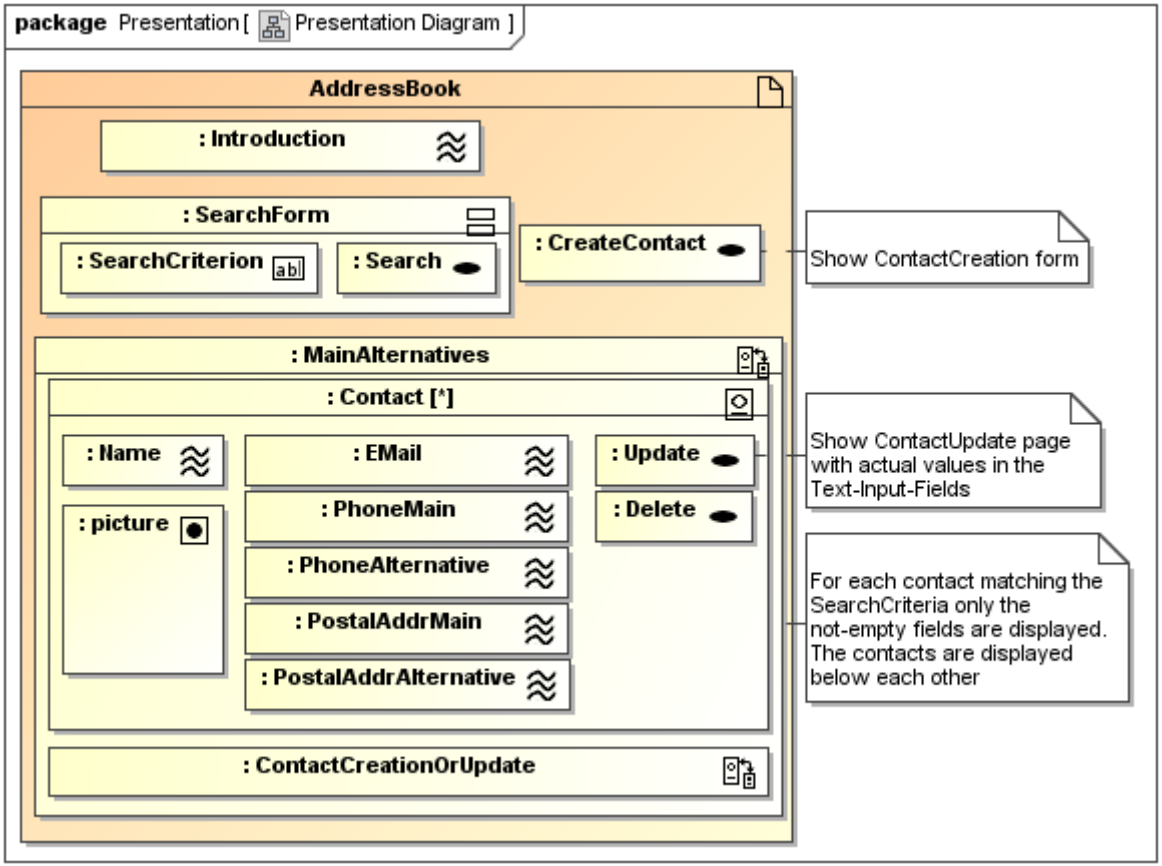
- Navigation Model kuvaa sovelluksen hypertekstinäkökulman
  - Perustuu Content Model:iin
    - Tuotetaan mallimuunnoksella Navigation Model:ista
  - Käyttäjälle näytettävien sivujen väliset linkit
  - Sivujen tyypit
    - mm. navigointisivu, menu, kysely, prosessointisivu, lista (indeksi)



Kuva: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Presentation Model

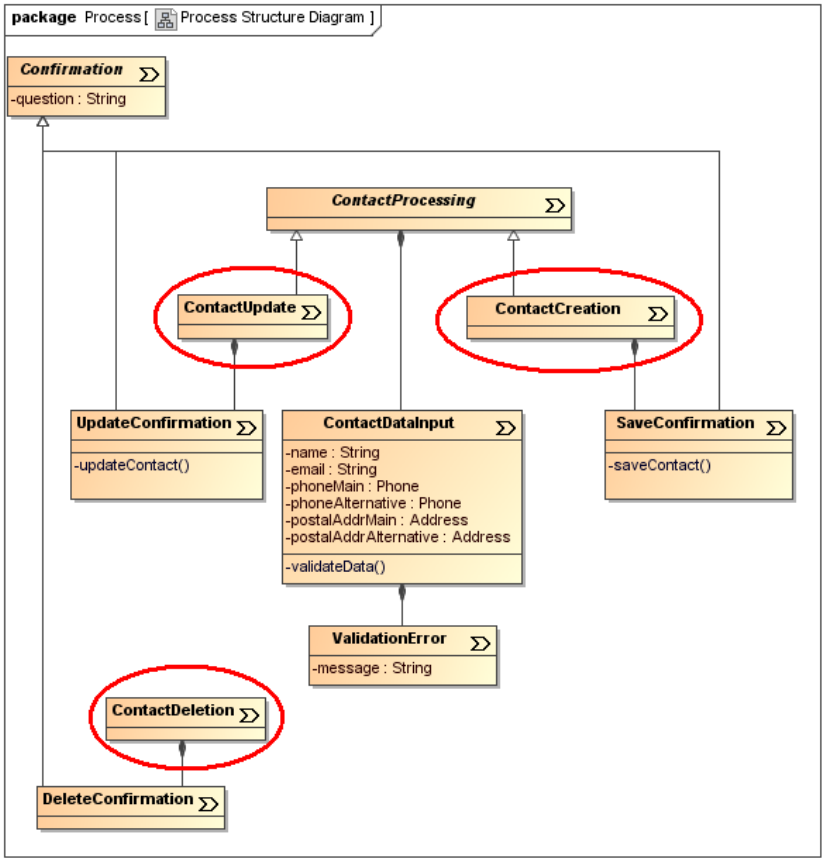
- Presentation model määrittelee yksittäisten sivujen sisällön ja niiden sisältämät komponentit

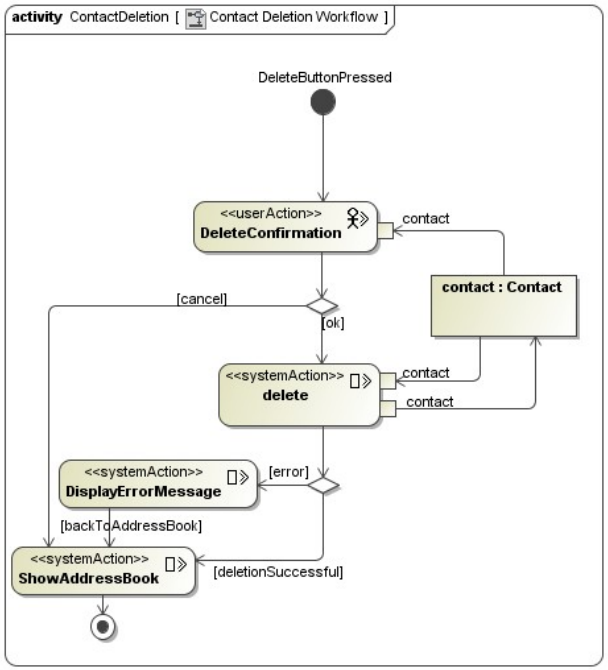


Kuva: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Process Model

- Määrittelee
  - prosessointisivujen väliset yhteydet (Process structure model)
  - Prosessointisivujen sisältämät käyttäytymismallit (Process flow model)





Kuvat: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Vastaavuuksien määrittely UWE:ssä

- Näkökulmien väliset vastaavuudet määritellään UML:ää ja OCL:ää käyttäen
- UML-profiileissa ollaan määritelty valmiiksi yleisesti esiintyviä vastaavuuksia stereotyyppeinä
  - Esimerkiksi *sameName*
  - Vastaavuusstereotyypeillä annotoidaan malleja

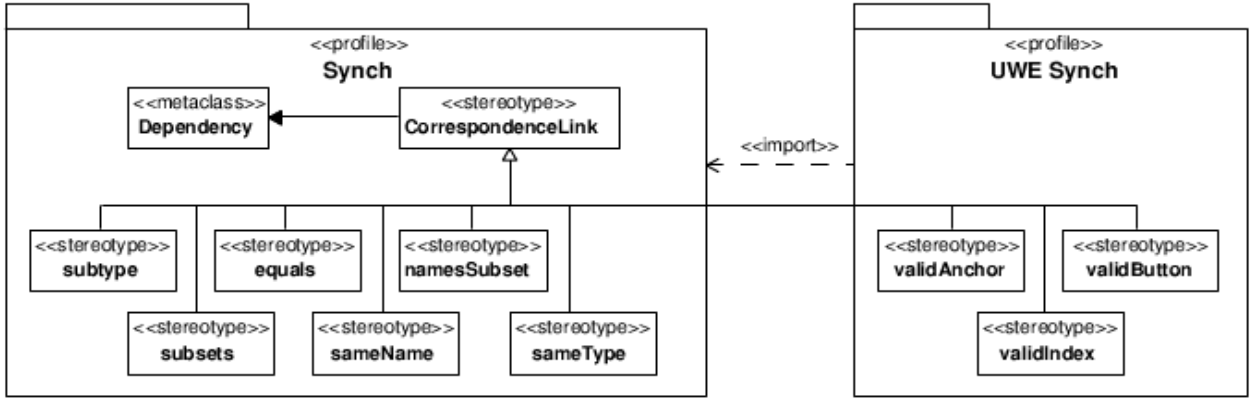


Fig. 6. The Synch and UWESynch Profiles (excerpt).

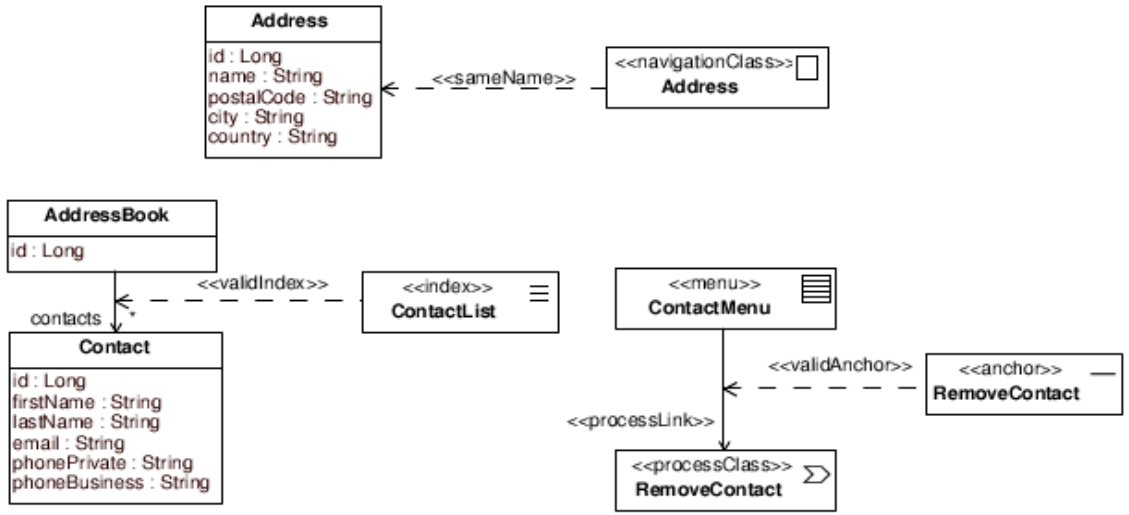
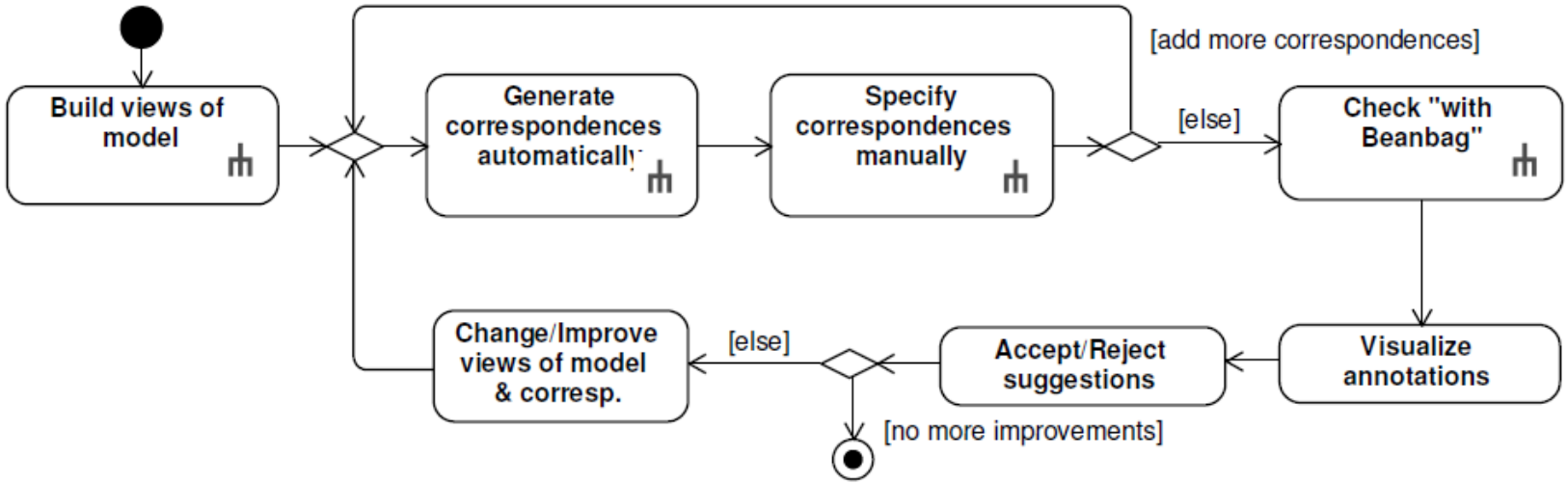


Fig. 7. Example of correspondence specifications using our profile.

Näkökulmien synkronointi UWE:ssä

- UWE käyttää eksplisiittisiä näkökulmien välisiä vastaavuuksia
  - Perustuu ODP:n mukaiseen lähestymistapaan
  - Käytää sekä intentionaalisia että ekstensionaalisia vastaavuusmäärityksiä
  - Vastaavuuden määritellään UML-profiileihin pohjautuvalla mallinnuskielellä
- Halutaan pitää näkökulmat synkronoituna
  - Näkökulmamalleja muutettaessa, tai
  - Näkökulmien välisiä vastaavuuksia päivitettäessä



Kuva: Daniel Ruiz-González, Nora Koch, Christian Kroiss, José-Raúl Romero, and Antonio Vallecillo. Viewpoint Synchronization of UWE Models. In Proc. MDWE 2009 - 5rd International Workshop on Model-Driven Web Engineering, CEUR-WS, Vol 455, June 2009.

Aspektimallinnus *(Aspect-Oriented Modelling, AOM)*

- Aspektiperustaisen mallinnuskehikon elementit
  - Primäärimalli määrittelee kuvattavan järjestelmän toiminnallisuuden
  - Aspektimallit määrittelevät järjestelmän poikkileikkaavia (*cross-cutting)* piirteitä
    - Useasti ei-toiminnallisisa piirteitä kuten turvallisuus
    - Joukko sidontotakaavaimia *(point-cut),* jotka määrittelevät mihin kohtaa primäärimallia aspektit tulee sitoa
      - Vrt. mallien kudonta
      - Sidontakaavaimissa käytetään tyypillisesti *mallielementtimuuttujia*, jotka sidonnan aikana instantioidaan primäärimallin elementeillä
    - Joukko yhdistämissääntöjä *(advice*), jotka määrittelevät kuinka aspektit sidotaan primäärimalliin

Aspektien mallintaminen

- Sidontakaavaimet ja yhdistämissäännöt määritellään primäärimallin konkreettista syntaksia käyttäen
  - Aspektimallin ja primäärimallin tulee siis olla konformantteja saman metamallin kanssa
  - Esimerkkeinä aspektitmallit luokka- ja sekvenssikaavioina määriteltyinä

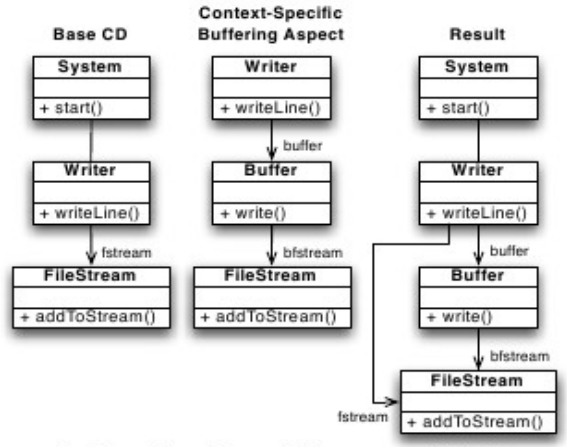


Figure 1: Merging Class Diagrams with Kompose

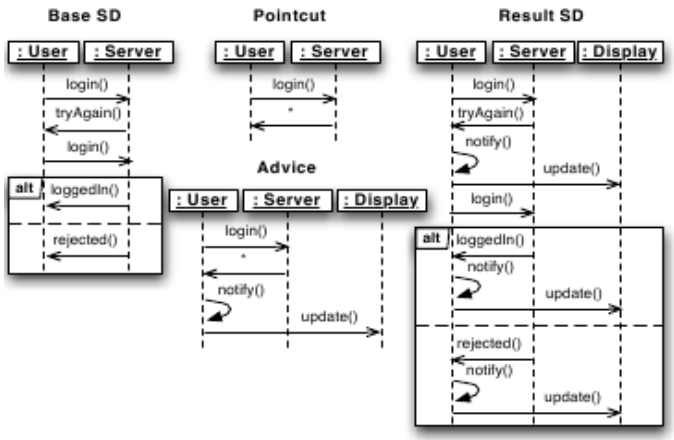


Figure 2: Sequence Diagram Weaving Example

Kuvat: Kienzle, J., Al Abed, W., and Klein, J. 2009. Aspect-oriented multi-view modeling. In Proceedings of the 8th ACM international Conference on Aspect-Oriented Software Development (Charlottesville, Virginia, USA, March 02 - 06, 2009). AOSD '09. ACM, New York, NY, 87-98. DOI= http://doi.acm.org/10.1145/1509239.1509252

Uudelleenkäytettävien aspektimallien määrittely

- Aspektit voivat vaikuttaa useaan näkökulmaan kerrallaan
  - Yksittäinen aspektimalli (esim. luokkamalliin pohjautuen) ei ole tällöin itsenäinen, uudelleenkäytettävä artefakti
  - Tarvitaan joukko aspektimalleja tarvittavien muutosten (rakenteellisten, käyttäytymisen) aikaansaamiseksi → näkökulmien välisen konsistenssin tapaaminen vaikeutuu

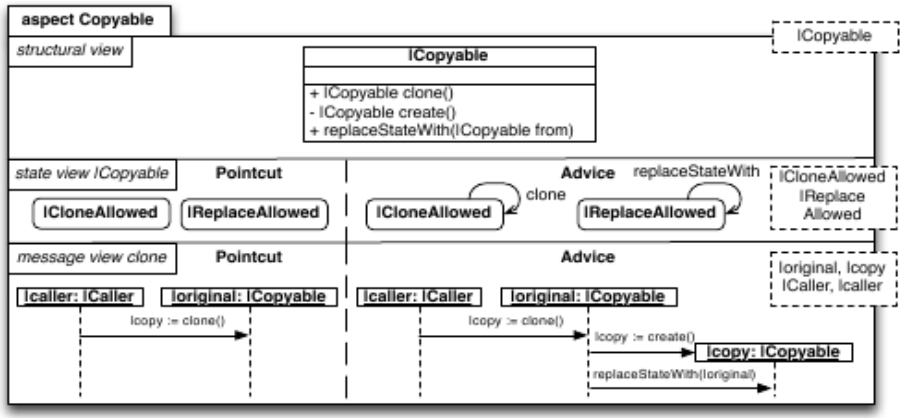


Figure 3: The *Copyable* Aspect Model

  - Uudelleenkäytettävien aspektimallien tulisikin määritellä kaikki tarvittavat efektit modulaarisesti
- Eräs lähestymistapa on RAM (*Reusable Aspect Model)*
  - RAM aspektimallit määrittelevät rakenne-, tila- ja viestintänäkymät
    - Näkymät hyödyntävät mallielementtimuuttujia (kuvassa merkitty ”|” etuliitteellä)
  - Rakennenäkymä (*structural view)* määritellään UML:n luokkakaavioina
    - Attribuutit ja metodit
  - Tilanäkymä (*state view)* määritellään UML:n tilakaavioina
    - Rakennenäkymässä määritellyille luokille määritellään käyttäytymisprotokolla
      - Kuvaa missä tiloissa luokka voi olla
  - Viestintänäkymä (*message view)* määritellään UML:n sekvenssikaavioina
    - Määrittelee viestien vaihdon julkisen metodin kutsumisen yhteydessä

Kienzle, J., Al Abed, W., and Klein, J. 2009. Aspect-oriented multi-view modeling. In Proceedings of the 8th ACM international Conference on Aspect-Oriented Software Development (Charlottesville, Virginia, USA, March 02 - 06, 2009). AOSD '09. ACM, New York, NY, 87-98. DOI= http://doi.acm.org/10.1145/1509239.1509252

Aspektimallien sidonta

- Aspektien sidonnassa sovelletaan elementtien sovitusta (*matching)* ja mallien kudontaa
  - Primäärimallista etsitään sidontakaavaimen mukaista rakennetta
  - Sidontakaavaimeen sovitettu primäärimallin osioon lisätään yhdistämissäännössä määritellyt elementit
- Ennen sidontaa täytyy aspektin riippuvuudet muista aspekteista ratkaista
  - Esimerkkikuvassa aspekti *Z* riippuu aspektista *A*, joka riippuu aspekteista...
  - Aspektista täytyy muodostaa siis itsenäinen aspektimalli
  - Itsenäinen aspektimalli muodostetaan sitomalla aspekteja toisiinsa
    - Erityisesti aspektimalleissa määritellyt mallielementtimuuttujat instantioidaan

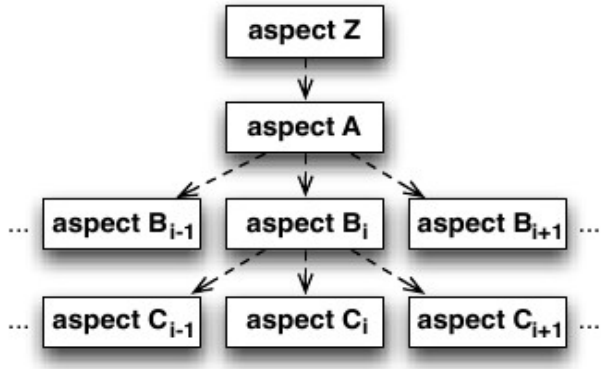


Figure 5: Aspect Dependencies

RAM lähestymistavan konsistenssisäännöt

- Jokaisen aspektimallin itsessään tulee olla konsistentti
  - Jokaiselle rakennenäkymän luokalle tulee olla määriteltynä tilanäkymä
  - Tilanäkymän tulee määritellä protokolla jokaiselle julkiselle metodille
  - Jokaiselle rakennenäkymän julkiselle metodille tulee olla määriteltynä vastaava viestintänäkymä
  - Tila- ja viestintänäkymien tulee olla ristiriidattomat
    - Viestintänäkymän käyttäytymisen tulee olla yhteneväistä siinä käytettävien luokkien protokollien kanssa
- Toisistaan riippuvien aspektimallien tulee olla keskenään konsistentteja
  - Aspektimalleissa vaaditut mallielementtimuuttujat pitää pystyä instantioimaan
- Itsenäisen aspektimallin ja primäärimallin tulee olla keskenään konsistentteja
  - Itsenäisessä aspektimallissa vaaditut mallielementtimuuttujat pitää pystyä instantioimaan eli sitomaan primäärimallin elementteihin

Kurssin luento-osuuden yhteenveto

- Ekosysteemit ja palveluperustaiset järjestelmät
- Palvelut ja palvelukoosteet
- Palveluekosysteemien elinkaaret
- Palvelusuuntautuneen ohjelmistotuotannon lähestymistavat
- Ei-toiminnalliset piirteet
- Väliohjelmistojen rooli
- Palvelusuuntautuneen ohjelmistotuotannon prosessit
- Malliperustaisen palvelusuuntautuneen ohjelmistotuotannon perusteet

Yhteenveto: palveluekosysteemit ja palveluperustaiset järjestelmät

- Palveluekosysteemi koostuu joukosta autonomisia toimijoita
  - Organisaatiota ja yksilöitä
  - Tavoitteena tehokas yhteistyöverkostojen muodostaminen
- Ekosysteemi määrittelee
  - Toimijoiden välisen yhteistyön mahdolliset muodot ja niihin liittyvät käsitteet
  - Elinkaaret, joita toimijoiden tulee ekosysteemissä noudattaa
- Palveluperustainen järjestelmä
  - Tuottaa palveluekosysteemissä toimimiseen tarvittavat infrastruktuuripalvelut
  - Perustana palvelusuuntautunut lähestymistapa
    - SOA arkkitehtuurityyli
    - Palvelusuuntautuneet ohjelmistotuotantomenetelmät
    - Palveluperustainen väliohjelmisto

Yhteenveto: palvelut ja palvelukoosteet

- Palvelun määritelmä
  - Täyttää käyttäjätarpeen
  - Hyvin määritelty rajapinta
  - Itsensä kuvaava
  - Avoimesti saatavilla
- Palveluabstraktiot
  - Liiketoiminnallinen palvelu: täyttää liiketoiminnallisen, asiakkaan tarpeen
  - Komponenttipalvelu: täyttää liiketoiminnalisten palveluiden tarpeita
  - Teknologiset palvelu: toteuttaa komponenttipalvelun
- Palvelukoosteet täyttävät jonkin yhteistyötarpeen
  - Koreografiat määrittelevät yhteistoiminnan muodon
    - Roolit, niiden ominaisuudet ja roolien väliset interaktiot
  - Orkestraatiot toteuttavat yhteistoimintaroolin mukaiset vaatimukset
  - Koostamiseen ja koosteiden määrittelyyn useita erilaisia lähestymistapoja
    - Dynaaminen vs. Staattinen, Imperatiivinen vs. Deklaratiivinen, ...
    - Koostamistapa riippuu ekosysteemin ja palveluperustaisen järjestelmän ominaisuuksista

Yhteenveto: palveluekosysteemien elinkaaret

- Palveluperustaisten järjestelmien elinkaaret
  - Palveluiden elinkaaret
  - Yhteistyöverkostojen elinkaaret
- Esimerkkejä muista elinkaarista palveluekosysteemeissä
  - Toimijoiden elinkaaret
  - Tuotteiden elinkaaret
  - Tietämyksen hallinnan elinkaaret
  - Ekosysteemin hallinnalliset prosessit

Yhteenveto: palvelusuuntautuneen ohjelmistotuotannon lähestymistavat

- Toimijoiden lähestymistavat palvelutuotantoon
  - Liiketoiminta- vs. teknologialähtöisyys
  - Perinnejärjestelmien (*legacy)* rooli!
- Kolme peruslähestymistapaa
  - Top-down: liiketoimintalähtöinen tapa
    - Tuottaa liiketoiminnallisesti ”parhaimmat” palvelut
  - Bottom-up: teknologialähtöinen tapa
    - Puhtaasti pohjalta lähtemällä ei ole mahdollista tuottaa mielekkäitä liiketoiminnallisia palveluita
      - Elleivät sitten teknologiset palvelut = liiketoiminnalliset palvelut
  - Meet-in-the-middle: kompromissi edellisten välillä
    - Ottaa huomioon erityisesti perinnejärjestelmät
    - Business-IT alignment

Yhteenveto: ei-toiminnalliset piirteet

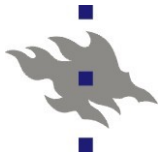
- Ei-toiminnalliset piirteet tuovat jotain lisäarvoa palvelun toiminnallisuuteen
  - Voidaan käyttää palveluiden etsimis- ja valintakriteereinä
- Esimerkkejä
  - Salattu tiedonsiirto; Kolmannen osapuolen varmentamat interaktiot; Saatavuus; Vasteaika; Luotettavuus; ...
- Ei-toiminnallinen piirre
  - Määrittelee lisäarvoa tuottavan piirteen semantiikan ja arvojoukon
- Ei-toiminnallinen ominaisuus
  - Ei-toiminnallisen piirteen ”instanssi”
- Ei-toiminnallisten piirteiden interaktiot
  - Toiminnallisten piirteiden kanssa
  - Toisten ei-toiminnallisten piirteiden kanssa
  - Abstraktiotasojen sisällä vs. välillä
- Ei-toiminnallisten piirteiden hallinta
  - Tunnistaminen, mallintaminen, realisointi

Yhteenveto: väliohjelmistojen rooli

- Palveluperustaisten järjestelmien toiminta perustuu väliohjelmistoihin
- Väliohjelmisto
  - Väliohjelmistoalusta toteuttaa yhtenäisen, jaetun yhteistoimintaympäristön
  - Ohjelmointirajapinta
  - Asiakas- ja palvelunedustajat ja -tyngät
- Toteuttavat ekosysteemin toimintaan vaadittavat infrastruktuuripalvelut
- Tukevat erityisesti
    - Elinkaarten hallintaa
    - Ei-toiminnallisten piirteiden hallintaa ja realisointia
    - (Hajautettuja) ohjelmistotuotantoprosesseja

Yhteenveto: palvelusuuntautuneen ohjelmistotuotannon prosessit

- Ohjelmistotuotantoprosessi
  - Määrittelee kuka tekee, mitä, milloin, kuinka paljon,...
- Metodi
  - Määrittelee miten jokin aktiviteetti prosessista tulee suorittaa
- Metodologia
  - Joukko metodeja joka kattaa ja yhdistää tuotantoprosessin eri vaiheet
- Palvelusuuntautunut ohjelmistotuotantoprosessi nivoutuu ekosysteemin elinkaariin sekä palvelutuotannon lähestymistapoihin
  - Ekosysteemin elinkaarten vaiheiden vaatimukset vaikuttavat ohjelmistotuotantoprosessin aktiviteettien sisältöön
    - Minkälaisia rooleja ja toimintoja tuetaan tai edellytetään?
    - Mitä metainformaatiota on saatavissa ekosysteemistä? Missä vaiheessa?
    - Mitä metainformaatiota tulee julkistaa ekosysteemissä toimiakseen?
  - Palvelutuotannon lähestymistavat
    - Ohjaavat ohjelmistotuotantoprosessien vaiheiden järjestystä ja aktiviteettien keskinäisiä riippuvuuksia
    - Määrittelevät metodien sisältöä: minkälaisia kriteerejä käytetään esimerkiksi liiketoimintapalveluiden identifioimiseen?



Yhteenveto: malliperustaisen palvelusuuntautuneen ohjelmistotuotannon perusteet

- Käytiin läpi malliperustaisen ohjelmistotuotannon (MDE) perusteet
  - Mallit
  - Mallimuunnokset
- Esiteltiin muutamia palvelusuuntautuneen ohjelmistotuotannon mallinnusmenetelmiä
  - UMM
  - SoaML
  - UML4SOA
- Tähän asiaan syvennytään harjoitustyössä