

Kehittää
ohjelmointitehtävien
ratkaisemisessa
tarvittavia
metakognitioita!

SPESIFIKAATIO

spesifikaatio	→	implementaatio
“mitä tehdään?”	→	“miten toteutetaan?”

1. Are we building the right product?
2. Are we building the product right?

VAATIMUSLUOKITTELU

Toiminnalliset vaatimukset: mitä ohjelma tekee?

laskutoimitukset, rakenne, pohjatiedot, rajapinnat, ...

Ei-toiminnalliset vaatimukset: miten ohjelma tekee?

käytettävyys, tekniset ratkaisut, suorituskyky, luotettavuus, turvallisuus, siirrettävyys, ...

Rajoitteet: mistä ei saa tinkiä?

hinta, lainsäädäntö, standardit, aikataulut, työkalut, ...

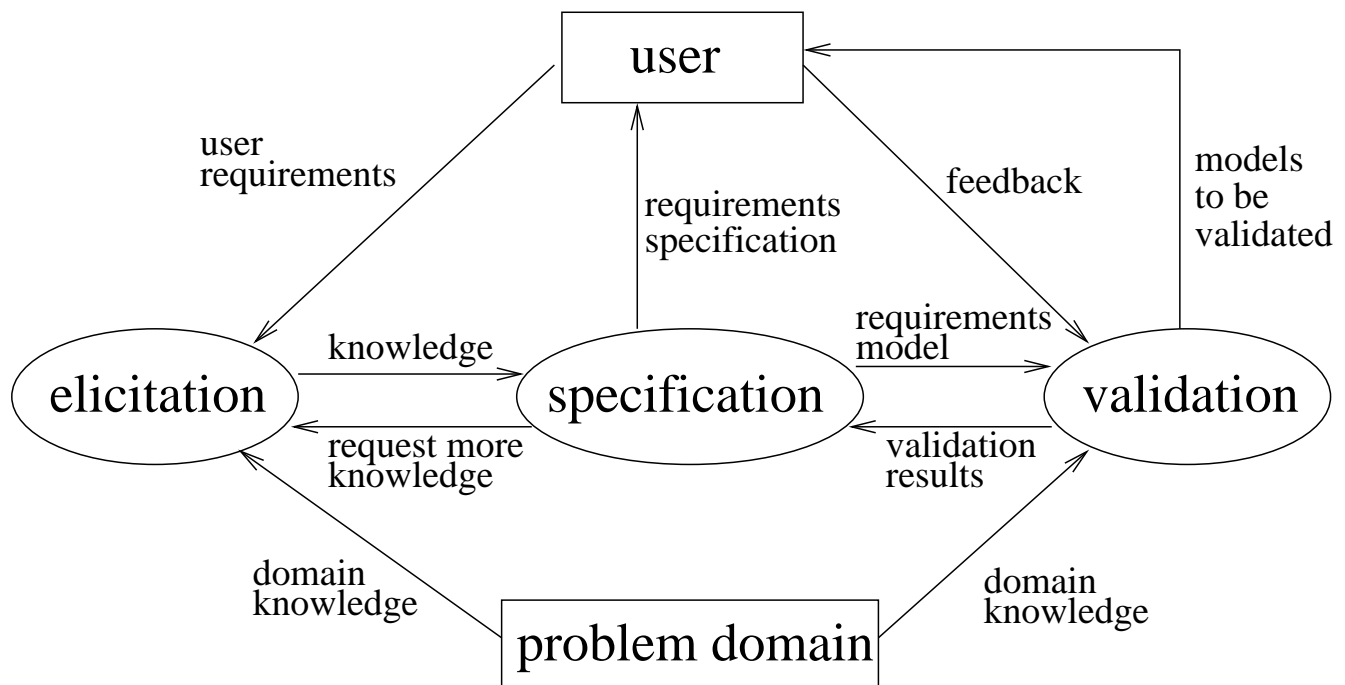
VAATIMUSTEN KERÄÄMINEN:

i): käyttäjäkyselyt

ii): olemassaolevien ratkaisujen mallit

iii): asiakasjärjestelmän piirteiden selvittäminen

iv): prototyypitys



VAATIMUSTEN LAATUKRITEEREJÄ:

oikeellisuus (*correctness*)

yksiselitteisyys (*unambiguity*)

kattavuus (*completeness*)

johdonmukaisuus (*consistency*)

luokiteltavuus (*rankness*)

todennettavuus (*verifiability*)

muutettavuus (*modifiability*)

jäljitettävyys (*traceability*)

KÄYTTÖTAPAU

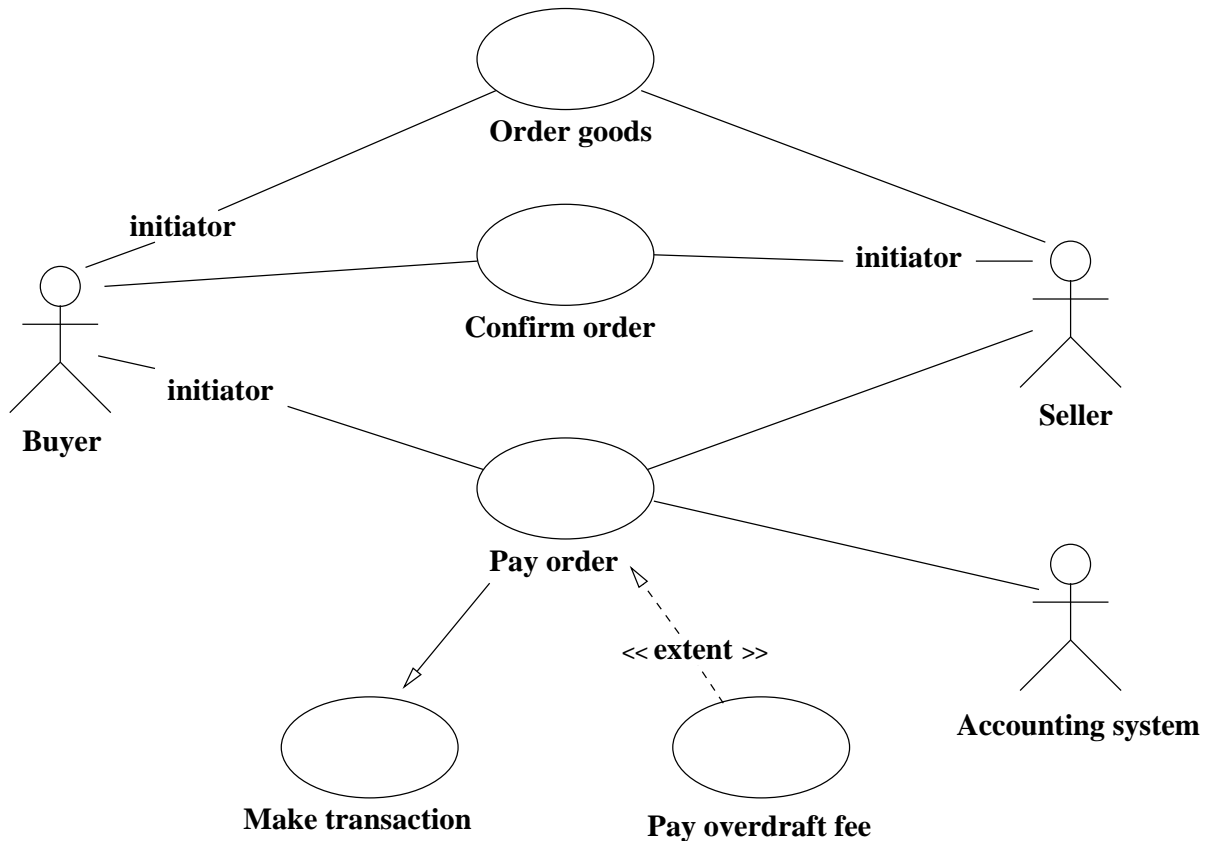
=

SANALLINEN KUVAUS

+

KÄYTTÖTAPAU

Käyttötapauskaavio:



Kaavion rakenne ja käytetyt merkinnät:

Osa:	Symboli:	Merkitys:
<i>käyttötapaus</i>	ellipsi&nimi	järjestelmän toiminnan kuvaus
<i>tekijä (aktori)</i>	tikku-ukko	käyttäjän, laitteen tai toisen järjestelmän suorittava rooli
<i>kommunikointisuhde</i>	yhtenäinen viiva	aktorien ja käyttötapauksen liittäminen toisiinsa (yhteydenottaja = initiator)
<i>yleistyssuhde</i>	yhtenäinen nuoli	nuolen alkupään käyttötapaus yleistyy nuolen loppupään vastaavaksi
<i>sisältymissuhde</i>	katkoviivanuoli	nuolen alkupään käyttötapaus sisältyy nuolen loppupään tapaukseen
<i>laajennussuhde</i>	katkoviivanuoli	nuolen alkupään käyttötapaus laajentaa nuolen loppupään tapausta

SPESIFIKAATIOT JA KUVAUSKIELET

- spesifi kaatioilla monia tehtäviä (ongelmat, kommunikointi, dokumentointi; lähtökohta V&V:lle, implementoinnille ja ylläpidolle sekä laajennuksille)
- spesifi kaatio esitetään kuvauskielen avulla
- ei yhtenäistä standardia

KUVAUSKIELTEN RYHMITTELY

Epäformaalit menetelmät
<ul style="list-style-type: none">• luonteeltaan kuvailevia• esim. seinätaulutekniikat• eksaktia kuvausta vaikea tehdä
Puoliformaalit menetelmät
<ul style="list-style-type: none">• semantiikka ei tarkasti määrätty• mahdollisuuksia määrätä formaalisuuden aste• esimerkiksi OMT(++)-menetelmä
Formaalit menetelmät
<ul style="list-style-type: none">• tarkka syntaksi, eksakti spesifi kaatio• perusteina esim. joukko-oppi ja matemaattinen logiikka• esimerkkejä: Z, B, LOTOS, ESTELLE, SDL, VDM

Nimi	Yleiskuvaus
Päätöstaulu	<i>Decision Table</i> . Matriisimuotoinen esitys tilasiirtymien logiikasta.
ER	<i>Entity-Relationship Model</i> . Tietoalkioiden koostumusta ja niiden välistä rakennetta esittävä diagrammitekniikka.
Vuokaaviot	<i>Flowcharts</i> . Diagrammitekniikoiden perhe, joiden avulla kuvataan kontrollin muutoksia ohjelman eri osissa ja niiden välillä.
Tilakone	<i>FSM = Finite State Machine</i> . Järjestelmän kuvaaminen sen tilojen ja niiden välisten siirtymien (= tilasiirtymien) avulla.
JSD	<i>Jackson System Development</i> . Järjestelmän kuvaaminen rakenteellisten (<i>Jackson</i>) diagrammien (sekvenssi, iteraatio, valinta) ja pseudokoodin avulla.
JSP	<i>Jackson Structured Programming</i> . JSD:tä edeltänyt <i>programming-in-the-small</i> tekniikka tietorakenteiden pohjalta tehtävälle kuvaukselle.
LCP	<i>Logical Construction of Programs/Warnier-Orr method</i> . JSP:tä vastaava myös tietorakenteisiin perustuva tekniikka.
Meta IV	Mallipohjainen VDM-menetelmän käyttämä kuvauskieli.
OBJ	Algebrallinen kuvausmenetelmä.
OOD	<i>Object-oriented Design</i> .
PDL	<i>Program Design Language</i> . Rakenteiseen luonnolliseen kieleen perustuva tapa esittää ohjelmaa eri abstraktiotasoilla.
Petriverkot	<i>Petri nets</i> . Graafinen, erityisesti rinnakkaisprosessien esittämiseen käytetty menetelmä, jossa järjestelmä kuvataan joukkona tiloja ja niiden välisiä siirtymiä. Tilasiirtymät tapahtuvat siirtymäehtojen "lauetessa", jolloin eri prosesseja voidaan synkronoida keskenään (lisätietoja TTKK).
SA/SD	<i>Structured Analysis/Design</i> . Tietovuon kuvaustekniikka.
SA/WM	<i>Ward-Mellon Extension of Structured Analysis</i> . SA:n laajennus reaaliaikaisuuden huomioonottamiseksi.
SADT	<i>Structured Analysis and Design Technique</i> . Graafinen kuvauskieli erityisesti hierarkisten suhteiden esittämiseksi.
SSADM	<i>Structured Systems Analysis and Design Method</i> . Edellisiä käyttämä ohjelmistosuunnittelumenetelmä.
Tilakaaviot	<i>Statecharts</i> . Tilansiirtokaaviotekniikat, joita on usein laajennettu rinnakkaisuuden ja eri abstraktiotasojen kuvaamista varten.

FORMAALI KUVAUSKIELI

Formaali kuvauskieli on kolmikko (Syn, Sem, Sat) , missä Syn ja Sem ovat joukkoja ja $Sat \subseteq Syn \times Sem$ on niiden välinen relaatio. Joukkoa Syn sanotaan kuvauskielen syntaktiseksi alueeksi (*syntactic domain*), joukkoa Sem sen semanttiseksi alueeksi (*semantic domain*) ja relaatiota Sat sen merkitysrelaatioksi (*satisfies relation*).

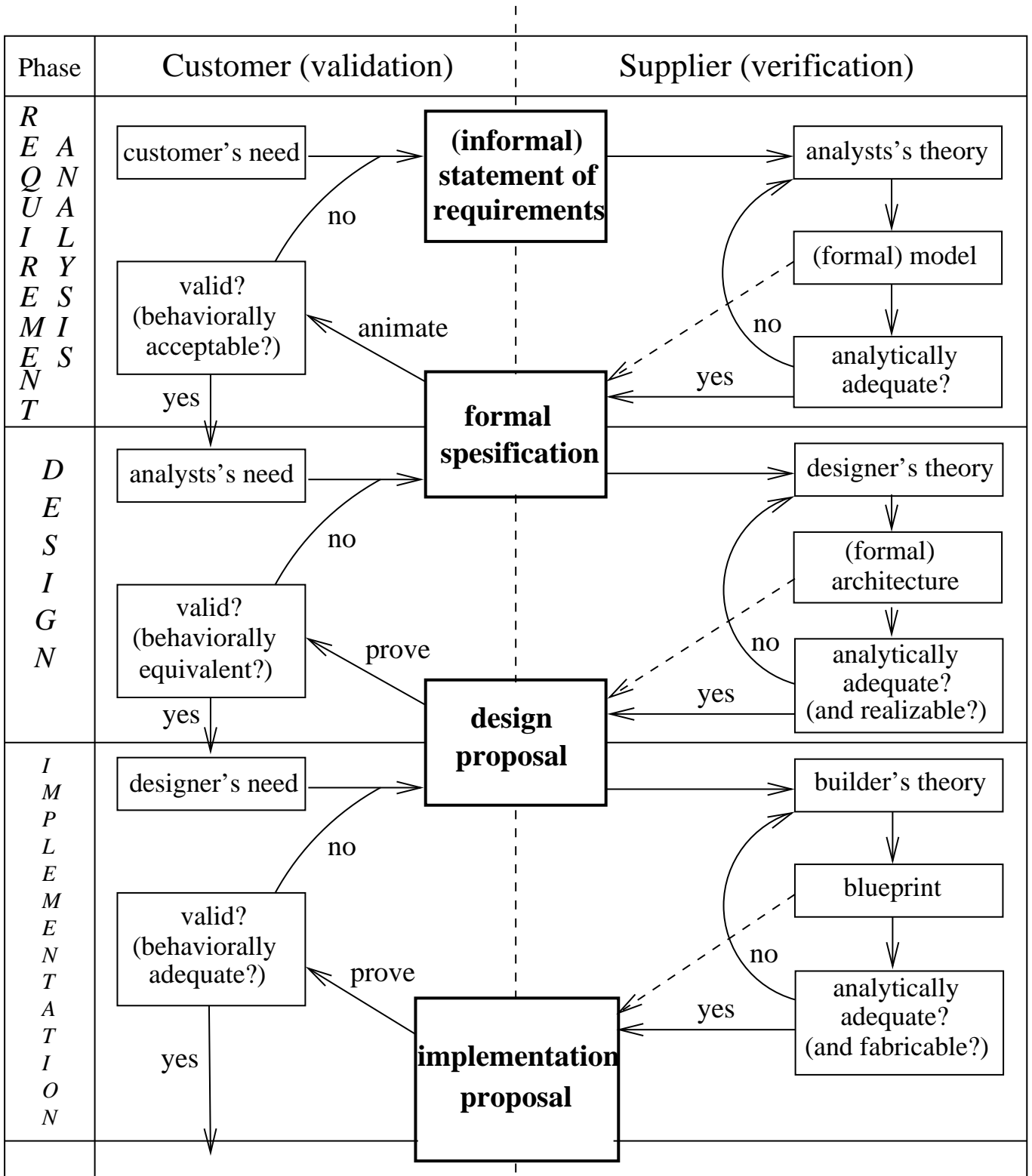
FORMAALIEN MENETELMIEN LUOKITTELU

Mallipohjaiset menetelmät: joukko-opillinen mallistruktuuri, jossa toiminta kuvataan joukolla loogisia rakenteita ja näiden välisiä operaatioita. Esim. Z, B ja VDM.
tiedonhallintarakenteet (esim. tietokannat).

Ominaisuuspohjaiset menetelmät: järjestelmän epäsuora kuvaus sille asetettujen (toiminnallisten) ominaisuuksien (rajoitteiden) kautta. Voidaan edelleen jakaa

Algebrallisiin: ohjelman käyttämien tietotyyppien esitys algebrallisina kokonaisuuksina, esim. ESTELLE, LOTOS, SDL, CSP, Petriverkot, Raise, jne.
rinnakkaisjärjestelmät&ajastus (tietoliikenne)

Aksiomaattisiin: ohjelman funktionaalisuuden määrittäminen esi- ja jälkiehtojen sekä näiden välisen loogisen muunnoksen avulla. esim. Floyd-Hoare'n loogikka, Dijkstran WP-calculus, Leslie Lamportin TLA, jne.
perintöä ohjelmien oikeaksi todistamisesta



SPESIFIKAATIOESIMERKKI (I)

Ongelman kuvaus: Tarkastellaan pientä kirjastotietokantaa seuraaville transaktioille:

1. Lainaa kirjan kopio. Palauta kirjan kopio.
2. Lisää kirjan kopio kirjastotietokantaan. Poista kirjan kopio tietokannasta.
3. Muodosta lista tietyn kirjoittajan kirjoista tietyltä aihealueelta.
4. Muodosta lista tietyn lainaajan lainaamista kirjoista.
5. Palauta tieto siitä, kuka lainaaja on viimeksi lainannut tietyn kirjan kopion.

Kirjastotietokantaa käyttävävät kahden tyyppiset käyttäjät: henkilökunta ja lainaajat. Transaktiot 1, 2, 4, 5 on rajoitettu vain henkilökunnan käyttöön, paitsi että tavalliset lainaajat voivat suorittaa transaktion 4 selvittääkseen omat lainansa. Tietokannan toimintaa rajoittavat seuraavat ehdot:

- Kaikkien kirjojen kopioiden tulee olla lainattavissa tai lainassa.
- Mikään kirjan kopio ei saa olla samaan aikaan lainattavissa ja lainassa.
- Yhdellä lainaajalla saa olla korkeintaan ennalta määrätty määrä lainoja.

SPESIFIKAATIOESIMERKKI (II)

Wing jaottelee spesifi kaatiomenetelmiä

- **orientaation:** operaatio- vai tieto-orientoitunut
- **modulaarisuuden:** tukeeko vai ei
- **luettavuuden:** diagrammeja vai ei

mukaisesti. Hän vertailee toisiinsa myös annettujen spesifi kaatioiden

- **monimerkityksellisyyttä:** Ovatko kirjasto, käyttäjä, kirja/kopio, saatavuus ja viimeiseksi lainattu -käsitteet yksikäsitteisesti määritelty?
- **epätäydellisyyttä** alustuksen, puuttuvien operaatioiden, virheiden käsittelyn, puuttuvien rajoitteiden, tilanmuutosten spesifi oinnin ja ei-toiminnallisen käyttäytymisen (UI) suhteen.

Loppupäätelmänä Wing toteaa, että *sovellusalueen kokonaisvaltainen huomioonottaminen* tekee yksinkertaisestakin järjestelmästä monimutkaisen. Spesifi kaatioiden puutteita/ongelma-alueita olivat mm.

- rinnakkaisuuden hallinta
- luotettavuus
- virheistä toipuminen
- turvallisuus
- versionhallinta, vaihedokumenttien muoto ja projektin ohjaus