

Loeng 5. Kontseptuaalmodelleerimine ja arendustsüklid

Eesmärgid:

1. Anda ülevaade iteratiivse arendusprotsessi ehitusfaasist.
2. Anda ülevaade kontseptuaalmudelist.
3. Käsitleda kontseptuaalmodelleerimise probleeme iteratiivse arendusprotsessi loogikas.

Kava

1. Iteratiivse arendusprotsessi ehitustsükli ülevaade ja käivitamine
2. Analüüsifaasi ülevaade ehitustsükli
3. Kontseptuaalmodelleerimine ehitusfaasis

Arendustsükli käivitamine

Arendustsükli alguse tegevused on seotud projektijuhtimisega (arendustsükli plaani täpsustamine).

Samal ajal toimub diagrammide ja dokumentatsiooni (mis on pärit eelmisest arendustsüklist) sünkroniseerimine programmikoodi tegeliku seisuga, kuna viimase tsükli kodeerimise faas tekitab erinevuse disaini (mudelite) ja koodi vahel.

Seejärel sisenetakse analüüsi faasi, kus uuritakse täpsemalt antud arendustsükli probleeme. Analüüsifaasi üheks tähtsamaks tegevuseks on kontseptuaalmudeli arendamine.

Analüüsifaasi tegevused ning sõltuvused

Arendustsükli analüüsifaasi põhitegevused:

- Oluliste Use Case 'ide defineerimine (laiendatud kirjelduste ehk stsenaariumide vormis ning juhul, kui seda seni tehtud pole)
- Use Case diagrammide peenendamine
- Kontseptuaalmudeli peenendamine (jäme kontseptuaalmudel võis olla tehtud planeerimisfaasis)
- Mõistete sõnastiku uuendamine (pidev protsess)
- Süsteemi jadadiagrammide loomine
- Süsteemi operatsioonide lepingute defineerimine
- Olekudiagrammide loomine (mittekohustuslik)

Üks varasemaid tegevusi arendustsükli on kontseptuaalmudeli loomine jooksva arendustsükli use case 'ide jaoks. See tegevus sõltub use case 'ide ja teiste dokumentide olemasolust, kust on võimalik mõisteid ja objekte leida. Kõige efektiivsem on arendada kontseptuaalmudelit paralleelselt use case 'idega.

Kontseptuaalmudeli ehitamine

Eesmärgid:

- Identifitseerida jooksva arendustsükli vajadustega seotud kontseptid
- Luua esialgne kontseptuaalmudel
- Teha vahet korrektsete ja mittekorrektsete atribuutide vahel
- Lisada spetsifikatsiooni kontsepte, kus vajalik

Kontseptuaalmudel on keskne osa (tuum) domeenimudelisis (analüüsi mudelis).

Kontseptuaalmudel illustreerib olulisi mõisteid probleemvaldkonnas. Kontseptuaalmudel on kõige tähtsam OO analüüsi tulemus (use case 'id on vajaduste analüüsi põhitulemus, mis pole tegelikult objektorienteeritud, vaid keskendub probleemvaldkonna protsessivaatele).

Rikkaliku objektide või kontseptide hulga identifitseerimine on OO analüüsi tuum, mis tasub end ära disaini ja realiseerimise faasis.

UMLis saab kasutada staatilist struktuuridiagrammi (klassidiagrammi üks variatsioon, kus operatsioone ei näidata) kontseptuaalmudeli esitamiseks.

Oluline moment kontseptuaalmudeli kvaliteedi hindamisel on asjaolu, et ta peab esitama reaalse maailma asju, mitte tarkvara komponente.

Kontseptuaalmudeli näide.

Kontseptuaalmudel is näidatakse:

- Kontseptid (probleemvaldkonna olulised mõisted või objektid)
- Seosed (assotsiatsioonid) mõistete vahel
- Kontseptide atribuudid

Kontseptuaalmudel aitab selgitada mõistete sõnastiku terminoloogiat, kuid ei asenda sõnastikku (graafiline ja tekstiline esitus täiendavad teineteist).

Kontseptil on

sümbol: näiteks *Müük* (*kuupäev, kellaeg*) tekstilises või graafilises esituses

definitsioon (*intension*), näiteks “*Müük tähistab müügitoimingut. Tal on kuupäev ja kellaeg.*”

Laiendus (*extension*) ehk näited selle kontsepti kohta (*müük-1, müük-2, müük-3,...*)

Põhiline erinevus objektorienteeritud ja struktuurse analüüsi vahel: Probleemvaldkond dekomponeeritakse pigem kontseptide (objektide) järgi kui funktsioonide (protsesside) järgi.

Use Case modelleerimine kui funktsionaalne analüüs on OO modelleerimise suhtes siiski väga oluline taust.

Kuidas leida kontsepte ?

Kontseptuaalmudeli ülespetsifitseerimine paljude peenendatud mõistetega on parem, kui alaspetsifitseerimine ehk mõistete puudujääk. Ei tasu mõelda, et kontseptuaalmudel on parem, kui temas on vähem kontsepte. Pigem on asi vastupidi.

Loomulik on see, et ei suudeta kohe avastada kõiki olulisi kontsepte, vaid nende avastamine ja lisamine jätkub hilisemas atribuutide ja seoste modelleerimise käigus ning ka disainifaasis.

Ära jäta kontsepti kõrvale lihtsalt sellepärast, et vajadustes pole näha selget nõuet vastavat informatsiooni salvestada, või et kontseptil ei ole atribuute. Täiesti korrektsed on atribuutideta kontseptid või kontseptid, millel on puhtalt käitumislik roll, kuid puudub inforoll.

Kontseptide leidmine kategooriate (liikide) loetelu alusel

“Käegakatsutavad” ehk füüsilised objektid

Rollid

Organisatsioonid ja nende üksused

Asukohad

Protsessid / Tegevused

Sündmused

Spetsifikatsioonid

...

Kontseptikandidaatide leidmine vajaduste ning use case 'ide kirjeldustest nimisõnade alusel

Kontseptuaalmodelleerimise sammud

1. Koosta kontseptikandidaatide loetelu, kasutades kategooriate loetelu ning / või nimisõnade identifitseerimist vajaduste kirjeldustest
2. Joonista kontseptid kontseptuaalmudelisse
3. Lisa seosed oluliste seoste kohta, mida on tarvis mäletada
4. Lisa atribuudid, mis on vajalikud infovajaduste rahuldamiseks.

Nimed olgu probleemvaldkonna omad.

Ärge modelleerige kontsepte atribuutidena. Näiteks lennu sihtpunkt on kontsept iseseisvate atribuutidega, mitte lennu atribuut, järelikult modelleerida seose kaudu.

Seosed

(NB! Järgnev osa on pärit loengumaterjalide hulka lisatud dokumendist “Klassidiagrammid.doc”, mis tasuks tervikuna läbi lugeda)

Klassidiagramm sisaldab klasse ja nendevahelisi seoseid. Liigi järgi jagunevad seosed:

- Assotsiatsioonid
- Üldistused (generalisation)
- Sõltuvused (dependency)
- Peenendused (refinement)

Assotsiatsioon on ühendus klasside vahel ning ühtlasi nende klasside objektide vahel.

Üldistus on seos üldisema ja spetsiifilisema elemendi vahel., kus spetsiifilisem element võib sisaldada üksnes lisainformatsiooni. Spetsiifilisema elemendi eksemplari võib kasutada kõikjal, kus üldisem element on lubatud.

Sõltuvus on seos ühe sõltumatu ja ühe sõltuva elemendi vahel. Sõltumatu elemendi muudatus mõjutab sõltuvat elementi, mitte vastupidi.

Peenendus on seos kahe sama asja kirjelduse vahel, mis omavad erinevat abstraktsioonitaset.

Assotsiatsioonid

Assotsiatsioon on klassidevaheline ühendus, semantiline ühendus seoses osalevate klasside objektide (eksemplaride) vahel.

Assotsiatsioon on tavaliselt kahe-suunaline, s.t. kui objekt on seotud teise objektiga, siis mõlemad objektid “tunnevad teineteist”, “on ühendatud” jne.

Tavaline assotsiatsioon

Tavaline assotsiatsioon on lihtsalt ühendus klasside vahel. Joonistatakse pidevjoonena kahe klassi vahel. Assotsiatsiooni nimi (tavaliselt tegusõna, võib olla ka nimisõna) kirjutatakse seosjoone lähedusse. Seosenimed peavad tulenema probleemvaldkonnast, nagu klassinimedki.

Võimalus kasutada navigeeritavaid seoseid, lisades seose lõpuotsa noole. Nool näitab, et seost saab kasutada ainult noole suunas. Assotsiatsioon võib omada kahte nime, kummaski suunas ühte. Nime suunda näidatakse väikese täidetud kolmnurgaga nime ees või taga, sõltuvalt suunast. Seost on võimalik lugeda lausena, näiteks:

“Autor kasutab arvutit”.

Assotsiatsiooni osaks on ka seoses osalevate objektide (eksemplaride) arv (multiplicity): Autol on 1 või enam omanikku ja isik võib omada 0 või enamat autot.

Arvukuse (multiplicity) variandid:

Nullist üheni (0..1)

Null või enam (0..*) ehk (*)

Üks või enam (1..*)

Kaks (2)

(5..11)

(1, 4, 6, 8..12)

Vaikimisi arvukus on (1)

Mudel peab olema asjaosalistele arusaadav, tõlgitav loomulikku keelde:

- Kindlustusfirma omab kindlustuslepinguid, mis viitavad ühele või enamale kliendile.
- Klient omab kindlustuslepinguid (0 või enam), mis viitavad ühele kindlustusfirmale

- Kindlustusleping sõlmitakse kindlustusfirma ning ühe või enama kliendi vahel. Kindlustusleping viitab nii kliendile (klientidele) kui ka kindlustusfirmale.
- Kindlustusleping väljendatakse (nullis või ühes) kindlustuspoliisis (kirjalik kindlustusleping)
- Kindlustuspoliis viitab kindlustuslepingule

Kui helistate kindlustusfirmasse ja kindlustate oma auto, tekib kindlustusleping, kuid mitte kindlustuspoliis. Kindlustuspoliis tehakse hiljem ja saadetakse kliendile. Niisugust reaalsust on oluline ka modelleerida. Kui modelleeriksime kindlustusäri poliisikeskselt (mitte lepingukeskselt), võib tekkida suuri probleeme. Näiteks kui klient kindlustab oma auto ja veidi hiljem teeb avarii, kui kindlustuspoliisi veel pole, kuid suuline kokkulepe kindlustusagendiga saavutatud.

Mis saab siis, kui juurutatakse uut tüüpi kindlustuspoliisid (kindlustamine Web-is), või üldse äri (ärireeglite) muutumise korral? Modelleerides tegelikku elu, lisab Web kindlustus lihtsalt vastava klassi (Web kindlustuspoliis), millel võib olla teistsugune käitumine, kui tavalisel poliisil (n. kliendid saavad neid ise muuta ja muudatused kanduvad automaatselt kindlustuslepingusse; kindlustuspoliisi saab saata e-maili teel otse kliendile).