

# Järjestelmäsuunnittelun kehitys

Historia, nykypäivä ja tulevaisuus

Konetekniikan kandidaatintutkielma

Laatija(t):  
Väinö Ruikkala

14.5.2024  
Turku

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu  
Turnitin OriginalityCheck -järjestelmällä.

Kandidatutkielma

**Oppiaine:** Konetekniikka  
**Tekijä(t):** Väinö Ruikkala  
**Otsikko:** Järjestelmäsuunnittelun kehityksen motivaatiot  
**Ohjaaja(t):** Jani Heikkinen  
**Sivumäärä:** 22 sivua  
**Päivämäärä:** 15.5.2024

Tiivistelmän tyyli on **Abstract**.

**Avainsanat:** järjestelmäsuunnittelu, kirjallisuuskatsaus

# Sisällysluettelo

<b>1 Pohjatietoa ja määritelmiä.....</b>	<b>4</b>
1.1 Järjestelmän määritelmä.....	5
1.2 ”Malli” terminä ja käsitteenä.....	6
1.3 Mallisuunnittelun määritelmä.....	6
1.4 Mallisuunnittelu vastaan dokumenttisuunnittelu (MBSE vs. DBSE).....	7
1.5 Mallinnus ja simulaatio (M&S).....	8
<b>2 Kehitystä.....</b>	<b>10</b>
2.1 Mallisuunnittelun kiteytyminen omaksi alakseen.....	10
2.2 Mallikielien alkuperät.....	11
2.3 ”Metodisodat” .....	12
2.4 UML.....	12
2.5 SysML.....	14
2.6 Mallikielien kehityksen takana olevat motivaatiot.....	15
<b>3 Metodologia.....</b>	<b>16</b>
3.1 Metodologian määritelmä.....	16
3.2 Metodologioiden akateeminen vertailu.....	16
<b>4 Negatiivisia tuloksia.....</b>	<b>18</b>
4.1 Milloin ei kannata käyttää järjestelmäsuunnittelua?.....	18
<b>5 Tulevaisuus.....</b>	<b>19</b>
<b>6 Lopetus.....</b>	<b>21</b>
6.1 Lähteet.....	21

# 1 Pohjatietoa ja määritelmiä

Tässä tutkielmassa järjestelmäsuunnittelulla tarkoitetaan alaa, joka nimensä mukaan auttaa järjestelmien suunnittelussa. Kommunikaatiolla tarkoitan sitä, kuinka suunnittelijat pystyvät kommunikoimaan toisilleen järjestelmän toiminnasta ja sen kehityssuunnasta. Hallinnalla tarkoitetaan tässä tutkielmassa sitä, kuinka järjestelmän suunnittelijat pystyvät hallitsemaan järjestelmän kehitystä ja toimintaa pidemmälläkin aikavälillä. Ymmärryksellä tarkoitan sitä, kuinka järjestelmäsuunnittelu auttaa suunnittelijoita ymmärtämään monimutkaisen järjestelmän toimintaa paremmin. Nämä tavoitteet ovat tämän tutkielman kontekstissa järjestelmäsuunnittelun tärkeimmät tehtävät, ja usein tukevat toisiaan. Tässä tutkielmassa perehdytään järjestelmäsuunnittelun historiaan ja kehitykseen, järjestelmäsuunnittelun kehitysvaiheiden tärkeisiin innovaatioihin, ja siihen miten nämä innovaatiot auttavat järjestelmäsuunnittelun tärkeimmissä tehtävissä.

Tämä voi sinänsä tarkoittaa mitä tahansa prosessia, mutta useimmiten puhutaan teknisistä prosesseista, kuten projektisuunnittelusta, työkulusta ja tuotesuunnittelusta.

Järjestelmäsuunnittelua voidaan tavallaan pitää yhtä vanhana kuin suuria useamman ihmisen projekteja. Hyvin vapaa määritelmä voisi pitää vaikka eri työtehtävien jakamista eri ihmisille järjestelmäsuunnitteluna. Kuitenkin järjestelmäsuunnittelu itse tunnistettuna alana alkoi vasta 1900-luvun puolella välissä. Pääimmäisiä motiiveja järjestelmäsuunnittelun kehitykselle omaksi alakseen olivat lähinnä järjestelmien kasvava monimutkaisuus sekä komponenteissa että metatasolla. Komponenteista eli itse laitteen tai ohjelmiston toimivista osista itsestään tuli niin monimutkaisia, että tarvittiin selkeää dokumentaatiota, jotta kaikki muutkin kuin itse komponentin suunnittelijat ymmärtäisivät niitä. Myös koko suunnitteluprosessi monimutkaistui ihmisten ja työvaiheiden lisääntyessä. Lisääntynyt monimutkaisuus prosessin joka vaiheessa motivoi siirtoa organisoituun järjestelmään, missä kommunikaatio projektin eri osapuolien välillä olisi selkeää ja suoraviivaista. Tällaiset lähtökohdat synnyttivät modernin järjestelmäsuunnittelun. Varhaisia yrityksiä missä järjestelmäsuunnittelu otettiin vakavasti on esimerkiksi amerikkalainen Bell Labs, joilla oli jo 1950-luvulla oma osasto järjestelmäsuunnittelulle. <sup>i ii</sup>

On huomioimisen arvoista, että järjestelmäsuunnittelulla ei ole täysin yksimielistä määritelmää. Määritelmässä on eroja, mutta erot ovat usein pieniä ja johtuvat osittain sanojen muotoilusta. Itse järjestelmäsuunnittelun määritelmät eivät ole myöskään aina täysin yhtenäisiä. NASA:n määritelmässä on sisällytetty asiakastarpeet ja vaatimukset keskeiseksi osaksi järjestelmäsuunnittelun määritelmää. <sup>iii</sup> Järjestelmäsuunnittelu on tässä mielessä monialainen laji, sekä siinä mielessä että se yhdistelee monia eri tieteen aloja, mutta myös siinä mielessä, että siinä yhdistyvät ihmisten käytös ja ajattelu monimutkaisen järjestelmän toiminnan kanssa.

### 1.1 Järjestelmän määritelmä

Järjestelmäsuunnittelussa puhutaan paljon järjestelmistä, ja sanalla ”järjestelmä” on jokseenkin tietty tarkoitus tässä kontekstissa. Sana järjestelmä tarkoittaa eri konteksteissa eri asioita. Sana tuli suosituksi termodynamiikan yhteydessä, missä se tarkoittaa jotain tiettyä aluetta jonka rajojen läpi voi kulkea termodynaamista energiaa. Tämän energiansiirron kautta termodynaaminen järjestelmä on tekemisissä ulkomaailman kanssa. <sup>iv</sup> Biologiassa voidaan puhua ekosysteemeistä, jotka ovat tavallaan järjestelmiä. Ekosysteemi terminä on valunut uudestaan insinöörimaailmaan esimerkiksi ohjelmoinnissa ja tuoteperheissä. <sup>v</sup> Näillä määritelmillä on yhteistä se, että ne ovat rajattuja tiloja joihin sisältyy liikkuvia osia. Järjestelmä on siinä mielessä samanlainen kuin kone, että siinä on pakko olla muutosta ollakseen järjestelmä. Sen täytyy myös olla erillinen epämääräisestä ”kaikkeudesta”. Järjestelmäsuunnittelussa järjestelmän määritelmässä on samoja elementtejä, mutta joitakin kriittisiä eroja. Melkein mitä tahansa insinööriyön tulosta voisi pitää järjestelmänä, jos sanaa järjestelmä ajattelee tarpeeksi laajasti. Esimerkiksi lentokoneen siipi on järjestelmä, sillä se sisältää liikettä ja toimintoa. Näitä kahta elementtiä voisi pitää riittävänä siihen, että siiven kategorisoisi järjestelmäksi. Tämä kuitenkin tekisi järjestelmäsuunnittelusta aivan liian laajan alan, ja sanalla ei olisi juurikaan merkitystä. Järjestelmäsuunnittelussa puhutaankin siis systeemeistä, joiden vaiheita ei pysty helposti käsittämään kokonaisuudessaan. Järjestelmäsuunnittelun tarkoituksena on tehdä näistä monimutkaisista systeemeistä luettavampia, jotta järjestelmäsuunnittelija voisi muuttaa ja optimoida järjestelmää.

## 1.2 ”Malli” terminä ja käsitteenä

Sana ”malli” esiintyy ymmärrettävästi paljon mallipohjaisessa järjestelmäsuunnittelussa. Sanana ”malli” tarkoittaa monessa eri aihealueessa montaa eri asiaa, kuten roolimalli, muotimalli, pienoismalli ja matemaattinen malli. Asiaa vaikeuttaa vielä se, että järjestelmäsuunnittelun sisällä sanalla ”malli” on monta eri tarkoitusta; sanaa käytetään vaihdannaisesti puhuttaessa sekä pienoismalleista, simulaatioista ja matemaattisista malleista, mitkä kuvaavat yksittäisen tuotteen tai komponentin kehitysvaihetta tai olemusta, sekä malleista, joilla tarkoitetaan laajempia, koko järjestelmän ja järjestelmän logistiikan kattavia käsitteitä. Asiaa vaikeuttaa vielä enemmän se, että mallipohjaisen järjestelmäsuunnittelun esimerkit voivat käyttää tätä sanaa kuvaamaan näitä molempia asioita osana järjestelmän kokonaisuutta. <sup>vi</sup>

## 1.3 Mallisuunnittelun määritelmää

Sanalla ”malli” tarkoitetaan yksinkertaistettua versiota jostakin asiasta, joko siksi että sitä olisi helpompi suunnitella tai siksi että sitä on helpompi esitellä ihmisille, jotka eivät ole välittömästi suunnittelussa mukana, kuten asiakkaat tai rahoittajat. <sup>vii</sup> Aikaisemmin sanalla malli tarkoitettiin suuremman rakenteen suunnitelmia. <sup>viii</sup> Malli voi olla eräänlainen ”kartta” mikä kuvaa järjestelmän toimintaa, tai se voi olla yksinkertaistettu laskelma, joka kuvastaa järjestelmän vaikutusta. Useimmiten kun yleiskäytössä kuulee sanan ”malli”, tulee mieleen joko fyysinen pienoismalli tai tietokone malli. Nämäkin ovat järjestelmäsuunnittelumielessä malleja, ja niillä on käyttöä samassa missä muillakin malleilla. Mallien perimmäinen tarkoitus on kommunikoida konsepteja muille suunnittelijoille ja rahoittajille, jotta voitaisiin välttää väärinkäsityksiä ja näin vähentää myöhempää kallista ja aikaa vievää muutosta ja korjausta. Koska mallin integroiminen järjestelmäsuunnitteluun vaatii myös järjestelmään liittyvän teknologian formalisointia tekstipohjaista järjestelmäsuunnittelun metodologiaa enemmän, niin tekniset riskit, kuten se että jokin järjestelmään liittyvä teknologia ei pystykään toteuttamaan suunniteltua tehtävää, vähenee. Mikäli uudelleensuunnittelua tulee (mikä on melkein väistämätöntä monimutkaisessa järjestelmässä), niin mallisuunnittelu antaa kiinteän lähtökohdan ja voi näin helpottaa uudelleensuunnitteluprosessia. <sup>ix</sup>

Koska malli ei pohjimmiltaan voi olla yhtä monimutkainen kuin valmis tuote tai järjestelmä, on mallin tekemisessä pakko tehdä valintoja. Mallin täytyy keskittyä siihen tiettyyn asiaan, mitä kyseisessä suunnitteluvaiheessa halutaan eniten tarkastella.

#### 1.4 Mallisuunnittelu vastaan dokumenttisuunnittelu (MBSE vs. DBSE)

Kun puhutaan mallisuunnittelusta, täytyy myös ottaa huomioon mallisuunnittelun ympäröivä konteksti. Useimmiten mallisuunnittelu asetetaan vastakkain dokumenttisuunnittelun kanssa, jota voisi pitää ”perinteisempänä” järjestelmäsuunnittelumetodina. Dokumenttisuunnittelu nimensä mukaisesti keskittyy vahvasti kirjoitettuihin dokumentteihin ja muuhun perinteisempään tiedon tallentamiseen järjestelmän tai projektin toiminnasta. <sup>x</sup> Tällä suunnittelumetodilla on edellä mainittuja haittoja, kuten puhekielestä tulevia epätarkkuuksia ja vaikeuksia päivittää kaikkien dokumentit automaattisesti ja samanaikaisesti mallin muuttuessa. Vaikka mallipohjainen suunnittelu on uudempi metodi, on dokumenttipohjaisella suunnittelulla silti kannattajia. <sup>xi</sup> Sandian laboratorion kirjallisuuskatsaus sai tulokseksi, että mallisuunnittelun käyttöönotto oli tutkituissa tapauksissa kannattava valinta. <sup>xii</sup>

Dokumenttipohjainen suunnittelu ei kuitenkaan välttämättä ratkaise järjestelmäsuunnitteluprosessin ongelmia. Jean-Philippe Lerat väittää paneelissaan Three Reasons why Document-based SE (usually) works better than (most of) MBSE että mallipohjainen järjestelmäsuunnittelu ei itsessään korjaa järjestelmäsuunnitteluympäristön ongelmia. Leratin mukaan mallipohjainen järjestelmäsuunnittelu ei välttämättä ole esimerkiksi järjestelmistä kommunikaatiossa dokumenttipohjaista järjestelmäsuunnittelua tehokkaampaa, mikäli mallipohjaista järjestelmää ei ole tehty helposti ymmärrettäväksi. Siinä missä dokumentti etenee loogisesti kirjoittajan suunnitelman mukaan ja useimmiten sisältävää esimerkiksi hakemiston, mallit voivat olla vaikeaselkoisia ja ne saatetaan hakea suuresta järjestelemättömästä varastosta. Tässä mielessä mallien olemassaolo itsessään ei ratkaise ongelmia kommunikaatiossa, vaan pohjalle tarvitaan toimivat järjestelmät jotka tukevat mallipohjaista järjestelmäsuunnittelua. <sup>xiii</sup>

Dokumenttipohjaisella suunnittelulla ja mallipohjaisella suunnittelulla on molemmilla rooli järjestelmäsuunnittelun tavoitteisiin nähden. Dokumenttipohjaisella suunnittelulla voidaan tosiaankin saavuttaa kaikki järjestelmäsuunnittelun perustavoitteista, mutta monessa mielessä voisi ajatella, että mallipohjainen suunnittelu täyttää tavoitteet paremmin. Toisaalta valinta riippuu myös projektin koosta, sillä hyvään mallipohjaiseen järjestelmän tarvitsee panostaa paljon. Voisi ajatella, että mallipohjaisessa suunnittelussa kiteytyy ajatus siitä, että kun alussa panostaa ja tekee asiat heti kunnolla niin sitten ei tarvitse korjata niin paljon myöhemmissä vaiheissa. Tämä näkyy esimerkiksi Sandian laboratorion kirjallisuuskatsauksessa, missä huomattiin että vaikka suurin osa ajasta mallipohjainen järjestelmäsuunnittelu vähensi virheitä tulevaisuudessa, niin se saattoi myös tehdä tuotteista ja prosesseista suurempia, hitaampia ja monimutkaisempia. <sup>xiv</sup>

## **1.5 Mallinnus ja simulaatio (M&S)**

Mallisuunnitteluun vahvasti liittyvä mutta silti eri asiaa tarkoittava termi on mallinnus ja simulaatio (Modelling and Simulation, M&S). Tämä on mallisuunnitteluun liittyvä tutkimuksen ja insinööriyön ala, joka kuitenkin painottaa enemmän itse mallien kehittämistä toisin kuin mallisuunnittelu, joka painottaa pikemminkin mallien käyttöä koko järjestelmän tasolla ja sitä, miten mallien käyttäminen vaikuttaa järjestelmän toimivuuteen ja järjestelmän eri osapuolien väliseen kommunikaatioon. Voisi siis sanoa, että mallinnus ja simulaatio on mallisuunnittelun alalaji tai pienoismalli. Lisäksi terminä mallinnus ja simulaatio on enemmänkin käytössä yksittäisten asioiden ja koneiden mallinnuksessa, kun taas mallisuunnittelulla puhutaan usein järjestelmien ja alajärjestelmien mallintamisesta. <sup>xv</sup> On tärkeää huomioida, että sanaa ”malli” käytetään molemmista asioista usein ilman erottelua, mikä tekee näiden kahden erottamisen toisistaan joskus hankalaa. Sanaa ”simulaatio” käytetään myös yksittäisten tuotteiden ja suunnitelmien simuloimisesta samalla tavalla kuin sitä käytetään laajemmista, kokonaisten järjestelmien simulaatioista tai järjestelmien mallien simulaatioista.



Mallinnus ja simulaatio tukee järjestelmäsuunnittelun tavoitteita aika lailla samoin tavoin mitä itse järjestelmäsuunnittelu, mutta voisi ajatella, että termi tarkoittaa enemmänkin asioita tietyissä suunnittelun konteksteissa. Malli tai simulaatio auttaa suunnittelijoita ymmärtämään toistensa ajatuksia, ja se voi hyvinkin helpottaa suunnittelijoita huomaamaan mitä potentiaalisia heikkouksia suunnitelmassa on ja mihin suuntaan projektin pitäisi jatkua, että projektin tavoitteet saavutetaan. Mallinnus ja simulaatio voi auttaa suunnittelussa ja näin vähentää virheitä. <sup>xvi</sup>

## 2 Kehitystä

### 2.1 Mallisuunnittelun kiteytyminen omaksi alakseen

Mallisuunnittelu, niin kuin kaikki muukin suunnittelussa ja insinööriyössä, on olemassa syystä. Aikaisemmin mainitut syyt sille, miksi mallipohjainen suunnittelu on käytännöllinen ja/tai haluttava metodi järjestelmien suunnitteluun ovat jokseenkin samanlaisia niihin syihin mitkä johtivat koko alan syntyyn. Mallipohjaiselle suunnittelulle alkupään motivaationa on ollut muun muassa hieman aikaisemmin suosituksi tullut ”Agile” projektinohjaus- ja tuotekehityssuuntaus. ”Agile” metodit vaativat usein pikaista iteraatiota ja nopeita muutoksia asiakastarpeiden ja kehittäjien ymmärryksen kehittyessä ja muuttuessa. Tällainen nopea muutos voi olla hankala toteuttaa monimutkaisia tuotteita tai systeemejä kehittäessä. Lisäksi dokumentaatiosta johtuvat mahdolliset virheet, kuten epätarkka puhekieli tai yksinkertaisesti kirjoitusvirheet, ovat kalliimpia korjata monimutkaisemman tuotteen kanssa. Monimutkaisen tuotteen kehittämiseen vaadittu aika myös pahentaa ongelmaa, sillä korjauksien hinta ja vaadittu aika kasvavat merkittävästi mitä pidemmälle tuote on suunniteltu. Ideana on, että mallintamiseen ja mallipohjaiseen järjestelmiin käytetään aluksi enemmän aikaa ja resursseja, jotta myöhempää ajan ja resurssien investointia voitaisiin minimoida. <sup>xvii</sup>

Ensimmäinen hyvin tunnettu maininta mallisuunnittelussa sen modernissa merkityksessä on Wayne Wymoren kirjassa *Model-Based Systems Engineering* vuodelta 1993. <sup>xviii</sup> Wymore kirjassaan on vahvasti mallipohjaisen suunnittelun kannalla, jopa sisällyttäen mallien tärkeyden itse järjestelmäsuunnittelun määritelmään, vaikka vakiintunutta jakaumaa mallipohjaisen ja dokumenttipohjaisen suunnittelun välillä ei vielä ollut. Wymore sisällyttää perinteisen insinööriyöhön kuuluvan mallintamisen, lähinnä fyysiset mallit, prototyypit ja piirrokset, ja lisää niiden joukkoon uudenlaisen mallin. Tätä järjestelmäsuunnitteluun tarkoitettua mallia Wymore kutsuu järjestelmäteoreettiseksi malliksi. (s. 18-20) Tällainen malli on enemmän keskittynyt pienemmän katsauksen mallien sisällyttämiseen suurempaan systemaattiseen kokonaisuuteen, ja tämän kokonaisuuden kykyyn saavuttaa järjestelmän perustarpeet. Wymore myös painottaa malleista puhuessan sitä, kuinka mallien täytyy olla yksinkertaistetusta luonnostaan huolimatta tarpeeksi hyviä jotta niiden pohjalta voidaan rakentaa lopullinen valmis järjestelmä. Tämä asenne ja perusteellisuus malleihin erottaa järjestelmäsuunnittelussa käytettävät järjestelmätason mallit muista yksinkertaisemmista havainnollistamiseen tarkoitetuista malleista.

Diagrammi on työkalu, joka havainnollistaa mallisuunnittelun tarkoitetun hyödyn suunnitteluprosessin aikana. Ideana on, että alussa käytetty aika ja vaiva mallin ja tukevien järjestelmien kehitystä varten on kannattavaa, koska se säästää aikaa myöhemmiltä kalliilta korjauksilta.

Vaikka toisaalta voisi ajatella, että mallisuunnittelu on jossain mielessä ollut yhtä kauan olemassa kuin suunnittelu yleensä, on sen kiteytyminen omaksi määritellyksi alakseen antanut sille kehittymismahdollisuuksia. Jos mallisuunnittelulla puhutaan tietystä asiasta, niin muut ymmärtävät helpommin mistä tosiasia on kyse, ja järjestelmäsuunnittelun näkemyksessä kommunikaatio helpottuu. Mallisuunnittelun paikka omana alanaan antaa myös mallisuunnittelulle mahdollisuuden erikoistua tavalla, joka tekee siitä tehokasta.

## **2.2 Mallikielien alkuperät**

Mallinnuksessa ja mallisuunnittelussa puhutaan usein mallikielistä. Mallikielet ovat keinotekoisia kieliä, joiden tarkoituksena on olla standardisoitu terminologia tietylle asialle. Järjestelmäsuunnittelun kontekstissa mallikielet sisältävät tiettyjä termejä järjestelmän sisäisille elementeille ja elementtien välisille interaktioille. Termit eivät välttämättä ole sanoja, vaan mallikielet koostuvat usein sekä tietyistä sanoista että graafisista havainnollistuksista. Mallikielet kuuluvat isompaan mallinnusstandardien kategoriaan, mikä kattaa myös tiedonvaihdon standardit ja standardeja liittyen itse mallinnukseen ja simulaatioon.

Koska “malli” sanan monimuotoisuus aiheuttaa jo valmiiksi hämmennystä, on asiallista ajatella että itse mallisuunnittelun sisällä sanoilla tulisi olla tietyt tarkoitukset jotta mallisuunnittelun yksi tärkeimmistä olemassaolon syistä, eli kommunikaatio eri osapuolten välillä, onnistuisi sujuvasti ja vaivattomasti. Tämä toisaalta tuo takaisin aikaisemmin mainitun tiedostopohjaisen suunnittelun ongelman, eli kielenkäytön ja kirjoitusvirheet. Ongelma ei kuitenkaan ole kovin suuri, koska mallikielet ovat enemmän ohjelmiston tapaisia ja siksi sanat ovat huomattavasti vähemmän tulkinnanvaraisia.

Mitä tulee mallipohjaiseen järjestelmäsuunnitteluun, yleinen mallikieli on Systems Modelling Language eli SysML ja muut OMG:n (Object Modelling Group, suom. Oliomallinnusryhmä) standardeihin sopivat mallikielet.<sup>xix</sup> SysML johtaa juurensa aikaisempaan UML:ään, joka juurtuu amerikkalaisen ohjelmoijan ja ohjelmistotuottajan Grady Boochin kehittämään ja käyttämään metodiin nimeltä Boochin metodi.<sup>xx</sup>

Boochin metodi oli olio-ohjelmointiin tarkoitettu metodi, jonka päätarkoituksena oli olla oliomallinnuskieli ja myös sisällyttää suositeltavat toimenpiteet ohjelmistoa kehittäessä. Boochin metodi keskittyi lähinnä olioihin, luokkiin ja luokkien keskenäisiin suhteisiin. Näissä ulottuvuuksissa sillä on paljon yhtäläisyyksiä myöhempien mallikielien kanssa, osittain kirjaimellisesti termeissä ja käsitteissä mutta enemmänkin yhtäläisyyksiä motivaatioissa, tarkoitusperissä ja perimmäisessä filosofiassa parhaasta tavasta suorittaa monimutkainen prosessi alusta loppuun. Boochin metodi oli vain yksi monista oliopohjaisesta mallinnusmetodeista, mitä tuohon aikaan oli saatavilla. Moni halusi kehittää oman versionsa oliopohjaisesta metodista, ja yhtenäisyyden puute teki kommunikaatiosta vaikeaa. Tämä taas heikensi kaikkien mallien tehokkuutta, joten suunnittelijoilla oli selkeä motiivi siirtyä yhtenäiseen mallinnuskieleen. Olemassaolevien oliopohjaisten metodien parhaat puolet yhdistettiin, kun ymmärrys ohjelmistosuunnittelusta parani, ja lopulta kehitys päätyi yhdistettyyn mallinnuskieleen eli UML:ään.<sup>xxi xxii</sup>

### 2.3 ”Methodisodat”

Olio-ohjelmointi oli melko tuoretta vielä 1980-luvulla, joten käytännöt eivät olleet vielä kiveen hakattuja. Oliopohjaiset ohjelmistosuunnittelumetodit olivat vasta muodostumassa. Näistä tekijöistä johtuen markkinat olivat herkäät ja suunnittelumetodeille oli tarvetta. Boochin metodin ilmestymisen jälkeen ilmestyi monia oliopohjaisia ohjelmistosuunnittelun metodeja. Kaikki halusivat markkinaosion tai ainakin hetkellistä huomiota omaan työhön tai omiin projekteihin. Lopulta näitä uusia metodeja oli kymmeniä, ja metodin valitseminen projektia varten muuttui hankalaksi. Ohjelmistosuunnittelijat kyllästyivät tähän ja ohjelmistosuunnittelun piireissä yritettiin yhdistellä metodeja. Tästä lopulta syntyi UML.<sup>xxiii</sup>

### 2.4 UML

Boochin metodista myöhemmin kehittyi UML eli Universal Modelling Language. UML on myös lähinnä keskittynyt suurten ohjelmistoprojektien mallintamiseen. Sen kehitys alkoi 1990-luvun alussa, ja sen tarkoituksena oli standardisoida järjestelmien mallintaminen.

Painopisteinä olivat selkeys, mikä arkisessa puhekielessä ei ole riittävä kaikkiin tarkoituksiin, ja standardisointi projektien eri osapuolten välillä. Standardisoinnin tarkoituksena oli myös ylittää organisaatorajat, jotta myös välittömän projektin ulkopuoliset jäsenet voisivat ymmärtää mihin suuntaan projekti on menossa. UML:än yksimielisyyden selkeyden tarkoituksena oli selventämisen lisäksi myös dokumentointi. Dokumentointi universaalissa mallikielessä olisi kriittisesti vähemmän altis puhekielen epätarkkuuksille ja inhimillisille virheille, kuten kirjoitusvirheille. Näin projektin dokumentaatio olisi ainakin teoriassa hyödyllisempää ja helppolukuisempaa mitä puhdas tekstipohjainen dokumentaatio olisi. Mallikielen käyttö myös voisi parantaa itse ydinprojektin sivuelementtien dokumentointia, kuten aikaisempien prototyyppien ja testien dokumentointia. UML:än tarkoituksena oli myös auttaa monimutkaisten ohjelmistojen visualisoinnissa, mikä hyödyttää sekä ohjelmoijia että asiakkaita. Ohjelmoijat ovat paremmin perillä koko projektin skaalasta, ja asiakkaat saavat paremmin käsityksen siitä, miten haluttu tulos saadaan aikaan. UML:än tarkoituksena oli myös olla niin tarkka ja kattava, että sen avulla suunniteltuja malleja voitaisiin suoraan käyttää lopullisen tuotteen valmistamisessa. Näin mallin luomiseen käytetty aika ei olisi pelkästään selvennystä, kommunikaatiota ja dokumentaatiota varten, vaan se olisi myös itse projektia suoraan edistävää työtä. <sup>xxiv</sup>

Mallikielenä UML koostuu muutamasta erilaisesta asiasta, ja tietyistä interaktioista näiden asioiden välillä. Nämä asiat ja niiden väliset suhteet voi visuaalisesti havainnollistaa diagrammissa. Tärkeää on kaikkien näiden elementtien tarkat semanttiset määritelmät, ja kuinka diagrammin tekeminen asettaa järjestelmän elementtien väliset interaktiot mahdollisimman yksinkertaisesti näkyville. <sup>xxv</sup>

UML:ssä on kuitenkin haittapuolia. Järjestelmän käyttöönotto vaatii paljon perehtymistä. Järjestelmä ei toimi, elleivät kaikki osapuolet ole perillä sen toiminnasta. Tarkat semanttiset määritelmät eivät auta, mikäli kaikki eivät tarkalleen tiedä kaikkien termien oikeaa tarkoitusta. Universaalien järjestelmän tulisi kattaa suurin osa asiallisista käyttötarkoituksista, joten järjestelmästä tulee väistämättä suuri. Järjestelmän oppiminen niin hyvin, että sillä voi tuottaa käyttökelpoisia malleja, vaatii paljon aikaa. <sup>xxvi</sup>

UML on edistänyt järjestelmäsuunnittelun tavoitteita monessa mielessä, lähinnä helppoudessa ja kommunikaatiossa. UML nimensä mukaan on eräänlainen kieli, jonka tarkoituksena ei ole pelkästään auttaa suunnittelijoita kommunikoimaan toisilleen projektista, vaan sen voisi myös ajatella auttavan suunnittelijaa ymmärtämään projektia paremmin. Ylipäätään voisi sanoa että metodologioiden kehittyminen mallikieliksi ja etenkin UML:än synty antoi osviittaa tuleville järjestelmäsuunnittelun työkaluille. Esimerkiksi paljon käytössä oleva SysML perustuu UML:ään. <sup>xxvii</sup>

Mallikielistä on tullut kriittinen järjestelmäsuunnittelun työkalu, joten mallikielet soveltuvat hyvin tässä tutkielmassa esitettyihin järjestelmäsuunnittelun tavoitteisiin.

## 2.5 SysML

UML:stä erkani myöhemmin enemmän järjestelmäsuunnitteluun keskittynyt SysML eli Systems Modelling Language. SysML:än tarkoituksena oli alunperin olla eräänlainen laajennus UML:ään, joka olisi vähemmän keskittynyt ohjelmistosuunnitteluun ja ohjelmointiin. SysML olisi sen sijaan enemmän keskittynyt järjestelmäsuunnittelun tehtäviin laajemmassa perspektiivissä. Vaikka SysML on ”laajennus”, niin se kuitenkin poistaa monia ohjelmistoon keskittyneitä elementtejä kielestä. UML:än alkuperäisistä kolmestatoista diagrammityyppistä SysML säilyttää seitsemän ja lisää kaksi. Tämä tekee siitä kevyemmän, ja mikä tärkeämpää, helpomman oppia verrattuna UML:ään. Näin SysML on ainakin teoriassa yksinkertaisempi ottaa käyttöön järjestelmäsuunnitteluprosessia varten. <sup>xxviii</sup>

Muitakin kilpailevia järjestelmäsuunnittelun mallinnuskieliä on olemassa, kuten Modelica. Vaikka mallinnuskielistä on nykyään enemmän yhteisymmärrystä mitä niistä oli 1980-luvulla, niin silti on monia kilpailevia standardeja, jotka yrittävät tuottaa tehokasta järjestelmäsuunnittelun työkalua hieman eri metodein tai eri tarkoituksiin. Todennäköisesti ei tule koskaan olemaan tilannetta, missä kaikki käyttäisivät samaa mallikieltä tai standardia. Mallikieli, joka kattaisi kaikkien prosessien käyttötarkoitukset, olisi liian suuri ollakseen kovinkaan käytännöllinen.

SysML tukee järjestelmäsuunnittelun tavoitteita helpottamalla järjestelmäsuunnittelijoiden työtä mallikielen käytössä ja opettelemisessa. SysML:ssä on kaksi hyötyä verrattuna UML:ään. SysML:än käyttöönottoa varten ei tarvitse opetella niin montaa erilaista käyttötapausta, ja täten se vähentää mallisuunnittelun alkutaakkaa jokseenkin. Se myös yksinkertaistaa, sillä se ei sisällä niin montaa elementtiä. Toisaalta SysML myös sallii uusia mahdollisuuksia, sillä se sisältää uusia elementtejä jotka voivat olla tärkeitä kyseisessä järjestelmäsuunnittelun tehtävässä.

## 2.6 Mallikielien kehityksen takana olevat motivaatiot

Mallikielien ja muiden järjestelmän toimintaa tai suunnittelua kuvastavien toiminnallisten työkalujen kehityksessä on pyritty usein saavuttamaan kaksi ymmärrettävää mutta usein ristiriitaista tavoitetta. Nämä kriittisimmät tavoitteet ovat helppokäyttöisyys ja mahdollisimman laajasti skaalattava toiminta. Järjestelmäsuunnittelun yksi tärkeimmistä tavoitteista on tehdä monimutkaisten järjestelmien käsittämisestä ja suunnittelemisesta ihmisille helpompaa. Mallikielet ovat luonnollisesti tämän tavoitteen jatkumoa, jossa ne yrittävät tehdä järjestelmäsuunnittelusta helpompaa tehdä ja käsittää, jotta järjestelmäsuunnittelu itse voi tehdä järjestelmistä helpompia tehdä ja käsittää. Toisaalta mallikielistä on yritetty kehittää universaaleja. Universaalissa ja yhtenäisessä mallikielessä on se hyöty, että yhtenäisyyten standardioituessa kaikki pystyisivät ymmärtämään sitä. Näin järjestelmien elementeistä ja järjestelmän toiminnasta itsestään kommunikointi muille osapuolille olisi vaivatonta. Universaali ja yhtenäinen mallikieli myös tekisi asian opettelemisesta helpompaa, ja tiettyyn yhtenäiseen mallikieleen perehtyneet asiantuntijat voisivat helposti siirtyä projektista toiseen ja tuoda asiantuntemuksensa mukanaan.

Näissä tavoitteissa on kuitenkin jonkin verran ristiriitaa. Jotta mallikieli olisi helppokäyttöinen ja hyvin ymmärrettävä, niin se ei voi sisältää liikaa elementtejä jotka on pakko opetella mallikielen tehokasta käyttöä varten. Mutta jos mallikieli ei voi sisältää tarpeeksi elementtejä, niin sitten se ei voi myöskään olla tarpeeksi tarkka tarpeeksi moneen eri käyttötarkoitukseen kattaakseen kaikenlaisien projektien ja järjestelmien käyttötarpeet. Jos mallikieli olisi tarpeeksi universaali toimiakseen kaikille järjestelmäsuunnittelun projekteille, niin sen täytyisi sisältää niin paljon erilaisia elementtejä eri reunatapauksille että sen opettelusta tulisi melkein mahdotonta. UML:ää on jo valmiiksi kritisoitu siitä, että se on liian monimutkainen ollakseen käytännöllinen.<sup>xxix</sup> Siltikin on järjestelmäsuunnittelun käyttötarkoituksia, joihin UML ei sovi. UML ja sen suhde SysML:ään toimii niin, että SysML käyttää UML:än ”kieltä” mutta ei sisällä kaikkia sen elementtejä ja lisää siihen vain muutamia omia elementtejä. Näin henkilö, joka osaa käyttää UML:ää, voisi periaatteessa siirtyä projektiin joka käyttää SysML:ää ilman että hänen tarvitsisi opetella käyttämään mallikieliä kokonaan uudelleen.

## 3 Metodologia

### 3.1 Metodologian määritelmä

Järjestelmäsunnittelussa sana metodologia tarkoittaa laajempaa käsitettä, joka sisältää järjestelmäsunnittelun ongelman yksittäiset elementit. James N. Martin määrittelee kirjassaan *Systems Engineering Guidebook: A process for developing systems and products* metodologian koostuvan neljästä elementistä. Ensimmäisenä on prosessi, mikä koostuu itse järjestelmän toiminnan loogisista vaiheista. Prosessi ei koske sitä, miten mikään järjestelmän vaihe tulee käytännössä toimimaan. Prosessin abstraktio pois käytännöstä sallii itse järjestelmän toimintavaiheiden helpomman jäsentelyn ja voi selkeyttää päätösten tekoa. Prosessia voi ajatella myös kysymyksenä mitä järjestelmän kuuluisi tehdä. Toinen elementti on metodi. Metodi määrittää, miten prosessin eri vaiheet käytännössä tapahtuvat. Se on siinä mielessä prosessin vastakohta. Metodia voisi ajatella kysymyksenä miten järjestelmä voisi tehdä mitä sen kuuluu tehdä. Kolmas elementti on työkalut, jotka korostavat prosesseja ja metodeja. Työkalujen tarkoituksena on tehdä prosessien ja metodien valinnoista ja suunnittelusta tehokkaampaa. Työkaluja ovat esimerkiksi mallikielet, mallinnuksen ja simulaation työkalut ja arkkitehtuurityökalut. Neljäntenä elementtinä on ympäristö, mikä kattaa ulkoiset metodologian elementteihin vaikuttavat tekijät. Näitä voivat olla vaikka kulttuurillisia tai sosiaalisia tekijöitä. Ne voivat myös olla logistisia rajoitteita tai fyysisen ympäristön vaikuttavia tekijöitä. Metodologiaa voi ajatella tapana, millä koko järjestelmäsunnittelun haastetta lähestytään. Metodologia määrittelee, mitä ongelmia lähestytään missä järjestyksessä ja miten. <sup>xxx</sup>

Metodologiat auttavat järjestelmäsunnittelun tavoitteita kohtaan siinä mielessä, että se voi auttaa prosessia tapahtumaan sujuvammin. Sen sijaan, että täytyisi joko itse kehittää oma tehokas työskentelytapa tai antaa kaikkien toimia miten toimivat, metodologia on valmis eetos jota seurattaessa kaikki ovat yksimielisiä projektin etenemissuunnasta. Hyvä metodologia vähentää projektin etenemisen aikana syntyviä virheitä ja auttaa päättämään, miten nykyinen vaihe projektissa tulisi suorittaa loppuun.



### 3.2 Metodologioiden akateeminen vertailu

Metodologioita on järkevää vertailla kuten työkaluja ja mallikieliä. Eri metodologiat on suunniteltu erilaisissa ympäristöissä ja eri filosofioilla, joten niillä tulee luonnollisesti olemaan omanlaisiansa heikkouksia ja vahvuuksia. Koska metodologiat ovat monen muun järjestelmäsunnittelun työkalun tavoin ihmiskäyttöliittymän järjestelmiä, niin vertailussa on vähemmän painoa objektiivisilla parametreilla ja enemmän painoa järjestelmän käyttäjän henkilökohtaisilla totumuksilla ja mielipiteillä. Erilaiset metodologiat ja työkalut soveltuvat myös paremmin tai huonommin erilaisiin haasteisiin, joten on tärkeää osata valita oikea työkalu oikeaan tehtävään. T. Weilkiens et al. tekivät vertailua erilaisista järjestelmäsunnittelun metodologioista. Tutkimuksessaan he huomauttavat, että tällainen akateeminen vertailututkimus metodologioihin on harvinaista tai täysin uutta tutkimuksen teon aikaan vuonna 2016. Tutkimus käytti metodologioiden arviointiin kehitettyä arviointikriteeriä nimeltä FEMMP (Framework for the evaluation of MBSE methodologies for practitioners) (suom. Viitekehys mallipohjaisen järjestelmäsunnittelun metodologioiden arvioinnista harjoittajia varten). FEMMP:in tarkoituksena ei sinänsä ole selvittää, mikä saatavilla olevista metodologioista on objektiivisesti paras. Tällainen selvitys ei olisi kovinkaan mahdollinen tai hyödyllinen, sillä eri metodologioilla on omat hyötynsä erilaisiin käyttötarkoituksiin. Metodologioiden valintaan vaikuttaa myös metodologian käyttäjän perehtyminen sen käytäntöihin ja toimintatapaan. Näin FEMMP ei yritä olla vahva neuvo siitä, mikä metodologia on paras valita. FEMMP yrittää pikemminkin olla objektiivinen ja neutraali katsaus siihen, mitkä elementit ovat eri metodologioiden heikkouksia ja vahvuuksia.

xxx

FEMMP käyttää useita eri parametreja metodologioiden arviointiin, kuten tarkkuutta, laajennettavuutta, helppokäyttöisyyttä ja mahdollisen tuen saatavuutta. Weilkiensin et al. alkuperäisiä parametreja on myös pyritty laajentamaan. Näitä lisäyksiä ja laajennuksia ovat yrittäneet esimerkiksi Christophe Ponsard ja Valery Ramon. <sup>xxxii</sup>

FEMMP ja muut akateemiset vertailut järjestelmäsunnittelun metodologioista voivat kehittää järjestelmäsunnittelua. Jos metodologioille on toimiva ja tehokas vertailurunko, voidaan päätellä mitkä elementit mistäkin metodologioista toimivat ja mitkä eivät ole niin tehokkaita. Vertailurungon rakentamisessa tulee aina kuitenkin olemaan subjektiivisyyttä mukana. Hyviä ja toimivia elementtejä verratessa voidaan metodologioista kehittää toimivampia järjestelmäsunnittelun perustavoitteiden toteuttamisessa.

## 4 Negatiivisia tuloksia

Dokumentaation määrä fyysisiin systeemeihin liittyen vaikuttaa olevan rajallista.

Järjestelmäsuunnittelun aihepiiri vaikuttaa ainakin akateemisessa tekstissä keskittyneen hyvin vahvasti ohjelmistosuunnitteluun. Mainintoja fyysisten järjestelmien suunnittelusta löytyy esimerkiksi 1950- ja 60-luvulta, mutta järjestelmäsuunnittelu oli silloin hyvin alkuvaiheissaan joten prosesseja ei kuvailta kovin tarkasti. On myös mahdollista, että prosessit näin alkuvaiheessa eivät noudattaneet mitään standardeja, vaan olivat pelkästään tiettyjen yritysten sisäisiä prosesseja, joista ei tehty akateemista tai julkista dokumentaatiota. Dokumentaatiota alkaa ilmestymään vasta kun puhutaan ohjelmistosuunnittelusta. Mallikieliin ja muihin työkaluihin liittyvä dokumentaatio usein olettaa, että lukija on ohjelmistosuunnittelija tai että ohjelmistosuunnittelu on selkeästi järjestelmäsuunnittelun päämääräinen käyttötarkoitus. Oletettavasti järjestelmäsuunnittelua käytettiin alun perin fyysisiin systeemeihin, mutta näiden suunnittelumetodien toiminnasta ei löydy paljoa konkreettista tietoa.

### 4.1 Milloin ei kannata käyttää järjestelmäsuunnittelua?

Järjestelmäsuunnittelu ei kuitenkaan aina ole paras ratkaisu jokaiseen projektiin. Kaksi isoa ongelmaa mitä järjestelmäsuunnittelun käyttämisessä esiintyy ovat projektin alkupäässä oleva suurempi sijoitus ja järjestelmäsuunnitteluun tarvittavien taitojen mahdollinen puute.

Toimivan järjestelmäsuunnittelun sisällyttäminen projektiin vaatii aluksi suurta investointia, sillä siihen vaaditaan osaavia ihmisiä, työkaluja ja aikaa. Mikäli projekti on tarpeeksi lyhyt tai yksinkertainen, niin tämä alussa käytetty sijoitus voi olla liian suuri ollakseen järkevä.

Järjestelmäsuunnittelu ei myöskään tuota suurta hyötyä mikäli projektissa ei ole mukana osaavia järjestelmäsuunnittelijoita. Tämän voisi myös sisällyttää alkupään sijoituskustannukseksi. Nämä tekijät korostuvat entisestään, kun puhutaan mallipohjaisesta järjestelmäsuunnittelusta. Rafael M. Perez huomasi tutkimuksessaan, että mallipohjainen järjestelmäsuunnittelu lisäsi avaruuteen liittyvissä projekteissa luotettavuutta mutta teki niistä painavampia ja monimutkaisempia. <sup>xxxiii</sup>

## 5 Tulevaisuus

Järjestelmäsuunnittelu on ala joka kehittyy jatkuvasti. Se kehittyy sekä siksi jotta saavuttaa tavoitteensa paremmin, sekä siksi että sen täytyy mukautua muuttuvaan maailmaan. International Council on Systems Engineering (INCOSE, suom. Kansainvälinen järjestelmäsuunnittelun kanslia) julkaisee aina välillä tulevaisuuden vision järjestelmäsuunnittelun kehityksestä nimeltä INCOSE Systems Engineering Vision. Viimeisin versio on julkaistu vuonna 2022 ja se käsittelee järjestelmäsuunnittelun nykyisiä haasteita ja sitä miten järjestelmäsuunnittelun tulisi kehittyä tulevaisuudessa. INCOSE:n mukaan esityksen tarkoituksena on inspiroida ja ohjata järjestelmäsuunnittelun kehitystä.<sup>xxxiv</sup> Esite listaa elementtejä järjestelmäsuunnittelun nykytilanteesta ja esittää ehdotuksen siitä mihin suuntaan tulisi kehittyä. Esityksessä puhutaan paljon voitoista ja osakkaille arvon tuomisesta, mikä viittäisi siihen, että esitys ei ole puhtaasti akateeminen katsaus järjestelmäsuunnittelun potentiaaliseen tulevaisuuteen vaan on enemmänkin suunnattu järjestelmäsuunnittelun ratkaisuihin sijoittaville tahoille.

Esitteessä sanotaan, että järjestelmäsuunnittelun kehitystä on ajanut järjestelmien kasvava monimutkaisuus ja tämän monimutkaisuuden hallitsemisen tarve.<sup>xxxv</sup> Esitteessä väitetään, että nykyisen järjestelmäsuunnittelun tärkeimmät haasteet ovat pirstaloitumiseen ja standardisoinnin puutteeseen liittyviä. Useiden erilaisten erikoistuneiden työkalujen yhteensopimattomuus, standardisoinnin puute ja yksityisten dataformaattien käyttö on listattu isoimmiksi haasteiksi, mutta mainitaan että kehitystä on tapahtunut näillä alueilla. Järjestelmäsuunnittelijoiden yhteistyö on esitteessä keskeisessä roolissa, joten on ymmärrettävää miksi tällaista pirstaloitumista pidetään haitallisena. Lisäksi mainitaan tekoälyn rooli ja sen tuomat eettiset ja logistiset haasteet.<sup>xxxvi</sup>

INCOSE ennustaa, että tulevaisuudessa järjestelmäsuunnittelusta tulee entistäkin tärkeämpää ja poikkitieteellisempää. INCOSE:n mukaan tämä kehitys seuraa järjestelmien monimutkaisuuden kasvua.<sup>xxxvii</sup> Esitteessä myös väitetään, että tulevaisuuden järjestelmäsuunnittelu tulee olemaan enemmänkin mallipohjaista eikä tekstipohjaista. Toiveena on, että tulevaisuudessa mallinnusteknologia standardisoituu ja kehittyy tarpeeksi jotta mallipohjaiseen suunnitteluun voidaan siirtyä vahvemmin.<sup>xxxviii</sup> Esitteen ennusteet mallipohjaiselle suunnittelulle ovat optimistisia.

Esitteessä on myös esitetty tavoitteeksi vahvistaa järjestelmäsuunnittelun roolia akatemian maailmassa. Nykytilanteessa tarvitaan enemmän järjestelmäsuunnittelijoita mitä on saatavilla, ja tavoitteena on, että järjestelmäsuunnittelijaksi pyrkivä ihminen saisi useammasta paikasta kattavampaa tukea. Tavoitteena olisi, että tulevaisuudessa on enemmän päteviä järjestelmäsuunnittelijoita, ja järjestelmäsuunnittelijaksi valmistuminen olisi helpompaa, saavutettavampaa ja kattavampaa. <sup>xxxix</sup>

Koska esite on suunnattu mm. potentiaalisille sijoittajille, joten siinä ei puhuta järjestelmäsuunnittelun kehityksestä kovinkaan kriittisesti. Esitteessä on tästä huolimatta hyvin esillä järjestelmäsuunnittelun kehityksen motivaatioita. Alussa on esitetty, miksi järjestelmäsuunnittelua tarvitaan ja mitkä tekijät ovat tuoneet sitä sen nykyiseen tilaan. Esitteen tekijät myös todennäköisesti uskovat esitteen ennustuksiin, vaikka aikatauluja ja tuloksia on saatettu värittää optimistisemmiksi mitä olisi realistisesti mahdollista.

## 6 Yhteenveto

Järjestelmäsuunnittelu on ala, jonka tarkoituksena on tehdä monimutkaisten järjestelmien käsittelemisestä ja suunnittelemisesta helpompaa. Se myös auttaa näiden järjestelmien toiminnan kommunikoimisessa. Alan kehitys on laajalti keskittynyt näiden keskeisten elementtien kehittämiseen. Järjestelmäsuunnittelu on varmasti jonkinlaisessa muodossa ollut olemassa tuhansia vuosia, mutta alan kiteytyminen 1940-luvulla on mahdollistanut sen keskittyneen kehityksen, ja erilaisten työkalujen kuten mallikielien ja metodologioiden olemassaolon. Järjestelmäsuunnittelusta on myös tullut akateemisesti tunnistettu ja alasta on tehty tutkimusta ja sitä voi myös opiskella yliopistoissa. <sup>x1</sup> Järjestelmäsuunnittelu tulee mahdollisesti tulevaisuudessa kehittämään näitä elementtejä vieläkin pidemmälle, ja siitä saattaa tulla vielä universaalimpi. On kuitenkin mahdollista, että kehityksessä tulee vielä ennustamattomia haasteita vastaan.

## 6.1 Lähteet

- i”sh hdbk 2.1.pdf”, viitattu 12. elokuuta 2024, [https://www.rose-hulman.edu/class/cs/csse490-requirements/handouts\\_490/sh%20hdbk%202.1.pdf](https://www.rose-hulman.edu/class/cs/csse490-requirements/handouts_490/sh%20hdbk%202.1.pdf).
- iiKevin A. Reilley ym., ”Methodologies for Modeling and Simulation in Model-Based Systems Engineering Tools”, teoksessa *AIAA SPACE 2016* (AIAA SPACE 2016, Long Beach, California: American Institute of Aeronautics and Astronautics, 2016), <https://doi.org/10.2514/6.2016-5469>.
- iii”SEH 2.0 Fundamentals of Systems Engineering - NASA”, viitattu 15. toukokuuta 2024, <https://www.nasa.gov/reference/2-0-fundamentals-of-systems-engineering/>.
- iv”Thermodynamic System | Definition, Types & Examples”, study.com, viitattu 15. toukokuuta 2024, <https://study.com/academy/lesson/thermodynamic-system-definition-types-examples.html>.
- v”Ecosystem”, viitattu 15. toukokuuta 2024, <https://education.nationalgeographic.org/resource/ecosystem>.
- viReilley ym., ”Methodologies for Modeling and Simulation in Model-Based Systems Engineering Tools”.
- vii”An Introduction to Model-Based Systems Engineering (MBSE)”, 21. joulukuuta 2020, <https://insights.sei.cmu.edu/blog/introduction-model-based-systems-engineering-mbse/>.
- viii”Model, n. & Adj. Meanings, Etymology and More | Oxford English Dictionary”, viitattu 15. toukokuuta 2024, [https://www.oed.com/dictionary/model\\_n](https://www.oed.com/dictionary/model_n).
- ixRobert Halligan, ”Benefits of Model-Based Systems Engineering”, PPI, 8. toukokuuta 2019, <https://www.ppi-int.com/articles-systems-engineering/benefits-of-model-based-systems-engineering-2/>.
- xMorayo Adedjouma ym., ”From Document-Based to Model-Based System and Software Engineering”, 2016.
- xiSo Young Kim, David Wagner, ja Alejandro Jimenez, ”Challenges in Applying Model-Based Systems Engineering: Human-Centered Design Perspective” (Root, 9. toukokuuta 2024), <http://dataverse.jpl.nasa.gov/dataset.xhtml?persistentId=hdl:2014/51368>.
- xiiEdward Ralph Carroll ja Robert Joseph Malins, ”Systematic Literature Review: How Is Model-Based Systems Engineering Justified?” (Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); Sandia National Lab. (SNL-CA), Livermore, CA (United States), 1. maaliskuuta 2016), <https://doi.org/10.2172/1561164>.
- xiii”Lerat - 2010 - 5.5.2 Three Reasons why Document-based SE (usually.pdf”, viitattu 14. elokuuta 2024, [https://web.mst.edu/lib-circ/files/Special%20Collections/INCOSE2010/Three%20Reasons%20why%20Document%20based%20SE%20\(usually\)%20works%20better%20than%20\(most%20of\)%20MBSE.pdf](https://web.mst.edu/lib-circ/files/Special%20Collections/INCOSE2010/Three%20Reasons%20why%20Document%20based%20SE%20(usually)%20works%20better%20than%20(most%20of)%20MBSE.pdf).
- xivCarroll ja Malins.
- xvJose J Padilla, Saikou Y Diallo, ja Andreas Tolk, ”Do We Need M&S Science?”, 2011.
- xviLouis G. Birta ja Gilbert Arbez, *Modelling and Simulation: Exploring Dynamic System Behaviour*, Simulation Foundations, Methods and Applications (Cham: Springer International Publishing, 2019), <https://doi.org/10.1007/978-3-030-18869-6>.
- xvii”What is MBSE and why do industries start to use? - Model Based Systems Engineering (MBSE) on AWS: From Migration to Innovation”, viitattu 15. toukokuuta 2024, <https://docs.aws.amazon.com/whitepapers/latest/model-based-systems-engineering/what-is-mbse-and-why-do-industries-start-to-use.html>.
- xviiiA. Wayne Wymore, *Model-Based Systems Engineering* (CRC Press, 2018).
- xixJames Towers, ”Model Based Systems Engineering – The State of the Nation”, 2013.
- xxGrady Booch ym., ”Object-Oriented Analysis and Design with Applications, Third Edition”, *ACM SIGSOFT Software Engineering Notes* 33, nro 5 (31. elokuuta 2008): 29–29, <https://doi.org/10.1145/1402521.1413138>.
- xxiCecilia Haskins, ”4.6.1 A Historical Perspective of MBSE with a View to the Future”, *INCOSE International Symposium* 21, nro 1 (kesäkuuta 2011): 493–509, <https://doi.org/10.1002/j.2334-5837.2011.tb01220.x>.
- xxiiGrady Booch, James Rumbaugh, ja Ivar Jacobson, *The Unified Modeling Language User Guide*, The Addison-Wesley Object Technology Series (Reading (Mass.): Addison-Wesley, 1999).
- xxiiiRaman Ramsin ja Richard F. Paige, ”Process-Centered Review of Object Oriented Software Development Methodologies”, *ACM Computing Surveys* 40, nro 1 (helmikuuta 2008): 1–89, <https://doi.org/10.1145/1322432.1322435>.
- xxivBooch, Rumbaugh, ja Jacobson, *The Unified Modeling Language User Guide*.
- xxvBooch, Rumbaugh, ja Jacobson.
- xxvi”Ivar Jacobson on UML, MDA, and the Future of Methodologies”, InfoQ, viitattu 15. toukokuuta 2024, <https://www.infoq.com/>.
- xxviiCarroll ja Malins, ”Systematic Literature Review”.
- xxviii”SysML FAQ: What Is the Relation between SysML and UML?”, SysML.org, viitattu 15. toukokuuta 2024, <https://sysml.org/sysml-faq/sysml-faq/what-is-relation-between-sysml-and-uml.html>.
- xxixJohn Erickson ja Keng Siau, ”Theoretical and Practical Complexity of UML”, *New York*, 2004.
- xxxJeff A Estefan, ”Survey of Model-Based Systems Engineering (MBSE) Methodologies”, 2008.
- xxxiT. Weilkiens ym., ”Evaluating and comparing MBSE methodologies for practitioners”, teoksessa *2016 IEEE International Symposium on Systems Engineering (ISSE)* (2016 IEEE International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom: IEEE, 2016), 1–8, <https://doi.org/10.1109/SysEng.2016.7753174>.
- xxxiiChristophe Ponsard ja Valery Ramon, ”Applying and Extending FEMMP to Select an Adequate MBSE Methodology”, teoksessa *Proceedings of the 17th International Conference on Software Technologies* (17th

## 6.1 Lähteet

- International Conference on Software Technologies, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2022), 508–15, <https://doi.org/10.5220/0011312200003266>.
- xxxiii Rafael M. Perez, "Application of MBSE to Risk-Informed Design Methods for Space Mission Applications", teoksessa *AIAA SPACE 2014 Conference and Exposition* (AIAA SPACE 2014 Conference and Exposition, San Diego, CA: American Institute of Aeronautics and Astronautics, 2014), <https://doi.org/10.2514/6.2014-4411>.
- xxxiv "Objectives & Stakeholders", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/objectives-stakeholders>.
- xxxv "Historical Perspectives", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/historical-perspectives>.
- xxxvi "Challenges", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/challenges>.
- xxxvii "The Global Context for Systems Engineering", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/chapter-3-the-future-state-of-systems-engineering>.
- xxxviii "Model-Based Practices", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/model-based-practices>.
- xxxix "Building the Systems Engineering Workforce of the Future", *Systems Engineering Vision 2035*, viitattu 15. toukokuuta 2024, <https://violin-strawberry-9kms.squarespace.com/building-the-systems-engineering-workforce-of-the-future>.
- xl "Master of Systems Engineering Degree | Embry-Riddle Aeronautical University", viitattu 15. toukokuuta 2024, <https://erau.edu/degrees/master/systems-engineering>.