

Lauri Pesonen

# SUUNNITTELURATIONAALI

Kandidaatintyö  
Tekniikan ja luonnontieteiden tiedekunta  
Tutkijatohtori Jarkko Pakkanen  
Joulukuu 2021

# TIIVISTELMÄ

Lauri Pesonen: Suunnittelurationaliaali (Design Rationale)  
Kandidaatintyö  
Tampereen yliopisto  
Konetekniikan tutkinto-ohjelma  
Joulukuu 2021

---

Suunnittelurationaliaalilla (englanniksi Design Rationale) tarkoitetaan suunnitteluun liittyvien päätösten perustelua eli syitä sille, miksi jokin asia on tehty tai toteutettu juuri niin kuin se on. Suunnittelurationaliaalin käytön ja tutkimuksen tarkoituksena on avustaa ja tukea melkein mihin tahansa asiaan liittyvää päätöksentekoa antamalla esimerkiksi johonkin projektiin liittyvälle henkilölle kattavampi kuva tuotteesta tai prosessista. Yksityiskohtainen kuvaus jostakin tuotteesta vaatii syitä sille, miksi tuotteen jokin ominaisuus on toteutettu juuri sillä tavalla kuin se on, jolloin suunnittelulle vieraan ihmisen ei tarvitse pohtia syitä toteutukselle uudestaan.

Tämä työ on suoritettu kirjallisuuskatsauksena, jonka tavoitteena oli selvittää, mitä suunnittelurationaliaali on, miten sitä hyödynnetään, kuka sitä on tutkinut ja miksi. Työssä pyritään tuomaan esille suunnittelurationaliaalin teoriaan liittyvät pääkohdat, oleellimmat termit ja antaa aiheesta kiinnostuneelle kattava ensikosketus suunnittelurationaliaaliin.

Ensimmäisessä kappaleessa käsitellään suunnittelurationaliaalin teoriaa, johon sisältyy suunnittelurationaliaalitiedon tyypit, tallennus, esitys, haku ja käyttö. Nämä aiheet määrittelevät sen mitä tietoa tallennetaan, millaiseen muotoon, miten sitä tarvittaessa löydetään ja mihin kaikkeen suunnittelurationaliaalia voidaan hyödyntää. Suunnittelurationaliaalin käytössä suunnittelun tukena on paljon mahdollisuuksia ja käyttökohteita, mutta siitä löytyy myös omat haattansa ja haasteensa.

Toisessa kappaleessa esitellään neljä yleisimmin esille tulevaa suunnittelurationaliaalijärjestelmää, jotka ovat IBIS (Information Based Issue System), DSA (Design Space Analysis eli suunnittelu avaruuden tarkastelu), QOC (Questions, Options and Criteria eli Kysymykset, vaihtoehdot ja kriteerit), PHI (Procedural Hierarchy of Issues eli Ongelmien Prosessimainen hierarkia) sekä DRL (Decision Representation Language eli Päätösten esitys kieli). IBIS on näistä vanhin ja melkein kaikki myöhemmät suunnittelurationaliaalijärjestelmät, ja niitä käyttävät tietokoneohjelmat perustuvatkin argumentaatioon perustuvan IBIS-järjestelmän pohjalle muokaten sitä tai lisäten siihen tarpeen mukaan uutta. Suunnittelurationaliaalijärjestelmien ja näihin perustuvien tietokoneohjelmien käytön tarkoituksena on helpottaa suunnittelurationaliaalin tallentamista ja hyödyntämistä luomalla sille yhtenevät kehykset ja rakenteen, jolloin sen ymmärtäminen olisi helpompaa useille tutun

Viimeisessä kappaleessa keskitytään suunnittelurationaliaalin tutkimuksen historiaan eli selvittää, kuka on aloittanut tutkimuksen ja miksi ja miten suunnittelurationaliaalin tutkimus on jakautunut eri tieteen aloille omiksi kokonaisuuksikseen. Suunnittelurationaliaalin tutkimus ei ole niin yhtenevää kuin voisi kuvitella vaan eri tieteenalat tutkivat ja kehittävät sitä omien tarpeidensa mukaisesti. Tieteenalat, jotka ovat erityisen kiinnostuneita suunnittelurationalialista ovat muun muassa tietojärjestelmätieteet, kognitiotieteet sekä tietämyksenhallintatieteet, mutta suunnittelurationaliaalia hyödynnetään jossain muodossa hyvin laaja-alaisesti.

Avainsanat: Suunnittelurationaliaali, Design Rationale, IBIS, QOC, DRL, PHI

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. TEORIA .....	2
2.1 Tyypit .....	2
2.2 Tiedon tallennus.....	3
2.3 Tiedon esitys.....	6
2.4 Tiedon haku .....	7
2.5 Käyttö .....	7
2.6 Haasteet .....	10
3. JÄRJESTELMÄT .....	11
3.1 Information Based Issues System (IBIS).....	11
3.2 Design Space Analysis (DSA).....	13
3.3 Procedural Hierarchy of issues (PHI) .....	14
3.4 Decicion Representation Language (DRL) .....	16
4. AIHEEN TUTKIMUS .....	19
5. YHTEENVETO.....	21
LÄHTEET .....	22

# LYHENTEET JA MERKINNÄT

DRL	Decision Representation Language
DSA	Design Space Analysis
IBIS	Issue-Based Information System
PHI	Procedural Hierarchy of Issues
QOC	Question, Options and Criteria

# 1. JOHDANTO

Päätöksenteko on keskeinen osa kaikkea suunnittelutyötä, sillä monien asioiden ratkaisuun löytyy lähes aina useita eri ratkaisuita, joiden väliltä pitäisi osata valita paras. Usein kuitenkin, kun päätökset ja valinnat on tehty jää näiden syyt dokumentoimatta, jolloin jälkikäteen voi olla vaikeaa ja aikaa vievää yrittää päätellä alkuperäisen suunnittelijan tarkoitusperiä ja ajatusmaailmaa. Tämän työn aiheena on suunnittelurationaali eli englanniksi Design Rationale, jolla tarkoitetaan lyhyesti minkä tahansa asioiden suunnitteluun liittyvien päätösten perusteluja. Näiden perustelujen tallentamiseen, järjestämiseen sekä esittämiseen ovat vuosien aikana kehitetty paljon erilaisia työkaluja, joiden tarkoituksena on suunnittelun ja muun tuotteeseen liittyvän toiminnan tehokkuuden parantaminen.

Työn tutkimusmenetelmänä on kirjallisuuskatsaus aiheeseen ja julkaisuja on etsitty enimmäkseen Andorista ja Google Scholarista. Julkaisuja on etsitty englanninkielisillä termeillä kuten "Design Rationale", "IBIS", "DRL", "QOC" ja "PHI", sillä kaikki löytämäni julkaisut ovat englanninkielisiä.

Suunnittelurationaalia on tutkittu maailmanlaajuisesti jo vuosikymmeniä ja sitä tutkitaan myös joissakin Suomen yliopistoissa. Tieteenaloja, jotka ovat olleet erityisen kiinnostuneita suunnittelurationaalijärjestelmien tutkimuksesta, ovat erinäiset tietojärjestelmätieteet, kognitiotieteet sekä tietämyksenhallintatieteet.

Tämän työn tavoitteena on tutkia ja perehtyä suunnittelurationaalin teoriaan ja tutustua vähän myös sen historiaan. Tässä työssä selvitetään, miksi suunnittelurationaalia on alettu tutkimaan, kuka sitä on tutkinut ja miksi.

Työn ensimmäisessä kappaleessa esitellään aiheittain suunnittelurationaalin teoriaa, johon kuuluvat tiedon suunnittelurationaali tiedon tyytit, tallennus, esitys, haku ja käyttö. Seuraavassa kappaleessa luetellaan yleisimmät ja tunnetuimmat suunnittelurationaali järjestelmät eli IBIS, QOC, PHI ja DRL, joiden pohjalle suurin osa nykyaikaisista suunnittelurationaaliohjelmista perustuvat. Viimeisessä kappaleessa pohditaan aiheen tutkimusta ja syitä miksi.

## 2. TEORIA

Suunnittelurationalilla on melko suuri määrä eri määritelmiä, joita on vuosien mittaan kertynyt paljon, koska kyseistä aihetta on tutkittu todella mittavasti ja useat tutkijat ovat antaneet oman näkemyksensä siitä, mitä se oikein on. Iso osa näistä määritelmistä on pohjaltaan melko lähellä toisiaan: usein samankaltainen näkemys, mutta sanoitettu eri tavalla. Yleisimmin sillä tarkoitetaan jonkin tuotteen tai siihen liittyvän ominaisuuden historiallista dokumentaatiota, joka sisältää syyt ja tiedon sille, miksi jokin tuote tai sen ominaisuus on sellainen kuin on ja toimii kuten toimii (Yakemovic & Conklin 1990). Tämä on yksinkertaisin ja helpoimmin ymmärrettävä määritelmä, joka on hyvä pohja kaikkien muiden määritelmien ymmärtämiselle.

Carrollin ja Rosson (1991) mukaan suunnittelurationalilla voidaan tarkoittaa myös psykologisia väitteitä, joiden tulee olla tosia tuotteen käyttäjän kohdalla, jotta tuote tai ominaisuus toimisivat halutulla tavalla. Päätösten perustelu siis pohjautuu siihen, miten käyttäjän kuvitellaan sen kokevan, ja onko kyseiset väitteet tosia.

Macleenin et al. (1989) mukaan se on tuotteen tai ominaisuuden suhde muihin vaihtoehtoihin, eli miten se sijoittuu niin sanottuun suunnittelutilaan: Mitä ovat vaihtoehdot? Mikä on niiden välinen suhde? Mitä ovat kunkin hyvät ja huonot puolet?

Leen ja Lain (1997) mukaan rationaalini tulisi pelkkien syiden lisäksi sisältää myös suunnittelupäätösten perustelut: muut harkitut vaihtoehdot, ominaisuuksien vaihtokauppa (mistä luovutaan ja mitä saadaan vaihdossa) sekä argumentaatio eli keskustelu, joka loppujen lopuksi johti valintaan. Tämä määritelmä on myös melko helposti ymmärrettävä, mutta huomattavasti yksityiskohtaisempi, ja hyvä lisäys Yakemovicin ja Conklinin (1990) määritelmään.

### 2.1 Tyypit

Rationaali voidaan luokitella seuraavalla tavalla:

Argumentaatiopohjaisessa rationaalissa nimensä mukaisesti keskitytään tallentamaan ja esittämään suunnittelun argumentointia. Tähän siis sisältyvät suunnittelun ongelmat, vaihtoehdot ongelmiin sekä perustelut jokaisen vaihtoehdon puolesta ja vastaan (Garcia et al. 1993; Burge & Brown 1998).

Historiapohjaisessa rationaalissa keskitytään suunnittelun historiaan eli vaiheittain siihen, miten suunnittelu ja siihen liittyvät tapahtumat ovat edenneet. Dokumentaatio voi

olla missä vaan muodossa ja sisältää vaikkapa muistiinpanoja, sähköposteja tai muita dokumentteja, joihin on tallentunut suunnittelun eteneminen (Garcia et al. 1993, Burge & Brown 1998).

Laitepohjaisessarationaalissa hyödynnetään laitteen mallia rationaalien keräämiseen ja sen esittämiseen. Tässä siis on ideana, että selitykset ongelmiin tai kysymyksiin saatetaan simuloimalla varsinaisen laitteen toimintaa ja käytöstä hyödyntäen sen mallia. Suunnittelijan olisi myös mahdollista tarkastella itse mallia ja kysyä sen suunnittelusta ja toiminnasta. (Gruber 1990, Burge & Brown 1998).

Prosessipohjaisessarationaalissa suunnittelurationalin tallennus ja keräys on jollakin tavalla integroitu suoraan suunnitteluprosessiin itseensä, jolloin se myös määrää, mitä ja miten tallennetaan. Suunnitelman kuvaus siis muuttuu vain sen myötä, miten suunnittelun tavoitteita muutetaan ja parannellaan eli rationaali muuttuu prosessiin aikana luonnollisesti (Ganeshan et al. 1994).

Aktiividokumenttipohjaisessarationaalissa suunnittelurationali luodaan tietojärjestelmään valmiiksi. Kun suunnittelija työstää mallia, luo tietojärjestelmä tälle suoraan suunnittelurationalin, jonka jälkeen se vertaa tätä järjestelmän omaan tietoon. Se siis vertailee suunnittelijan päätöksiä niihin mitä tietojärjestelmä olisi itse tehnyt vastaavassa tilanteessa. Jos näissä kahdessa tulee ristiriitoja tietojärjestelmä ehdottaa korjausta suunnittelijan malliin (Burge & Brown 1998).

## **2.2 Tiedon tallennus**

Suunnittelurationalin tallennus on ensimmäinen ja oleellisin vaihe sen elinkaareissa, koska tallentamatonta tietoa ei voida hyödyntää myöhemmin. Tiedon tallennuksen pääpiirteet ovat se mitä tallennetaan ja miten. Suunnittelurationalijärjestelmän kehittämisessä on myös tärkeätä ottaa huomioon, kuinka paljon tietoa tallennetaan rationaalien luomista varten. Jos tallennetaan liian vähän tietoa, ei täydellistä rationaali voida muodostaa. Liian Suuren tietomäärän tallennus voi taas merkittävästi haitata itse suunnittelutyötä. (Regli et al. 2000)

Reglin et al. (2000) mukaan käytännössä mikä tahansa tieto formaatista riippumatta voidaan tallentaa suunnittelurationaliksi tai hyödyntää sen muodostuksessa, kuten esimerkiksi; CAD-mallit, piirustukset ja kokousten videotallenteet. Lähes mikä vaan data, joka kertoo jotain tuotteesta tai sen suunnittelusta.

Myös Burgen ja Brownin (1999) mukaan tämä tiedon kerääminen on hyvin olennainen osa suunnittelurationaali järjestelmää, mutta myös hyvin ongelmallinen. He myös korostavat sitä, että tiedonkeruu ei saisi häiritä varsinaista suunnittelutyötä; Ideaalisen järjestelmän tiedonkeruu ei häiritse itse suunnitteluprosessia.

Burge ja Brown (1999) erittelevät viisi erilaista suunnittelurationaalin tallennusmetodia:

1. Rekonstruointi eli uudelleenrakentaminen on keino, jossa suunnittelurationaali tallennetaan vasta suunnittelutyön jälkeen, jolloin se ei häiritse suunnittelua. Haittapuolena on epätarkkuus, jolloin kaikkea rationaalia ei saada tallennettua oikein (Lee 1997).
2. Metodinen sivutuote (methodological byproduct), jossa rationaalin tuottamista itse suunnittelun aikana edesautetaan käyttämällä suunnittelussa avustavaa metodia, jonka on samalla tarkoitus muodostaa rationaali. (Lee 1997)
3. Harjoittelija (apprentice) tässä suunnittelurationaali järjestelmä seuraa suunnittelijan työtä, ja pyytää selvennystä, jos se ei ymmärrä tämän tekemiä päätöksiä tai toimia. Näissä järjestelmissä täytyy olla valmiiksi tallennettuna jotain rationaalia, jotta se osaa luoda valmiin rationaalin, jota verrata suunnittelijan malliin ja toimiin. Järjestelmän ennustaessa suunnittelijan toimet oikein, se tallentaa valmiiksi generoimansa rationaalin. (Lee 1997).
4. Automaattinen muodostus (automatic generation), missä rationaali muodostetaan suoraan suorite/toimi historiasta suunnitteluprosessin jälkeen (Lee 1997).
5. Historioitsija (historian), missä henkilö tai tietokone ohjelma (järjestelmä) pitää kirjaa kaikista suunnitteluprosessin toimenpiteistä. Tässä tiedon kerääjä ei tee ehdotuksia, ja eroaa siten harjoittelija tavasta. Rationaali myös muodostetaan prosessin aikana, eikä sen jälkeen kuten automaattisessa muodostuksessa (Chen 1990, Burge & Brown 1999).

Seuraavat taulukot (Taulukot 1-3) on tehty Burgen & Brownin (1999) vastaavien taulukoiden pohjalta ja niiden on tarkoitus selvittää eri tiedon tallennusmetodien oleellisimpia ominaisuuksia. Taulukko 1 kertoo tallennustapahtuman ajankohdan suhteessa suunnittelutapahtumaan. Jotkin metodit siis tallentavat suunnittelurationaalia samanaikaisesti suunnittelun kanssa ja toiset vasta sen jälkeen. Taulukko 2 havainnoi metodien tallentaman suunnittelurationaalin tai tiedon laadun eli mitä siihen sisältyy. Taulukko 3 näyttää eri metodien vuorovaikutuksen määrän suunnittelijan tai suunnittelutiimin kanssa eli kuinka paljon metodi vaatii huomioita ja erillistä työtä suunnittelun ohella.



**Taulukko1.** Eri tallennusmetodien ajankohta. (Burge & Brown 1999)

Metodi	Suunnittelun jälkeen	Suunnittelun aikana
rekonstruktio	X	X
Methodinen		X
Harjoittelija		X
Automaattinen	X	
Historioitsija		X

**Taulukko 2.** Eri metodien tallentama tieto. Burge and Brown 1999)

Metodi	Suunnitteluhistoria	Suunnittelurationaali
rekonstruktio	X	?
Methodinen	X	X
Harjoittelija	X	X
Automaattinen	X	
Historioitsija	X	?

Taulukko 3. Metodien vuorovaikutus suunnittelijan työhön. (Burge &amp; Brown 1999)

Metodi	Korkea vuorovaikutus	Matala Vuorovaikutus
rekonstruktio	X	
Methodinen		X
Harjoittelija		X
Automaattinen	X	
Historioitsija	X	

Regli et al. (2000) jakaa tallennustavat karkeasti kahteen eri tyyppiin: manuaaliseen ja automaattiseen. Aikaisemmin mainitut tavatkin voidaan kaikki jakaa näihin kahteen kategoriaan. Manuaalisessa suunnittelija kirjaa ja tallentaa suunnittelurationaaliksi tarkoitettun tiedon järjestelmään itse, kun taas automaattisessa rationaalien ja tiedon tallentamisen hoitaa tietokone. Manuaalisessa tallentaminen tapahtuu yleensä vasta päätöksenteon jälkeen, eikä se siis varsinaisesti ehdi vaikuttamaan päätöksentekoon toisin kuin automaattisessa.

### 2.3 Tiedon esitys

Rationaalien esitystavalla tarkoitetaan, sitä millaisessa muodossa tietoa on tallennettu, mikä voidaan käsittää skaalana muodollisesta epämuodolliseen. Tietokone kykenee ymmärtämään ja hyödyntämään vain muodollisessa (formal) tietomuodossa tallennettua tietoa, mutta ei välttämättä kykene esittämään sitä ihmiselle ymmärrettävästi (Burge & Brown 1999). Eli tämä vaatii, että tietokoneelle syötetty tieto seuraa tiettyä kaavaa tai on tiettyssä muodossa, jotta tietokone osaa lukea sitä.

Vastakohta muodolliselle (formal) on tietenkin epämuodollinen (informal) tietomuoto, joka taas on helposti ihmisen ymmärrettävissä, mutta tietokone taas ei tätä kykene lukemaan, koska se ei seuraa mitään lukukelpoista mallia. (Burge & Brown 1999)

On myös olemassa muotoja, jotka ovat jotain muodollisen ja epämuodollisen väliltä (semiformal), jotka on kehitetty, jotta saataisiin hyödynnettyä molempien tietomuotojen hyviä puolia. Epämuodollisen tietomuodon mainittavana etuna on tallentamisen helppous,

sillä tietoa ei tarvitse muuttaa muodosta toiseen vaan se voidaan pitää halutessa alkuperäisessä formaatissaan, jolloin suuren tietomäärän tallentaminen on vaivattomampaa. Tämä toisaalta tekee tiedon uudelleenhakemisesta ja analyysistä vaikeampaa, koska se ei ole epämuodollisuutensa takia yhtä vertailukelpoista tai tietokoneen luettavissa. Helpottaakseen tiedon tallennusta on myös kehitetty vaiheittainen formalisointi (incremental formalization), jossa tieto tallennetaan ensin epämuodollisesti, jonka jälkeen se muutetaan tarpeen mukaan muodollisempaan muotoon. (Burge & Brown 1999)

Suunnitteluprosessin aikana suunnittelurationaalia voidaan hyödyntää paremmin, jos se on tallennettu muodollisempaan muotoon, koska tietokone pystyy tällöin vaikuttamaan tarvittaessa suoraan päätöksentekoon. (Prabhakar & Goel 1998, Buckingham-Shum & Hammond 1994)

## **2.4 Tiedon haku**

Tiedon haulla tarkoitetaan sitä, miten aikaisemmin tallennettu suunnittelurationaali löydetään järjestelmästä. Rationaalien keräämisestä ei ole mitään apua, jos tätä tietoa ei voida myöhemmin löytää ja käyttää hyödyksi. Tiedonhaku menetelmät voidaan jakaa kahteen eri kategoriaan. Käyttäjälähtöiseen ja järjestelmälähtöiseen (Lee 1997, Pena-Mora et al. 1995).

Käyttäjälähtöisessä järjestelmässä ihminen itse päättää, mihin tietoon haluaa päästä käsiin, kun taas järjestelmälähtöisessä tietokone päättää ihmisen puolesta, mitä tietoa tämän tulisi nähdä. Jälkimmäinen on tehokkaampaa, mutta vaatii järjestelmältä paljon aikaisempaa tietoa, jotta se osaa näyttää tarvitun ja halutun tiedon (Burge & Brown 1999).

Tiedonhaun kannalta on myös oleellista (varsinkin käyttäjäpohjaisessa haussa), miten tieto on organisoitu järjestelmässä. Yleinen tapa on organisoida suunnittelurationaali sen mukaan mihin tuotteen osaan se liittyy. Tämä on yksinkertaisuutensa takia hyvä tapa kokemattomammille tai aiheesta tietämättömämmille suunnittelijoille (Gruber 1990, Brown & Bansal 1991). Toinen vaihtoehto on organisoida tieto prosessien eikä yksittäisten osien mukaan, mikä taas voi olla parempi tapa antamaan kokeneemmille suunnittelijoille kattavampi kuva prosessinäkökulmasta, johon osat sijoittuvat (Brown & Bansal 1991).

## **2.5 Käyttö**

Yleensä jos ajatellaan tuotteen tai ominaisuuden dokumentaatiota ajatellaan, että ainut asia mitä dokumentaatiosta pitäisi löytyä on vain sen spesifikaatiot ja parametrit, ja harvemmin syyt miksi niihin on päädytty (Klein 1993). Tämä saattaa jättää pois oleellista

tietoa, joka auttaa ymmärtämään suunnittelua johtaen ylimääräisen kommunikaation tarpeeseen saman projektin parissa työskentelevillä osapuolilla.

Suunnittelurationaalia voidaan hyödyntää melkein missä tahansa tuotesuunnittelun ja sen elinkaaren vaiheessa. Burge ja Brown (1999) mainitsevat yhteensä seuraavat kahdeksan eri käyttötarkoitusta tai tilannetta, joissa suunnittelurationaalista voidaan saada apua:

1. Suunnittelun verifikaatio (design verification) eli varmistus siitä, että tuote tai sen ominaisuus saavuttaa tavoitteet ja on suunnittelijan alkuperäisen näkemyksen mukainen. Tätä voidaan hyödyntää missä tahansa suunnittelun tai projektin vaiheessa.
2. Suunnittelun arvostelu (design evaluation), mikä on lähellä edellä mainittua verifikaatiota, mutta tarkoituksena on enemmänkin vertailla päätöksiä keskenään.
3. Suunnittelun ylläpidossa (design maintenance) rationaalia hyödynnetään määrittämään mitä päätöksiä tehtiin suunnittelun aikana, millä pyritään paikantamaan tuotteen tai ominaisuuden mahdollisten vikojen lähteitä tai syitä. Sen avulla siis yritetään selvittää, mitä asioita tulee muuttaa tai muokata virheen korjaamiseksi. Myös mahdollisten hylättyjen vaihtoehtojen ylhäällä pitäminen estää suunnittelijaa tekemästä samoja virheellisiä tai vääriä päätöksiä myöhemmin.
4. Suunnittelun uudelleenkäyttö (design reuse) nimensä mukaisesti keskittyy siihen, miten aikaisempaa suunnittelua voidaan hyödyntää uusissa projekteissa, joko suoraan tai miten sen voisi muokkaamalla saada vastaamaan uusia tavoitteita tai tarpeita. Tässä käytössä on ehdottoman tärkeätä, että uudelleenkäytettävästä osasta on tallella kattava suunnittelurationaali, jotta uudelleenkäytön aikana ei tehdä sellaisia muutoksia, jotka ovat elintärkeitä osan oikealle ja turvalliseen toiminnalle.
5. Suunnittelun opetus (design teaching), jossa rationaalia käytetään apuvälineenä mallin, ominaisuuden tai tuotteen perehdyttämisessä sellaisille henkilöille, jotka eivät näitä tunne. Rationaalilla voidaan antaa hyvä peruskäsitys, siitä miten tuote tai sen ominaisuus toimii ja tarjoaa selityksen, miksi jokainen suunnittelupäätös on tehty kuten on. Jotkin suunnittelurationaalijärjestelmät, jopa kykenevät vastaamaan suunnitteluun liittyviin kysymyksiin suoraan sen sijasta, että henkilö joutuisi selaamaan ja etsimään haluamansa tiedon suuresta määrästä dokumentaatiota. Tällaiset suunnittelurationaalijärjestelmän ominaisuudet korostuvat varsinkin silloin, kun tuotteen alkuperäinen suunnittelijan ei ole itse mahdollista opettaa tai kertoa tuotoksestaan.

6. Suunnittelun viestintä (design communication) tarkoittaa projektiin liittyvien suunnittelijoiden välistä viestintää. Tiedon välittyminen oikeana ja suunnittelijan alkuperäisen ajatuksen viestintä suunnittelutiimin sisällä on tärkeää, jotta suunnittelu olisi sujuvaa, eikä virheitä tulisi pelkästään sen takia, että toinen suunnittelija ei ymmärrä toisen suunnittelijan tarkoitusta. Keräämällä rationaalia valintojen ja niiden perustelujen muodossa ja jakamalla tämä tieto muille suunnittelussa mukana oleville, on mahdollista antaa muille hyvä peruskäsitys mallista ja tarjota heille myös paremmat edellytykset vuorovaikuttaa projektiin ja antaa oma panoksensa suunnitteluprosessille (Fischer et al., 1995). Tämä auttaa myös useamman suunnittelijan välisen työn ristiriitojen havaitsemisessa (Pena-Mora et al., 1995). Suunnittelurationaalin avulla voidaan myös vastata suunnittelu arvioijoiden (design reviewer) kysymyksiin, joille on tärkeitä tietää valintojen syyt. Suunnittelurationaalia voidaan käyttää keskustelupohjana suunnittelutiimin sisällä suunnittelun historian tallentamisessa, suunnitelmien muokkauksessa ja ylläpidossa sekä samankaltaisten tuotteiden suunnittelussa (Guindon et al 1987; Goel 1998).
7. Suunnittelun avustuksessa (design assistance) rationaalista voi olla myös hyötyä itse suunnitteluprosessin aikana, sillä mahdollisuus valintojen arviointiin tekee suunnittelun tulosten todentamisen helpommaksi. Argumentaation dokumentoinnin vaikutuksia suunnitteluun ovat esimerkiksi argumenttien selventyminen kannustamalla suunnittelijaa dokumentoimaan tietoa ja varmistuminen siitä, että mahdolliset ongelmat ovat ratkaistut ja se täyttää asetetut vaatimukset.
8. Tuotteen dokumentaatiossa (design documentation) rationaalia voidaan käyttää apuna syiden ymmärtämisessä tarjoamalla kuva suunnitteluprosessin historiasta, joka auttaa antamaan kokonaisemman kuvan lopputuloksesta. Jos tieto on tallennettuna tietokoneuuttavaan (formal) muodolliseen muotoon, on mahdollista luoda tietokoneohjelma, joka voisi tilanteen mukaan muodostaa tarpeeseen sopivampia suunnittelurationaaleja esimerkiksi suunnittelijalle tai asiakkaalle. Eri käyttäjät hyötyisivät enemmän heille räätälöidyistä dokumentaatioista ja suunnittelurationaaleista, sillä eri käyttäjäryhmillä on eri tarpeita ja vaatimuksia tiedon yksityiskohtaisuudesta. Tällaisissa ohjelmissa käyttäjä voisi myös kysyä tietokoneelta suunnitteluun liittyviä kysymyksiä ja tietokone muodostaisi keräämänsä tiedon perusteella vastauksia.

## 2.6 Haasteet

Useiden aikasemmin mainittujen käyttökohteiden ja etujen lisäksi suunnittelurationaalien käyttöön liittyy myös omat haasteensa. Hornerin ja Atwoodin (2007) mukaan yksi oleellisia haasteita on suunnittelijan kognitiiviset rajoitukset, jolla tarkoitetaan ihmisen kapasiteettia ja kykyä tehdä useita eri asioita samanaikaisesti. Heidän mukaansa on mahdollista, että kaikki itse suunnittelutyön ohella tehtävä muu työ vie osan tästä kapasiteetista, joka tulisi kaikki käyttää vain itse suunnitteluun työn tuloksen parantamiseksi. Toinen ihmisen kognitiivisiin kykyihin liittyvä ongelma on Polanyin (1966) mukaan suunnittelijan mahdollinen hiljainen tieto (tacit knowledge) eli tieto jonka osataan tehdä päätöksiä, mutta näitä ei osata selittää tai kuvailla. Tätä voisi kuvailla siis ihmisen alitajunnan tietämykseksi, jota voi tämän takia olla todella vaikeata muuttaa suunnittelurationaaliksi, jolloin oleellista tietoa voi jäädä tallentamatta. Burgen ja Brownin (1999) mukaan ratkaisu suunnittelijan rajallisen kapasiteettiin keskittämiseen suunnittelutyöhön olisi kehittää sellainen suunnittelurationaalijärjestelmä, joka osaisi tallentaa tietoa itsenäisesti taustalla, jolloin suunnittelijan ei tarvitsisi keskittyä siihen itse laisinkaan.

Grudinin (1994) mukaan yksi iso haaste, joka rajoittaa suunnittelurationaalien hyödyntämistä on sen tallentamisesta saatu hyöty ja kenelle se kohdistuu.. Itse suunnittelija ei välttämättä saa ylimääräisen tiedon tallennuksesta mitään konkreettista hyötyä, mikä taas ei kannusta häntä tätä tekemään. Suunnittelurationaalista saatu hyöty pitäisi siis myös kohdistua suunnittelijalle eikä pelkästään niille, jotka ovat tekemisissä tuotteen kanssa suunnitteluprosessin jälkeen. Eli monissa tapauksissa paras tapa saada suunnittelija tallentamaan rationaalinsa on pyrkiä antamaan tälle välitöntä hyötyä sen tekemisestä.

Yksi suurimmista haasteista on kuitenkin Reglin et al. (2000) mukaan se, että mitään kaikille erillisille organisaatioille toimivaa yhtenäistä tai yleispätevää suunnittelurationaalijärjestelmää on lähes mahdoton kehittää, koska kaikilla eri tieteen-aloilla ja yrityksillä on omat tarpeensa, jotka vaativat lähes aina räätälöidyn järjestelmän jotta sen käyttö olisi tehokasta, jolloin sen käytöstä olisi enemmän hyötyä kuin haittaa.

### 3. JÄRJESTELMÄT

Suunnittelurationaalijärjestelmällä tarkoitetaan yleisesti mallia, jonka avulla pyritään muodostamaan objektille järjestelmällinen suunnittelurationaali, joka seuraa tiettyä kaavaa.

Suunnittelurationaalijärjestelmä antaa suunnittelijoille mahdollisuuden ajatella ja kommunikoida suunnitteluansa tietynlaisen ja standardisoidun kehyksen mukaisesti (Regli et al. 2000). Eli tarkoituksena on antaa suunnittelijalle työkalu, joka helpottaa hänen omaa ajatustyötänsä varmistamalla, että objektin rationaali tallentuu sellaisessa muodossa, jota suunnittelijan ja muiden on helppo ymmärtää. Esimerkiksi jos jokaisella projektissa mukana olevalla ihmisellä olisi eriävä tapa tallentaa ja esittää suunnittelurationaalinsa tarkoittaisi se sitä, että jokaisen henkilön pitäisi osata ja ymmärtää useita eri järjestelmiä.

Itsenäinen suunnittelurationaalijärjestelmä ei ole kovinkaan käytännöllinen ja tehokas tapa tuottaa rationaalia. Integroituna muihin suunnittelua tukeviin ohjelmiin kuten CAD (Computer Aided Design) tai CASE (Computer-Aided Software Engineering) sitä voidaan hyödyntää: tiedon esityksen kehyksenä, suunnittelurationaalien tallennukseen, suunnittelun järjestyksen ja kommunikointiin prosessin aikana. Tällainen ohjelmien integraatio voi merkittävästi parantaa tiedon ja rationaalien keruuta prosessin aikana, ja ideaalissa tilanteessa kykenisi jopa muuttamaan raa'an suunnittelurationaalien ja suunnitteluhistorian käyttökelpoiseksi tiedoksi. (Regli et al. 2000)

Kuvassa 1. on hyvin yksinkertaistettu malli, joka kuvaa suunnittelurationaalijärjestelmän käytön eri vaiheita.

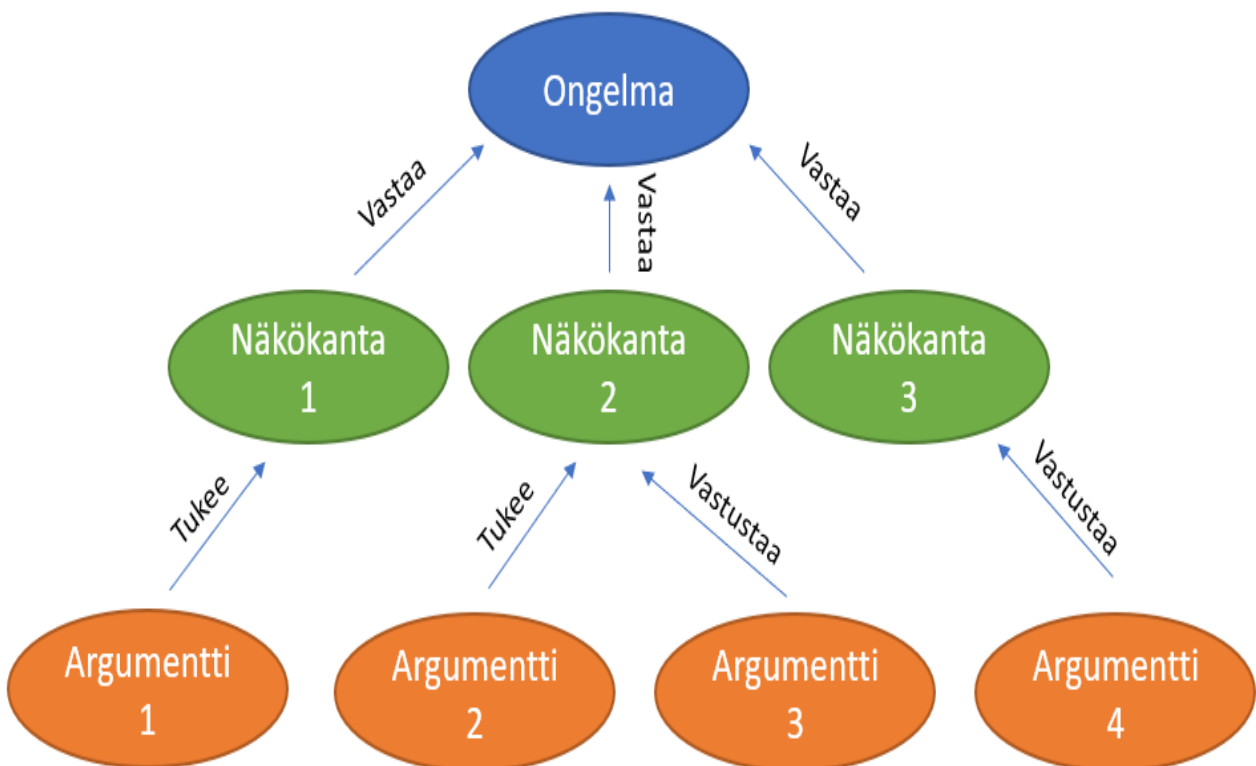
#### 3.1 Information Based Issues System (IBIS)

IBIS (Information Based issue system) eli niin sanottu ongelmalähtöinen tietojärjestelmä on argumentaation perustuva järjestelmä, jonka tarkoituksena on selvittää moniosaisia ongelmia, joissa on useampia alaongelmia ja ratkaisuja. (Kunz & Rittel 1970). Kunz ja Rittel kehittivät tämän järjestelmän 1960-luvulla apuvälineeksi poliittiseen ongelmanratkaisuun, mutta tätä on myöhemmin alettu hyödyntämään hyvin monilla eri aloilla yleisenä suunnittelutyökaluna, ja onkin pohjana isolle osalle tämän päivän suunnittelurationaali

järjestelmistä ja ohjelmista. Kunzin ja Rittelin mukaan IBIS-järjestelmän on tarkoitus ohjata ja auttaa ongelmien tunnistamisessa, muotoilussa sekä niiden ratkaisussa, ja tarjota olennaista tietoa näiden ratkaisuun liittyvästä keskustelusta.

IBIS-järjestelmä peruskomponentit graafisessa esityksessä ovat: Ongelmat, ratkaisut, Argumentit sekä näiden riippuvuuksia merkkaavat nuolet. Ongelmat esitetään useasti kysymyksen muodossa, joihin ratkaisut pyrkivät vastaamaan. Ratkaisuja taas tukee tai vastustaa argumentit, jotka voivat tilanteesta riippuen tukea tai vastustaa useampaa ratkaisua. (Regli et al. 2000)

Kuvassa 2 hyvin yksinkertainen graafinen esimerkki siitä miten IBIS-järjestelmän eri peruskomponentit voivat olla riippuvaisia toisistaan.



**Kuva 2.** Ongelmien, ratkaisujen ja argumenttien väliset suhteet IBIS-järjestelmässä. (Mukaillen Regli et al. 2000)

Kuvassa on siis ratkaistavana yksi ongelma, jolle on löydetty kolme mahdollista ratkaisua. Näiden ratkaisujen väliltä pitäisi päättää argumenttien avulla, jotka tukevat tai vastustavat kutakin ratkaisua. Ajankohtainen esimerkki voisi olla, vaikka henkilöauton alle



laitettavat renkaat, jossa ratkaisuna voisivat olla nasta-, kitka ja kesärenkaat. Näitä ratkaisuja tukevin argumentteina voisi olla vaikkapa vuodenaika, se minkälaisessa ilmastossa eletään, teiden kunnossapidon laatu ja niin edelleen.

IBIS-järjestelmän suhteellinen yksinkertaisuus ja sen keskittyminen kysymyksiin tekee siitä hyvin käyttökelpoisen työkalun varsinkin ongelmanratkaisun alkuvaiheessa eli, kun itse ongelma saattaa vielä olla hyvinkin epämääräisesti määritelty. (Culmsee & Awati 2013)

### 3.2 Design Space Analysis (DSA)

Design Space Analysis (karkeasti suomennettuna suunnitteluavaruuden tarkastelu) on vaihtoehtoinen järjestelmä argumenttipohjaisille suunnittelurationaalijärjestelmille (Maclean et al. 1989). Tässä pyritään sijoittamaan vaihtoehto laajaan suunnitteluavaruuteen, jonka tarkoituksena on selittää, miksi kyseinen vaihtoehto on valittu kaikkien muiden joukosta selventämällä näiden välisiä suhteita. (Maclean et al. 1991)

DSA:n pohjana ja mallina on QOC (Questions, Options and Criteria), joka on IBIS-järjestelmän kaltainen esitysmalli, sillä molemmat rakentuvat kolmesta peruskomponentista ja niiden välisiä suhteita kuvaavista nuolista. QOC:ssa nämä komponentit ovat kysymykset (questions) eli suunnitteluun liittyvät ongelmat, vaihtoehdot (options) eli mahdolliset ratkaisut ongelmiin ja kriteerit (criteria) eli vaatimukset, joiden avulla arvioidaan, miten ratkaisut täyttävät mahdolliset vaatimukset ja ominaisuudet (Maclean et al. 1996). QOC on tallentamis- ja esitystavaltaan epämuodollisen ja muodollisen väliltä (Semiformal) (Maclean et al. 1991). QOC:n esitystapa keskittyy systemaattisesti rakentamaan suunnittelutilan vaihtoehtoista, toisin kuin IBIS ja PHI, joiden tarkoituksena on enemmänkin vain tallentaa suunnitteluun liittyvän keskustelun historia. (Regli et al. 2000)

Regli et al. (2000) mukaan QOC:ta käyttäen on helppoa takaisinmallintaa eli selvittää järjestelmän osien toimintaa jälkikäteen.

Kuvassa 2 on esimerkki suunnitteluun liittyvästä ongelmasta, jota on lähdetty selventämään ja ratkaisemaan DSA:n graafisen esityksen avulla, jossa on käytetty QOC:een kysymys, vaihtoehto ja kriteeri mallia pohjana. Esimerkin ongelmana on tietokoneen vierityspalkin esitys, johon on annettu kaksi vaihtoehtoa, joita taas arvioidaan niihin liittyvien kriteerien avulla. Tässä myös havainnoidaan, sitä miten suunnittelussa voi syntyä lisäkysymyksiä esimerkiksi vaihtoehtojen pohjalta. Kuvassa viivojen päällä olevat plus ja miinus merkit kertovat, onko vaihtoehdolla positiivinen vai negatiivinen vaikutus kriteerin täyttymiseen.



**Kuva 2.** Esimerkki DSA:ta, joka käyttää QOC:ta. (Mukaillen Regli et al. 2000)

QOC:n ja IBI:ksen graafiset esitykset ovat hyvin samankaltaisia kuten kuvista 1. ja 2. voidaan todeta.

### 3.3 Procedural Hierarchy of issues (PHI)

PHI (karkeasti suomennettuna ongelmien prosessimainen hierarkia) on alkujaan myös IBIS-järjestelmään perustuva malli, jossa on uudelleenmääritelty ongelman kuvaus ja muutettu ongelmien, ratkaisujen ja argumenttien välisiä suhderakenteiden sääntöjä. Ongelmien ratkaisemiseen tarjotaan tässä kahta eri metodia: pohdinta tai hajottaminen. Ongelma ratkaistaan siis joko suoraan tai se hajotetaan useiksi eri alaongelmiksi. Huomat-

tavana erona on siis myös tämä alaongelmien huomioon ottaminen, missä jonkin ongelman ratkaiseminen voi vaatia siihen liittyvien alaongelmien huomioon ottamista ja ratkaisua. Pääongelmaksi lasketaan koko suunnitteluprojekti, joka siis hajotetaan suureksi määräksi alaongelmia, jotka sitten ratkaistaan yksitellen, minkä pitäisi loppujen lopuksi johtaa itse pääongelman ratkaisemiseen. (McCall 1991)

Kuvassa 3 on yksinkertainen esimerkki ongelman hajottamisesta alaongelmiksi, jotka voidaan ratkoa erikseen, mikä yksinkertaistaa ja selkeyttää ongelmanratkaisua. Jokaiselle alaongelmalle on vaatimuksista riippuen tarjottu useita eri ratkaisuja ja argumentteja, jotka näitä tukevat.

What kind of conveyers should be used in material handling?

SUBISSUE:

1. What kind of conveyers should be used in bulk material handling?  
...
2. What kind of conveyers should be used in unit material handling?

ANSWERS:

1. Gravity conveyers

SUBANSWERS:

1. chutes, skate wheel conveyers
2. roller conveyers

ARGUMENTS:

1. Its advantage is low cost, relatively low maintenance, and negligible breakdown rate.
2. Its requirement is the ability to provide the necessary gradient in the system configuration.

X...

2. Powered conveyers

...

3. Chain-driven conveyor

...

4. Power-and-free conveyers

...

**Kuva 3.** Esimerkki PHI:ssa ongelman hajottamisesta alaongelmiksi. (Regli et al. 2000)

PHI pyrkii välttämään itsestäänselvien ja irrelevanttien ongelmien puintia (McCall 1991). Verrattuna IBIS-järjestelmään, PHI tarjoaa myös riippuvuussuhteita ratkaisujen välille ja

ottaa huomioon vaihtoehtojen hyvät ja huonot puolet (Fischer et al. 1991). Fischer et al. 1989 mukaan PHI myös mallintaa suunnitteluprosessin rakenteen tarkemmin ja täydellemmin.

### **3.4 Decicion Representation Language (DRL)**

DRL (karkeasti suomennettuna päätösten esityskieli) on nimensä mukaisesti kieli, jonka tarkoituksena on tarjota yhtenevä sanasto päätösten kuvaamiseen ja syihin näiden takana. DRL:ssä siis keskitytään asioiden välisten suhteiden kuvaamiseen määrittelemällä erilaisia vaihtoehtoja kyseisen suhteen laadulle. DRL:än peruskomponentit ovat tavoitteet (goals), vaihtoehdot (alternatives) ja väitteet (claims). Vaihtoehdot ovat mahdollisia ratkaisuja ongelmaan, tavoitteet määrittelevät ideaalin ratkaisun ominaisuudet ja väitteet toimivat perustana argumenteille, joiden avulla lopullinen päätös tehdään. Pääkomponenttien lisäksi DRL sisältää myös muita osia kuten pääongelma (decicion problem), suhteet (DRL relation), kysymys (question), toimenpide (procedure), ryhmä (group) ja näkökanta (viewpoint). Pääongelma tarkoittaa itse suunnitteluongelmaa, jota yritetään ratkaista. Suhteilla tarkoitetaan kaikkien komponenttien välisiä ja erilaisia suhteita toisiinsa. Kysymyksellä nostetaan esille mahdollinen epävarmuus johonkin liittyen, johon vastaamalla pyritään selkeyttämään tai selittämään tarvetta, pyytämällä lisää tietoa. Toimenpide esittää konkreettisesti joko jotain toimenpidettä tai sen kuvausta. Ryhmän avulla voidaan korostaa eri komponenttien tiiviimpää suhdetta tai merkitystä toisiinsa. Näkökanta ryhmittää komponentteja, jotka ovat samojen oletuksien alaisia. (Lee 1989)

Kuvassa 4 on esitetty DRL:än sanastoa eli eri komponenttien mahdollisia suhteita toisiinsa. Ensin on kerrottu suhteen nimi ja kuvaus ja sulkeissa se minkä kahden komponentin välistä suhdetta tämä voi kuvata.

**Alternative**

**Goal**

**Decision Problem**

**Claim**

**DRL Relation**

**Is-A-Sub-Decision-Of (Decision Problem, Decision Problem)**

**Is-A-Goal-For (Goal, Decision Problem)**

**Is-A-Subgoal-Of (Goal, Goal)**

**Is-An-Alternative-For (Alternative, Decision Problem)**

**Is-A-Sub-Alternative-Of (Alternative, Alternative)**

**Facilitates (Alternative, Goal)**

**Supports (Claim, Claim)**

**Denies (Claim, Claim)**

**Qualifies (Claim, Claim)**

**Queries (Question, Claim)**

**Influences (Question, Claim)**

**Are-Arguments-For (Group of Claims, Claim)**

**Is-An-Answering-Procedure-For (Procedure, Question)**

**Is-A-Result-Of (Claim, Procedure)**

**Answers (Claim, Question)**

**Are-Possible-Answers-To (Group of Claims, Question)**

**Is-A-Sub-Procedure-Of (Procedure, Procedure)**

**Is-A-Component-Procedure-Of (Procedure, Procedure)**

**Question**

**Procedure**

**Procedure Description**

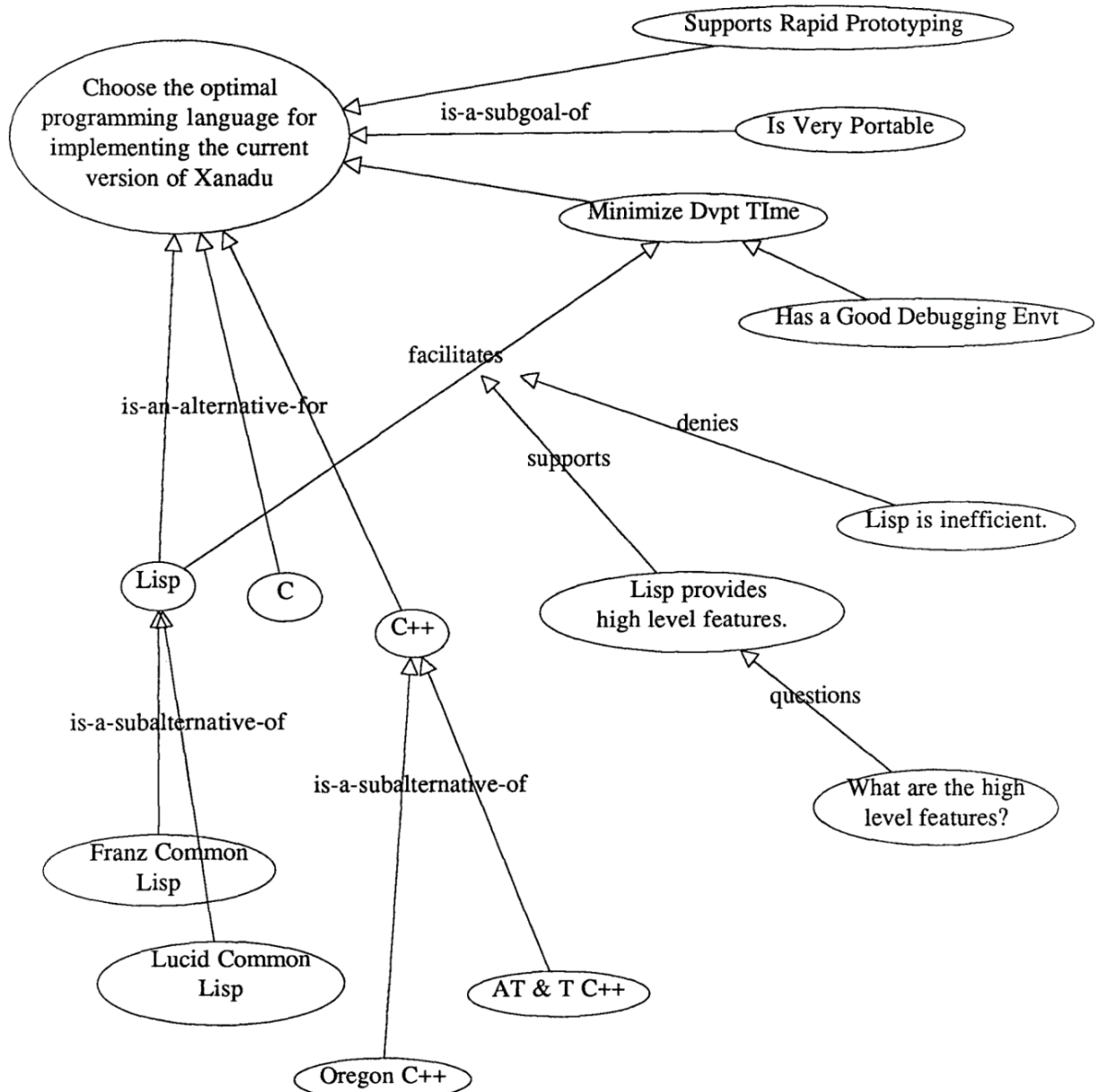
**Executable Procedure**

**Group**

**Viewpoint**

**Kuva 4.** DRL sanastoa, joka kuvailee osien mahdollisia suhteita toisiinsa. (Lee 1989)

Kuvassa 5 on esimerkki DRL:llä käytävästä graafisesta esityksestä, jossa pääongelmana on parhaan mahdollisen ohjelmointikielen valinta erästä ohjelmaa varten.

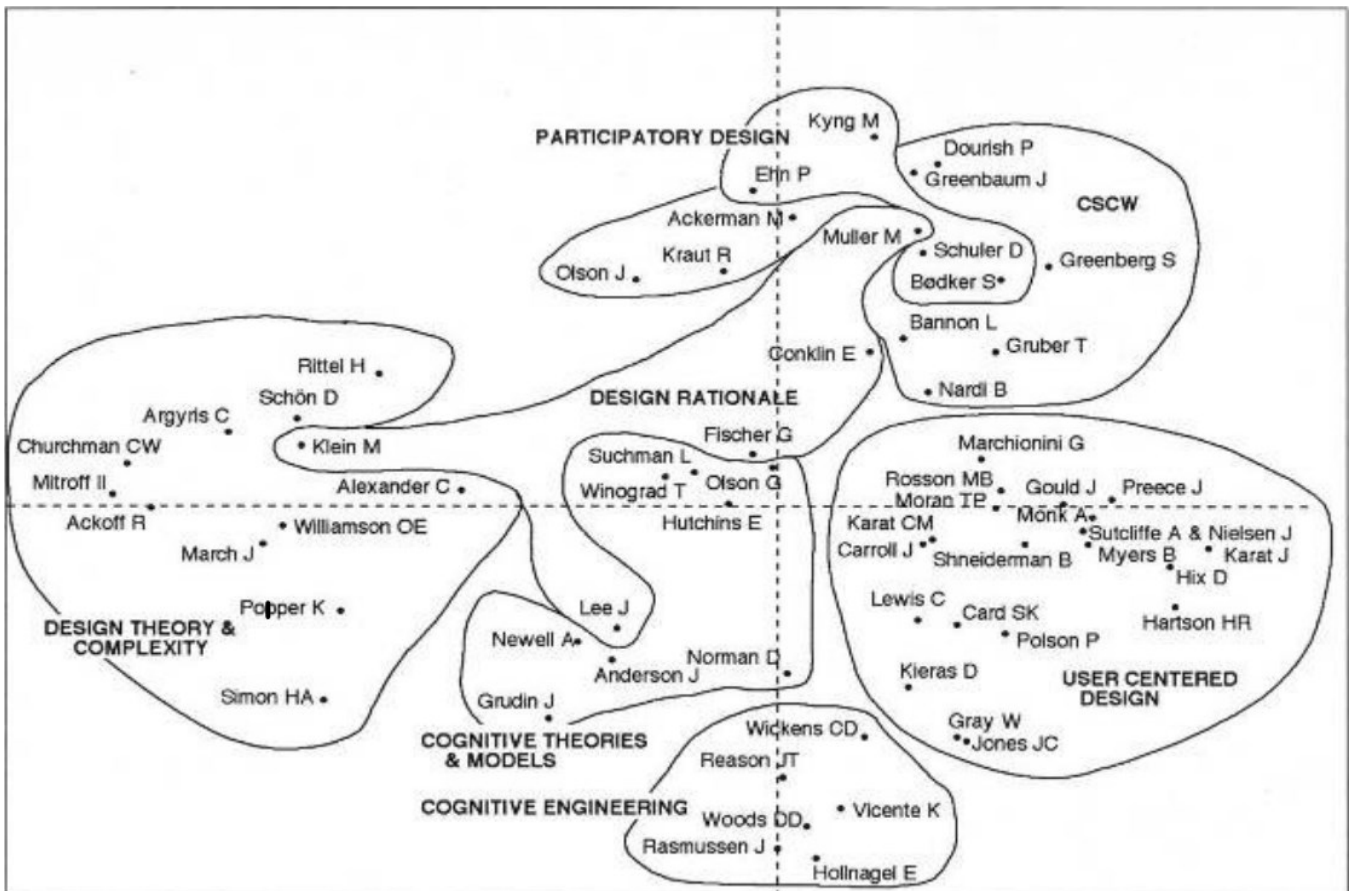


**Kuva 5.** Esimerkki DRL-järjestelmän käytöstä graafisessa esityksessä. (Lee 1989)

DRL:ää voidaan hyödyntää erikseen tai se voidaan integroida perinteisempään suunnittelujärjestelmään. DRL:ää käyttää esimerkiksi SIBYL niminen järjestelmä, jossa sitä käytetään erilaisten tietokone palveluiden tutkimiseen. (Regli et al. 2000)

## 4. AIHEEN TUTKIMUS

Voidaan todeta, että aihetta on jo vuosikymmeniä tutkittu paljon maailmanlaajuisesti ja useiden eri tieteenalojen toimesta, koska näillä on ollut tarvetta kehittää järjestelmiä suunnittelun avustamiseen. Kunz ja Rittel (1970) alkoivat tutkimaan aihetta pyrkimyksenä saada luotua jokin tehokas työkalu auttamaan poliittista päätöksentekoa, mutta sittemmin sen käytön antamia hyötyjä on alettu tutkimaan melkein kaikkeen muuhunkin suunnitteluun ja päätöksentekoon. Tästä todisteena melko kattava määrä jokseenkin samanlaisia, mutta kuitenkin vähän erilaisia järjestelmiä ja niihin pohjautuvia tietokoneohjelmia, joidenka kaikkien tavoitteena on ollut tehostaa suunnittelurationaalien tallentamista ja hyödyntämistä, joista tässä työssä esitetyt ovat vain jäävuoren huippu. Syy näiden suurelle määrälle on eri alojen eri tarpeet ja vaatimukset näille järjestelmille ja ohjelmille, jolloin kunkin alan ja jopa yrityksen on täytynyt kehittää omiin käyttötarkoituksiinsa ja tarpeisiinsa hyvin erikoistuneet ja yksilölliset järjestelmät ja ohjelmat. Tämän takia myös siihen liittyvä tutkimus ei ole ollut niin yhtenevää kuin voisi kuvitella, mikä on nähtävissä Kuvassa 3.



**Kuva 3.** Suunnittelurationaalien tutkijoiden sijoittuminen lainauksien persuteella eri tieteenaloihin. (Wania et al. 2006)

Atwood ja Horner (2007) huomauttavatkin, että suunnittelurationalin tutkimus yhteisöllä ei oikeastaan ole yhteisiä tavoitteita tai kiinnostuksen kohteita. Täten useat suunnittelurationalin tutkijat sijoittuvatkin eri tieteenalojen piiriin ja myös heidän näkemyksensä ja määritelmänsä siitä mitä suunnittelurationali oikeastaan on ja mihin sen tutkimuksen tulisi keskittyä voi olla hyvinkin erilainen. Suunnittelurationalin tutkimus ei siis monesti ole täysin itsenäistä vaan se sivuaa jotakin muuta tutkimusalaa. Reglin et al. (2000) summaa tämän asian toteamalla, että suunnittelurationalia ei kannattaisikaan pitää omana erillisenä tutkimusalueenaan vaan ala-aiheena tai alaongelmana tietämyksenhallintatieteelle.

Tämä ilmenee ehkä kaikkein parhaiten siitä, kuinka paljon eri määritelmiä, järjestelmiä ja ohjelmia suunnittelurationalin tallentamiseen ja hyödyntämiseen on kehitetty. Alkuun voidaan katsoa, että suunnittelurationalin tutkimus on saanut alkusysäyksensä Kunzin & Rittelin toimesta, mutta sittemmin heidän luomansa IBIS-järjestelmän pohjalta on luotu valtava määrä muita järjestelmiä. Eri alojen eri tarpeet ovat luoneet tarpeen määritellä suunnittelurationali hieman eri tavalla, mikä taas määrittelee järjestelmien ja ohjelmien tavat tallentaa ja esittää tietoa näiden tarpeiden mukaan. Eli siis tapa, jolla yksittäisellä tieteenalalla tai organisaatiossa toimitaan tiedon tallennuksen ja esityksen kannalta ei monesti ole toimiva tapa muissa.

Suunnittelurationalin tutkimukseen liittyviä papereita on ajan kuluessa julkaistu huomattava määrä, missä tutkitaan eri järjestelmien (IBIS, PHI, QOC ja DRL) niihin perustuvien tietokoneohjelmien hyödyntämistä suunnittelun tukena esittäen näille melko kattavan määrän eri hyötyjä. Mutta kuten Atwood ja Horner (2007) huomauttavat, löytyy näistä silti yllättävän vähän konkreettisia esimerkkejä näiden mahdollistamista onnistumisista. Toisaalta esimerkiksi Poorkianyn et al. (2016) tuoreemmassa tutkimuksessa, jossa tutkittiin suunnittelurationalin käyttöä eräässä yrityksessä, saatiin sen käyttöä tukevia positiivisia tuloksia. Hyödyiksi listattiin tässä tapauksessa ajan ja rahan säästöt varsinkin, kun yrityksen jostain tuotteesta suunniteltiin eri variantteja. Teknologian ja tietokoneiden kehitys on siis saattanut muuttaa tilannetta huomattavasti vain vajaassa kymmenessä vuodessa.

Suunnittelurationalijärjestelmien perimmäisenä tarkoituksena on tukea suunnittelutyötä tarjoamalla työkalu tallentamaan ja kommunikoimaan argumentointi ja syyt suunnitteluprosessin takana.



## 5. YHTEENVETO

Suunnittelurationalilla pohjimmillaan tarkoitetaan päätöksenteossa tai suunnittelussa niitä syitä, miksi johonkin päätökseen on päädytty tai miksi jokin asia on toteutettu juuri niin kuin se on. Aiheen tutkimuksen katsotaan saaneen alkunsa, kun Kunz ja Rittel (1970) kehittivät poliittisen päätöksen teon tueksi IBIS-järjestelmän, joka on sittemmin synnyttänyt valtavan määrän muita suunnittelurationalijärjestelmiä, joiden pohjalle on taas luotu suuri määrä erilaisia ja eri tarpeita palvelevia suunnittelurationalia tallentavia ja esittäviä tietokoneohjelmia. Näiden moninaisten järjestelmien tavoitteena on ollut tukea melkein mitä vain päätöksentekoa, suunnittelua ja dokumentaatiota, suuressa määrässä eri tieteen aloja ja käyttökohteita. Mahdollisia hyötyjä suunnittelurationalille löytyy melko paljon, mutta sen tehokkaaseen tallennukseen ja käyttöön liittyy myös huomattavia haasteita. Suunnittelurationalin tutkimus on melko hajaantunutta eri tieteenaloille niiden eri vaatimusten takia, mutta voidaan ajatella, että se on osa tietämyksenhallintatiedettä.

Työn tarkoituksena ja päämääränä oli luoda suhteellisen laaja katsaus suunnittelurationalin luonteeseen, sen termistöön ja tunnetuimpiin suunnittelurationalijärjestelmiin, ja antaa aiheesta kiinnostuneelle kevyt ensikosketus suunnittelurationalin ymmärtämiseen. Työ onnistui kokonaisuutena suhteellisen hyvin, mutta sen hyvinkin laajan rajauksen takia jokaisesta asiasta mahtui melko rajallisesti asiaa. Jos tekisin työn uudelleen, olisin varmaankin rajannut aluetta hyvinkin paljon enemmän esimerkiksi vain yhteen suunnittelurationalijärjestelmään ja käsitellyt vain sitä ja tietenkin suunnittelurationalin perusteoriaa.

Työ suoritettiin kirjallisuuskatsauksena aiheeseen, missä pääasiallisesti haettiin tietoa Andor ja Google Scholar -tietojärjestelmistä. Nämä molemmat tietojärjestelmät ovat keskittyneet tieteellisiin julkaisuihin, ja näistä löytyi paljon suunnittelurationalia käsitteleviä tutkimuksia ja kirjallisuutta, joten tiedon löytäminen aiheesta oli siis suhteellisen helppoa. Kaikki löytämäni ja tässä työssä käytetyt lähteet olivat englanninkielisiä, koska mitään työhön sopivia suomenkielisiä ei sattunut löytymään. Tästä syystä olen myös pitänyt englanninkieliset termit mukana, jotta aiheesta kiinnostuneen on helpompi etsiä tietoa näin halutessaan. Jatkoa ajatellen olen itse oppinut aiheesta todella paljon ja uskon, että tästä tulee olemaan hyötyä tulevaisuudessa riippumatta siitä, mitä työtä päädyn tekemään. Ajatusten vielä kypsyyssä en kuvittele sitä täydeksi mahdottomuudeksi, että oma diplomityö voisi jotenkin liittyä aiheeseen, sillä pidän tätä suhteellisen mielenkiintoisena.

## LÄHTEET

- Buckingham-Shum, S. J. & Hammond, N. (1994). Argumentation-based design rationale: What use at what cost? *Human-Comput. Stud.*, 40(4), pp. 603–652.
- Burge, J. & Brown, D.C. (1998). *Design Rationale: Types and Tools*, Technical Report, Worcester Polytechnic Institute, Computer Science Dept., Viitattu 6.11.2021, saatavilla: <http://web.cs.wpi.edu/Research/aidg/DR-Rpt98.html>
- Burge J. & Brown D.C. (2000). Reasoning with Design Rationale. In: Gero J.S. (eds) *Artificial Intelligence in Design '00*. Springer, Dordrecht, pp. 611–629.
- Carroll, J. M. & Rosson, M. B. (1991). Deliberated Evolution: Stalking the View Matcher in Design Space. *Human-computer interaction*. [Online] 6 (3-4), pp. 281–318.
- Chen, A., McGinnis, B., Ullman, D. & Dietterich, T. (1990). Design History Knowledge Representation and Its Basic Computer Implementation, The 2<sup>nd</sup> International Conference on Design Theory and Methodology, ASME, Chicago, IL, pp. 175–185.
- Conklin, J. & Burgess-Yakamovic, K. (1995). A Process-Oriented Approach to Design Rationale, T. Moran and J. Carroll, in *Design Rationale Concepts, Techniques, and Use*, (eds), Lawrence Erlbaum Associates, Mahwah, NJ, pp. 293–428.
- Culmsee, P. & Awati, K. (2013). "Chapter 7: Visualising reasoning, and Chapter 8: Argumentation-based rationale. The heretic's guide to best practices: the reality of managing complex problems in organisations. Bloomington, IN: iUniverse, Inc. pp. 153–211.
- Fischer, G. & McCall, R. (1989). JANUS: Integrating hypertext with a knowledge-based design environment. In: Halasz, F., Meyrowitz, N. (eds.), *Hypertext*, Addison-Wesley, pp. 105–117.
- Fischer, G., Lemke, A. C. & McCall, R. (1991). Making argumentation serve design. *Human-Comput. Interaction*, 6(3–4), pp. 393–419.
- Fischer, G., Lemke, A., McCall, R. & Morch, A. (1995). Making Argumentation Serve Design, in *Design Rationale Concepts, Techniques, and Use*, T. Moran and J. Carroll, eds., Lawrence Erlbaum Associates, pp. 267–294.
- Ganeshan R., Garrett J. & Finger, S. (1994). A framework for representing design intent, *Design Studies Journal*, 15(1), pp. 59–84.

- Garcia, A., Howard, H. & Stefik, M. (1993). Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design, Tech. Report 82, Stanford Univ. Center for Integrated Facility Engineering, Stanford, Calif.
- Gruber, T. (1990). Model-based Explanation of Design Rationale, in Proceedings of the AAAI-90 Explanation Workshop, Boston, July 30, 1990.
- Grudin, J. (1994). Groupware and social dynamics: eight challenges for developers. *Communications of the ACM* 37(1), pp. 92–105.
- Guindon, R., Krasner, H. & Curtis, B. (1987) Breakdowns and processes during the early activities of software design by professionals. In: Olson, G. M. (ed.) *Empirical Studies of Programmers: Second Workshop*, Ablex, pp. 65–82.
- Klein, M. (1993) Capturing design rationale in concurrent engineering teams. *IEEE Computer*, 26(1), pp. 39–47.
- Kunz, W. & Rittel, H. (1970), Issues as elements of information systems. Working Paper 131, Center for Urban and Regional Development, University of California Berkeley.
- Lee, J. & Lai, K. (1991). What's in Design Rationale?. *Human–Computer Interaction*, 6 (3-4), pp. 251–280.
- Lee, J. (1997) Design rationale systems: understanding the issues. *IEEE Expert*, 12(3), pp. 78–85.
- Maclean, A., Young, R. M. & Moran, T. P. (1989). Design Rationale: The argument behind the artifact. *Proceedings of the CHI '89 conference on human factors in computing systems*, pp. 247–252.
- Maclean, A., Young, R.M. & Moran, T. (1989), "Design rationale: the argument behind the artifact", *SIGCHI Bull.* 20, pp. 247-252 114–125.
- MacLean, A., Young, R., Bellotti, V. & Moran, T. (1991) Questions, options, and criteria: Elements of design space analysis. *Human-Comput. Interaction*, 6(3–4), pp. 201–250.
- Maclean, A.; Young, R.M.; Bellotti, V.M.E. & Moran, T. (1996). Questions, Options, and Criteria: Elements of Design Space Analysis, in Moran, T. & Carroll, J., *Design Rationale Concepts, Techniques, and Use*, Lawrence Erlbaum Associates, pp. 53–106.
- McCall, R. J. (1991) PHI: A conceptual foundation for design hypermedia. *Design Studies*, 12(1), pp. 30–41.
- Pena-Mora, F., Sriram, D. & Logcher, R. (1995). Design Rationale for Computer-Supported Conflict Mitigation, *ASCE Journal of Computing in Civil Engineering*, pp. 57–72.

Poorkiany, M., Johansson, J. & Elgh, F. (2016). Capturing, structuring and accessing design rationale in integrated product design and manufacturing processes. *Advanced Engineering Informatics*, 30 (3), pp. 522–536.

Polanyi, M. (1996). *The tacit dimension*. Doubleday, Garden City, NY.

Prabhakar, S. & Goel, A. K. (1998). Functional modeling for enabling adaptive design of devices for new environments. *Artif. Intell. in Eng.*, 12(4), s. 417–444.

Regli, W., Hu, X., Atwood, M. & Sun, W. (2000). A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. *Engineering with Computers; An International Journal of Simulation-Based Engineering*, 16 (3), pp. 209–235.

Wania, C., McCain, K. & Atwood, M.E.: How do design and evaluation interrelate in HCI research? In: *Proceedings of the 6th ACM conference on Designing Interactive systems*, June 26-28, 2006, University Park, PA, USA (2006).

Yakemovic, KC B., & Conklin, J. (1990). Report on a development project use of an issue-based information system. *Proceedings of the conference on computer supported cooperative work*. pp. 105–118.