

Anna Pertola

**KIELITEOREETTISIA NÄKÖKULMIA
TIETOJÄRJESTELMIEN MALLINTAMISEEN**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2024

TIIVISTELMÄ

Pertola, Anna

Kieliteoreettisia näkökulmia tietojärjestelmien mallintamiseen

Jyväskylä: Jyväskylän yliopisto, 2024, 82 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja(t): Halttunen, Veikko

Tässä pro gradu -tutkielmassa käsitellään tietojärjestelmien mallintamisessa hyödynnettäviä kieliteoreettisia näkökulmia. Aiheen käsittely on jaettu kahteen osaan: aluksi kirjallisuuskatsauksessa käydään läpi mallinnuksen lingvistisistä lähtökohdista kirjoitettuja artikkeleja, ja lopuksi asiantuntijahaastatteluissa pyritään luomaan käytännön näkemyksiä. Tarkoituksena on selvittää seuraavia tutkimuskysymyksiä: Miten kieliteoreettisia näkökulmia on hyödynnetty tietojärjestelmien suunnittelussa sekä teoriassa että käytännössä? Mitä hyötyä kielitieteestä on ollut tietojärjestelmätieteelle? Voisiko joistakin vielä hyödyntämättömistä lingvistisistä teorioista olla hyötyä tietojärjestelmien suunnittelussa, vai kannattaako lingvistisistä lähtökohdista luopua ja keskittyä johonkin muuhun lähestymistapaan? Tutkimuskysymyksiin pyritään vastaamaan etenkin teorian osalta kirjallisuuskatsauksella. Siinä pohjana on käytetty Kalle Lyytisen vuonna 1985 julkaistussa artikkelissa *Implications of Theories of Language for Information Systems* mainitsemia näkökulmia. Artikkelista on kulunut jo aikaa, joten aiheisiin syvennyttiin myös etsimällä muita aiheesta vuosien varrella tehtyjä tutkimuksia. Samalla tietojärjestelmätieteeseen yhdistyi lingvistiikkaa, filosofiaa ja psykologiaa. Lisäksi tutkielmassa kiinnitettiin huomio siihen, että viime aikoina ketterä kehitys on voinut viedä pohjaa mallintamiselta. Uusimman tekoälyn kehityksen myötä kuitenkin kielimallit ovat nousseet pintaan. Näin ollen myös tätä näkökulmaa on käyty läpi. Empiiriseen osioon valikoitiin kuusi haastateltavaa, joilla on käytännön taustaa mallintamisen ja sen tutkimuksen parista. Heitä haastateltiin touko- ja kesäkuussa 2020 etäyhteyksien välityksellä. Haastatteluista todettiin, että artikkeleissa esitetyihin menetelmiin saatettiin suhtautua positiivisesti, mutta varauksella. Myönteinenkin suhtautuminen kieliteorioiden mahdollisuuksiin ei välttämättä näkynyt käytännön toteutuksen tasolla, ainakaan kirjallisuuden ehdottamassa laajuudessa. Aina mallintaminen ei automaattisesti kannata, ja kannattavuuden kartoittamista ehdotettiin jatkotutkimuksen aiheeksi. Muina aiheina ehdotettiin alan ja tekoälyn tulevaisuuden kehityksen seurantaan, haastattelujen laajentamista sekä keskittymistä tutkimuksen ja käytännön parempaan yhteensovittamiseen.

Asiasanat: mallintaminen, tietojärjestelmien kehittäminen, kielitiede

ABSTRACT

Pertola, Anna

Linguistic perspectives on information systems modelling

Jyväskylä: University of Jyväskylä, 2024, 82 pp.

Information Systems, Master's Thesis

Supervisor: Halttunen, Veikko

This Master's Thesis brings forward linguistic perspectives used in information systems modelling. The topic is divided in two sections: the literature review covering articles on linguistic perspectives, and the empirical section with expert interviews and discussion. This thesis studies the following research questions: How have linguistic perspectives been utilised in information systems modelling, in theory as well as in practice? What have been the benefits of linguistics for IS science? Could some yet unutilised linguistic theories be useful for information systems design, or should the linguistic perspectives be put aside to concentrate on other approaches instead? The theoretical aspects of these research questions are answered via literature review. The foundations for the linguistic perspectives had been established in an article published in 1985 by Kalle Lyytinen, *Implications of Theories of Language for Information Systems*. This article was written decades ago, and therefore more recent studies covering these subjects have been explored as well. Consequently, information systems science has been complemented by ideas from linguistics, philosophy and psychology. In addition, this thesis points out that the increase of agile development may have reduced the use of system modelling. However, recent developments in artificial intelligent and large language models may have an effect on modelling and linguistics. Therefore, this perspective is also covered. The empirical section consists of interviews carried out with six experts who had practical experience from modelling and/or modelling research. They were interviewed in May and June 2020 on video meeting platforms. The interviews indicated that the theories suggested by the articles have had positive but somewhat reserved reception. Even if attitudes towards opportunities offered by linguistic theories were generally positive, they were not necessary used in the practical work, at least not in the extent put forward by the literature. Modelling is not always considered worthwhile, and topics for further research included studying in which situations and what kind of modelling would be worth the extra effort. Other avenues for future research were following the development of the field and the effects of AI, expanding/increasing the interviews and concentrating on better alignment of research and practice.

Keywords: modelling, information systems development, linguistics

KUVIOT

KUVIO 1. Mallit tietojärjestelmien kehittämisessä (Wandin ym., 2005, mukaan)	18
KUVIO 2 Yksinkertaistettu esimerkki UML-käyttötapauskaaviosta (Giaglisia, 2001, mukaellen)	26

TAULUKOT

TAULUKKO 1 Mallien eri tasot suhteessa toisiinsa. (Neton & Neton, 2011, mukaan)	28
TAULUKKO 2 Puheaktien lajittelu, Abbasin ym. (2018) mukaan	36
TAULUKKO 3 Kieliteoreettisten näkökulmien peruspiirteet. Byrerin ja Jelassin (1989) mukaan.	45
TAULUKKO 4 Kieliteoreettisten näkemysten yhteenvetoa Lyytisen (1985) mukaan.....	46
Taulukko 5. Yhteenveto haastattelujen toteutuksesta.....	53
TAULUKKO 6 Haastattelujen yhteenveto 1/2	60
TAULUKKO 7 Haastattelujen yhteenveto 2/2	62

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO.....	7
1.1 Taustaa	8
1.2 Aiempi tutkimus	10
1.3 Tutkimuskysymykset ja -menetelmät.....	11
1.4 Tutkielman rakenne	13
2 YLEISTÄ MALLINTAMISESTA.....	14
2.1 Tärkeimmät käsitteet.....	14
2.2 Mitä mallintamisella tarkoitetaan	17
2.3 Kielet mallintamisen välineenä.....	19
2.4 Mallintaminen ja ketterä kehitys.....	21
3 MALLINNUSKIELTEN KEHITTÄMINEN JA KEHITTYMINEN.....	23
3.1 Mallinnuskielten historia.....	23
3.2 UML.....	25
3.3 Metamallintamisen merkitys	27
3.4 Sovellusaluekohtaisten kielten kehitys.....	29
3.5 Tekoäly ja kielimallit	31
4 KIELITEOREETTINEN TARKASTELU	34
4.1 Puheaktiteoria	35
4.2 Language Action Perspective	37
4.3 Frege	38
4.4 Skinner.....	39
4.5 Chomsky	40
4.6 Piaget	42
4.7 Muita mahdollisia näkökulmia.....	43
4.8 Kirjallisuuskatsauksen yhteenveto.....	44
5 EMPIIRISEN TUTKIMUKSEN LÄHTÖKOHDAT JA TOTEUTUS	48
5.1 Haastattelujen tavoitteet	48

5.2	Teemahaastattelu	49
5.3	Haastateltavat.....	49
5.4	Haastattelukysymykset.....	51
5.5	Toteutus.....	52
6	TUTKIMUSTULOKSET	54
6.1	Mallintamisen kannattavuus	54
6.2	Teorian ja käytännön ristiriita.....	56
6.3	Kielitieteen näkökulma	57
6.4	Ajallinen muutos.....	58
6.5	Haastattelujen yhteenveto	60
7	POHDINTA	65
7.1	Johtopäätökset.....	65
7.2	Tulosten hyödynnettävyys ja luotettavuus.....	68
7.3	Jatkotutkimus	70
8	YHTEENVETO	72
	LÄHTEET	74

1 JOHDANTO

Tietojärjestelmä on pohjimmiltaan lingvistisen viestinnän väline, ja tietojärjestelmien kehittäminen on sosiaalista vuorovaikutusta organisaatiossa (Lyytinen, 1987). Se on semanttinen prosessi, jolla pyritään kuromaan umpeen sidosryhmien välistä viestintäkuilua (Charaf, Rosenkranz & Holten, 2013).

Tietojärjestelmien suunnittelussa ja toteutuksessa käytetään kieltä. Niin tietojärjestelmät kuin kielijärjestelmätkin ovat sosiaalisen vuorovaikutuksen välineitä, ja niillä pyritään hahmottamaan ja kuvaamaan ulkoista maailmaa yhteisesti sovittujen symbolien avulla. On siis perusteltua ajatella, että kieliteoreettisista näkökulmista voisi olla hyötyä tietojärjestelmien kehittämisessä. Näin voitaisiin pystyä parantamaan vuorovaikutusta ja sen myötä myös tietojärjestelmäprojektien onnistumista.

Tässä pro gradu -tutkielmassa kartoitetaan, minkälaisia lingvistisiä näkökulmia on hyödynnetty tai ehdotettu hyödynnettäviksi tietojärjestelmien mallintamista koskevassa kirjallisuudessa. Tarkoitus on tarkastella, miltä osin kielitieteen näkökulmista voi olla hyötyä tietojärjestelmätieteen puolella ja miten nämä alat voivat tukea toisiaan.

Empiirisessä osiossa pyritään selvittämään haastattelujen avulla, onko teoria toteutunut, onko kieliteorioista ollut hyötyä ja onko niitä käytetty käytännössä. Haastatteluihin valittiin kuusi asiantuntijaa tietojärjestelmien mallintamiseen liittyvän kokemustaustan perusteella. Osiossa luodaan myös katsaus ajalliseen muutokseen kieliteorioiden ja mallintamisen osalta ja pohditaan, minkälaisia ovat kieliteoreettisten näkökulmien ja tietojärjestelmien mallintamisen tulevaisuudennäkymät.

Tutkielma jakautuu siis kahteen osaan, kieliteorioihin liittyvään kirjallisuuskatsaukseen sekä teemahaastatteluihin. Aluksi aihetta taustoitetaan käsittelemällä lyhyesti mallintamista, metamallinnusta ja mallinnuskieliä yleisesti. Lisäksi luodaan katsaus tekoälyyn ja kielimalleihin sekä pohditaan, millaisia vaikutuksia niillä mahdollisesti on tietojärjestelmien mallintamiseen.

1.1 Taustaa

Tietojärjestelmäprojektit epäonnistuvat usein. IT-projektien onnistumista seuraava Standish Group luokittelee CHAOS-raportissaan projektit onnistuneisiin, haasteellisiin tai epäonnistuneisiin (Alves, Souza, Ribeiro & Machado, 2021). Haasteilla tarkoitetaan, että projekti toimitetaan myöhässä, budjetti ylittyy tai osa toivotuista ominaisuuksista puuttuu. Osa projekteista puolestaan keskeytyy tai perutaan kokonaan, koska niillä voi olla haitallisia vaikutuksia organisaatioille. (Yeo, 2002.)

Vuoden 2020 CHAOS-raportin mukaan projekteista 31 % onnistui, 50 %:lla oli haasteita ja 19 % epäonnistui (Varajão, Trigo, Pereira & Moura, 2022). Analyysien perusteella on myös arvioitu, että kaksi kolmesta projektista epäonnistuu (Nelson, 2007) tai vain yksi kahdeksasta onnistuu, mikäli onnistumista arvioidaan jonkin ns. projektinhallintakolmion tavoitteista – aika, kustannukset ja laatu – saavuttamisen perusteella (McManus & Wood-Harper, 2008). Projektien epäonnistumisesta voi seurata kustannusten lisäksi tuottojen ja työpaikkojen menetystä (Nelson, 2007).

Mistä projektien ongelmat sitten johtuvat? Syiksi on todettu esimerkiksi huono vaatimusten määrittely, ja ratkaisuksi on ehdotettu viestintäsuunnitelman laadintaa ja parhaiden käytäntöjen noudattamista (Nelson, 2007). Nelson (2007) myös arvioi, että liiallinen luottaminen yhteen tiettyyn uuteen ratkaisuun tai sen tuomien säästöjen yliarvioiminen aiheuttaa usein epäonnistumisia. McManus ja Wood-Harper (2008) listaavat lukuisia projektin peruuntumiseen johtaneita syitä, joihin lukeutuvat vaikeudet liiketoimintaprosessien yhteensovittamisessa, määritysten hallinnan ongelmat sekä viestinnän ongelmat. Standish Groupin raporteissa ongelmien syiksi on myös todettu kommunikaation puutteet käyttäjien ja järjestelmän analyysoijien välillä sekä puutteellinen käyttäjien osallistuminen (Jaakkola, Henno, Welzer-Druzovec, Thalheim ja Mäkelä, 2016).

Loppukäyttäjien ja projektin parissa työskentelevien väliset ongelmat helposti kertautuvat myös muissa osa-alueissa: esimerkiksi kommunikaation ongelmat muutospyyntötilanteissa tai vaatimusten määrittelyssä aiheuttavat myös liiallista kustannusten paisumista. Sen McManus ja Wood-Harper (2008) toteavat yhdeksi suurimmista projektien peruuntumisen syistä.

Puutteellisista järjestelmän määrityksistä aiheutuvat ongelmat havaitsi jo Bostrom vuonna 1989 julkaistussa artikkelissaan. Hänen mukaansa kehittäjien ja käyttäjien välinen tehokas viestintä olisi erityisen tärkeää, jotta järjestelmistä saadaan toimivia: jos määritykset ovat pielessä, ei järjestelmäkään voi olla onnistunut (Bostrom, 1989). Artikkelista on jo kulunut aikaa, mutta tehokkaan viestinnän tarve on kaikkea muuta kuin poistunut. Bostromin (1989) mainitsemat ongelmatkin ovat ennallaan: käyttäjät puhuvat toimialakohtaisilla sanoilla, kehittäjät teknisillä.

Saman ongelman ovat todenneet Liebel, Tichy, Knauss, Ljungkrantz ja Stieglbauer (2018) tuoremassa autoteollisuuden vaatimusten määrittelyä kä-

sitelleessä tutkimuksessaan. Sen perusteella projekteissa havaituista ongelmista monet, esimerkiksi tiedon puutteet ja epäselvyydet, olisivat ratkaistavissa viestinnän ja palautteen keinoin. Tutkimuksessa myös arveltiin toimintojen ja vaatimusten kuvaamisen malleilla (esim. UML, jota käsitellään luvussa 3.2) voivan parantaa ymmärrystä eri alojen toimijoiden kesken välillä. Vastaavasti Fernándezen, Wagnerin ja Kalinowskin (2017) mukaan kommunikaation puutteet ovat yksi suurimmista ongelmista vaatimusten määrittelyssä, joka puolestaan on tärkeää ohjelmistoprojektin onnistumisen kannalta. Samoin Jaakkolan ym. (2016) mukaan vaatimustenmäärittelyn virheet kertautuvat jokaisessa seuraavassa vaiheessa ja lisäävät samalla kustannuksia kolminkertaisesti verrattuna siihen, että ongelmat korjattaisiin jo vaatimusten analysoinnissa.

McManusin ja Wood-Harperin (2008) mukaan toisinaan syynä ovat ongelmat kommunikaatiossa sidosryhmien kanssa, ja toimiva kommunikaatio olisi olennaista myös palautteen antamisen kannalta. Sen avulla voidaan myös onnistua voittamaan käyttäjien vastustus, jota Yeo (2002) pitää myös projektin epäonnistumisen merkinä, vaikka projektin tavoitteet muutoin olisikin saavutettu. Paremmalla viestinnällä voidaan lisätä organisaation jäsenten luottamusta muutosta kohtaan ja siten parantaa muutoksen läpimenon todennäköisyyttä (Clarke, 1999).

Etenkin ketterässä kehityksessä kommunikaation merkitys korostuu. (Luong, Sivarajah & Weerakkody, 2019). Tunneällyn on todettu korreloivan ihmisiin liittyvien haasteiden kanssa ahdistuksen, motivaation, luottamuksen ja viestintätaitojen osalta. Työntekijöiden tunneäly- ja kommunikointikyvyt tulisi-kin heidän mukaansa huomioida rekrytoinnissa, jotta voidaan parantaa ketterien projektien onnistumisen mahdollisuutta. (Luong ym., 2021.)

Voidaankin todeta, että kommunikaatio olisi monessa mielessä tietojärjestelmäprojektien onnistumisen kulmakivi. Siksi on ollut loogista, että pyrittäessä löytämään ratkaisuja järjestelmien kehittämisen ja mallintamisen parantamiseen on käännytty myös kielitieteellisten ratkaisujen puoleen. Lingvistiikan tarjoamiin mahdollisuuksiin alettiin kiinnittää huomiota etenkin 1980-luvulla LAP-näkökulman myötä. Aiempia tutkimuksia käsitellään tarkemmin seuraavassa aliluvussa, ja kieliteoreettisiin ratkaisuihin perehdytään luvussa 4.

Aluksi mallintamisessa oli käytössä useita kilpailevia kieliä. Käytäntöjen yhtenäistämiseksi kehitettiin UML (Unified Modeling Language) 1990-luvulla (Bézivin & Muller, 1998). Samoihin aikoihin laadittiin kuitenkin myös ketterän kehityksen manifesti (Agile Manifesto), jonka myötä mallintaminen alettiin nähdä monimutkaisena ja hankalana (Liddle, 2011). Mallintamista tukeva teoria ja sitä kaihtava käytäntö erkaantuivat toisistaan. Tämä ilmiö on nähtävissä myös haastatteluissa, joita käsitellään osiossa 6.

IT-alalla teknologia kuitenkin kehittyy ja muuttuu nopeasti, ja edistysaskeleet vaikuttavat väistämättä myös tietojärjestelmien kehittämiseen. Tämän tutkielman kirjoittamisen aikana kielimalleja hyödyntävän tekoälyn toiminta on kehittynyt suuresti, ja muun muassa luonnollisella kielellä ”keskusteleva” tekoälypalvelu ChatGPT on julkaistu (Ooi ym., 2023). Onkin mahdollista, että kieli- malleja hyödyntävä tekoäly nostaa lingvistiset lähtökohdat uuteen arvoon tieto-

järjestelmien kehittämisessä tai auttaa löytämään uusia ratkaisuja siitäkin näkökulmasta.

Tässä pro gradu -tutkielmassa pyritään tutustumaan kielitieteellisten lähtökohtien hyödyntämiseen tietojärjestelmien kehittämisessä sekä kirjallisuuskatsauksen että haastattelujen avulla. Tarkoitus on tarkastella, miten lingvististen lähtökohtien on teoriassa ajateltu voivan hyödyntää tietojärjestelmien kehittämistä ja mitä hyötyä niistä on ollut käytännössä. Lisäksi tarkastellaan, miten tilanne mahdollisesti on muuttunut ajan myötä ja minkälaista kehitystä tulevaisuudessa on odotettavissa; nostetaan esiin menneisyyden unohtuneitakin näkökulmia ja tarkastella niitä nykypäivän vaatimusten valossa.

Tietojärjestelmätieteen tutkimus on monialaista, sillä siinä nivoutuvat toisiinsa johtamistieteet ja tekniikka, psykologia ja sosiologia (Yeo, 2002). Näin ollen tässäkin tutkielmassa on hyödynnetty monen alan tutkimuksia. Lopputuloksen toivotaan hyödyntävän niin tietojärjestelmätiedettä itsessään kuin muita siihen liittyviä alojakin, lisäävän lingvististen aiheiden huomiointia tietojärjestelmätieteen tutkimuksessa sekä täydentävän käytännön ja tutkimuksen välistä kuilua.

1.2 Aiempi tutkimus

Tässä aliluvussa luodaan lyhyesti katsaus aiempiin aiheesta tehtyihin tutkimuksiin. Kyseisiä tutkimuksia käsitellään kuitenkin tarkemmin myöhemmissä mallintamista ja kieliteorioita koskevissa luvuissa.

Tietojärjestelmien mallintamisesta yleisesti ja sen historiasta on kirjoittanut esimerkiksi Schichl (2004). UML-kieltä ovat käsitelleet Sommerville (2011), Lange ja Chauldron (2006) sekä Fowler ja Scott (2000). Metamallintamisesta ovat kirjoittaneet Kühne (2006) ja sovellusaluekohtaisesta mallintamisesta Liddle (2011). Näitä aiheita tarkastellaan luvussa 3.

Tietojärjestelmien mallintamisessa käytetään kieltä, mutta Lyytisen (2004) mukaan lingvististä tarkastelua ei ole nähty osana alan peruskäytäntöjä eikä sitä ole otettu tutkimuksen keskipisteeksi. Hän on kuitenkin itse ollut edistämässä lingvistispainotteista tutkimusta, ja alalle ovat tuoneet oman panoksensa muun muassa Auramäki, Hirscheim ja Winograd.

Kielitieteellisessä näkökulmassa hyödynnetään usein puheaktiteoriaa, joka perustuu J. R. Searlen (1969) ja J. L. Austinin (1966) ajatuksiin. Puheaktia ovat tutkineet tietojärjestelmätieteen alalla esimerkiksi Janson ja Woo (1996, 1995), ja se on toiminut myös Auramäen, Lehtisen ja Lyytisen (1988) sekä Auramäen, Hirscheimin ja Lyytisen (1992) kehittämän SAMPO-lähestymistavan taustalla. Tässä teoriassa tietojärjestelmiä pidetään kommunikatiivisena toimintana, ja puheaktiteoriaa hyödynnetään toimiston tietojärjestelmien kehittämisessä (Auramäki ym., 1988).

Puheaktiteoriasta on kehitetty myös Winogradin (1980) ja Floresin ja Ludlow'n (1976) tutkimuksiin perustuva Language Action Perspective (LAP). Viime aikoina tämä lähtökohta on ollut esillä esimerkiksi Abbasin, Zhoun, Dengin

ja Zhangin (2018) sosiaalista mediaa käsittelevässä tutkimuksessa, ja sen soveltamisen puolesta analyysin välineenä ovat puhuneet myös Bider ja Perjons (2012).

Lyytisen artikkelissaan *Implications of Theories of Language for Information Systems* (1985) esiin tuomia kielinäkökulmia ovat käsitelleet lisäksi Byrer ja Jellasi (1989) sekä Basden ja Klein (2008), jotka keskittyvät kielifilosofiseen lähestymistapaan ja käsittelevät lisäksi Dooyeweerdin ja Habermasin ajatuksia. Stamper, Liu, Hafkamp ja Ades (2000) puolestaan tutkivat semanttisten mallien laatimista organisaatioista ja esittävät, että tietojärjestelmä on itsessäänkin sosiaalisista normeista koostuva organisaatio. Semantiikkaan tietojärjestelmissä keskittyvät artikkelissaan myös Mingers ja Willcocks (2014). Filosofialla järjestelmänkehittämiseen ovat yhdistäneet Hanseth ja Monteiro (1994) ja lingvistisen käänteiden vaikutusta tietojärjestelmäteoriaan on pohtinut Dreiling (2006).

Artikkeleissa mainittuja Fregen, Chomskyn, Skinnerin ja Piaget'n näkökulmia käsitellään tarkemmin luvussa 4. Näistä teorioista on kirjoitettu runsaasti etenkin filosofian, logiikan, kielitieteen ja psykologian tieteenaloilla. Tässä tutkimuksessa on kuitenkin keskitytty tietojärjestelmätiedettä ja IT-alaa sivuvaan sovellutuksiin.

Lingvivististä lähtökohdista ja metamallintamisesta on siis kirjoitettu useita artikkeleja, ja erilaisia lähestymistapoja on kehitetty ja tutkittu. Monista julkaisuista on kuitenkin jo kulunut vuosia, joiden aikana teknologia ja tietojärjestelmät ovat kehittyneet. Onko kieliteoreettisten näkökulmien aika jo ohi vai onko niillä vielä annettavaa nykyaikana? Minkälainen tulevaisuus ylipäättään mallintamisella on odotettavissa? Entä ovatko teoria ja käytäntö samoilla linjoilla mallintamisesta vai ovatko ne erkaantuneet toisistaan? Tätä pyritään selvittämään empiirisessä osiossa valikoitujen asiantuntijoiden haastatteluilla.

1.3 Tutkimuskysymykset ja -menetelmät

Tässä pro gradu -tutkielmassa esitetään seuraavat tutkimuskysymykset:

Miten kieliteoreettisia näkökulmia on hyödynnetty tietojärjestelmien suunnittelussa sekä teoriassa että käytännössä?

Mitä hyötyä kielitieteestä on ollut tietojärjestelmätieteelle?

Voisiko joistakin vielä hyödyntämättömistä lingvivistisiä teorioista olla hyötyä tietojärjestelmien suunnittelussa, vai kannattaako lingvivististä lähtökohdista luopua ja keskittyä johonkin muuhun lähestymistapaan?

Tutkimuskysymyksiin vastataan kirjallisuuskatsauksen sekä asiantuntija-haastatteluiden ja niistä esiin nousevien seikkojen avulla. Kirjallisuudessa on pyritty tutustumaan mallintamiseen yleisesti, mallinnuskielten historiaan sekä metamallintamisen merkitykseen. Soveltuvia tekoälyyn ja kielimalleihin liittyviä artikkeleita on tarkasteltu. Lisäksi kieliteorioita ja niiden yhdistämistä tietojärjestelmien mallintamiseen käsittelevää kirjallisuutta on käyty läpi.

Artikkelit on haettu Google Scholarista mallintamista, metamallinnusta ja kieliteorioita koskevien sekä muiden tutkimuksen kannalta olennaisten haku-

sanojen avulla. Haussa on pyritty etsimään artikkeleja monipuolisesti eri vuosilta, viime vuosien tutkimukset mukaan lukien. Tarkoitus on kuitenkin kehittää kattava kirjallisuuskatsaus myös ajallisesti, joten myöskään vanhimpia julkaisuja ei ole karsittu pois. Lisäksi kieliteoriaa koskeva tutkimus tietojärjestelmätieteen parissa on joiltakin osin vähentynyt viime vuosina. Tosin esimerkiksi aiheesta Language Action Perspective löytyy Google Scholarista 882 osumaa ajalta 1980–2000 mutta vuodesta 2001 lukien 2 720. Vastapainoksi hakuvuosia on välillä rajattu uusimpiin, jotta voitaisiin hahmottaa alan uusimmat kehitysaskelet.

Hakujen yhteydessä on pyritty keskittymään luotettaviin alan julkaisuihin. Lisäksi lähteinä on käytetty muissa artikkeleissa esiin tulleita, hyödyllisiltä vaikuttaneita viitteitä. Tietoa kielimallien uusimmasta kehityksestä on haettu myös suuria kielimalleja (*large language models*, LMM) koskevilla hakusanoilla. Kielimalleja on sovellettu monella alalla, mutta tätä tutkielmaa varten keskityttiin uusimpiin julkaisuihin, jotka ovat joko yleisluonteisia tai eniten tietojärjestelmiin liittyviä ja siltä osin kuin aihe voi vaikuttaa lingvististen lähtökohtien hyödyntämiseen tietojärjestelmien mallintamisessa.

Kieliteorioiden osalta pyrittiin myös valikoimaan tietojärjestelmiin liittyviä lähteitä, vaikka aiheista on julkaistu muidenkin alojen artikkeleja. Lisäksi aiheita on pohjustettu hakemalla tietoa vertaisarvioituista ja luotettaviksi katsotuista tietolähteistä *Stanford Encyclopedia of Philosophy* ja *Internet Encyclopedia of Philosophy*, sillä tutkimusartikkelien laajuus ei ollut aina tarkoituksenmukainen tämän tutkielman mittakaavassa.

Vastaavasti kielitieteen peruskäsitteisiin on tutustuttu artikkelien lisäksi alan perusteoksesta *Johdatus kielitieteeseen* (Häkkinen, 1995). Näin saadun taustatiedon avulla voidaan kartoittaa kielitieteen ja tietojenkäsittelyn yhtäläisyyksiä ja hahmottaa, mitä kielitieteellisiä lähtökohtia on käytetty tähän mennessä niin tutkimuksessa kuin käytännössäkin. Vastaavasti filosofista taustaa tarjosi Panu Raatikaisen toimittama teos *Ajattelu, kieli, merkitys: Analyyttisen filosofian avainkirjoituksia* (1997). Taustoitusta varten tutkittiin myös projektien epäonnistumista koskevia kirjoituksia, mallintamista, sen historiaa ja metamallintamista.

Haastattelujen kohteiksi valittiin pohjana olleen artikkelin laatinut tutkija Kalle Lyytinen, SAMPO-metodologiaa kehittämässä ollut Auramäki sekä neljä muuta tietojärjestelmien mallinnuksen asiantuntijoita. Tarkoituksena oli kerätä avartavaa lisätietoa ja -näkemystä kirjallisuuskatsauksen tueksi. Haastattelut järjestettiin touko–kesäkuussa 2020, vallinneen koronavirustilanteen ja maantieteellisten etäisyyksien vuoksi etäyhteyssovellusten kautta. Haastattelut taltioitiin ja ne litteroitiin tutkimusta varten.

Työssä laaditaan ensin yleiskatsaus tietojärjestelmien mallintamisessa käytettyihin kieliteoreettisiin näkemyksiin kirjallisuuskatsauksen avulla. Sitten haastatteluilla pyritään saamaan selville, onko lingvistikista näkökulmista ollut käytännön hyötyä vai jäävätkö niistä saatavat edut teoreettiselle tasolle. Lisäksi luodaan katsaus ajalliseen muutokseen, pohditaan, minkälaisia tarpeita tulevaisuus tuo mukanaan ja voiko tuleviin ongelmiin löytyä vastauksia kieliteorioista.

1.4 Tutkielman rakenne

Tässä tutkielmassa käydään läpi tietojärjestelmien mallintamista, kielitiedettä ja niiden yhtymäkohtia. Lingvististen näkökulmien soveltamista tietojärjestelmien kehittämisessä käsitellään sekä teoriassa kirjallisuuskatsauksen avulla että empiirisessä osiossa asiantuntijahaastatteluilla.

Tutkielmassa on kahdeksan lukua. Aluksi johdannossa taustoitetaan aihetta ja selvennetään tutkimuskysymykset sekä tutkielman rakenne. Toisessa luvussa käydään läpi tärkeimmät käsitteet, kuvataan mallintamista yleisesti ja tarkastellaan kielen käyttöä mallintamisen välineenä. Kolmannessa luvussa tutustutaan mallinnuskielten historian lisäksi UML-kieleen, metamallinnukseen, sovellusaluekohtaisten kielten kehittämiseen sekä viimeaikaiseen kehitykseen tekoälyn ja kielimallien alalla.

Neljännessä luvussa esitellään kirjallisuudesta esiin tulleita kieliteoreettisia näkökulmia. Teorioista käydään läpi peruspiirteet ja joitakin käytännön sovellutuksia. Lisäksi kirjallisuuskatsauksen teorioista laaditaan yhteenveto.

Viidennessä luvussa käydään läpi empiirisen tutkimuksen lähtökohdat ja toteutus: haastattelujen tavoitteet, haastattelukysymykset, haastateltavien esittely sekä haastattelun toteutus. Tutkimustuloksia tarkastellaan kuudennessa luvussa, jonka lopuksi esitetään yhteenveto esiin tulleista seikoista.

Seitsemännessä luvussa esitetään pohdinta ja johtopäätökset, analysoidaan tulosten hyödynnettävyyttä ja luotettavuutta sekä esitetään jatkotutkimuksen aiheita. Lopuksi kahdeksannessa luvussa on yhteenveto.

2 YLEISTÄ MALLINTAMISESTA

Tutkielman tässä osiossa käydään läpi yleisiä mallintamiseen liittyviä aiheita. Ensin esitellään tärkeimmät tutkielmassa käytetyt käsitteet. Sitten käsitellään mallintamista yleisesti sekä erityisesti kieliä mallintamisen välineenä. Lisäksi pohditaan ketterän kehityksen vaikutusta tietojärjestelmien mallintamiseen.

2.1 Tärkeimmät käsitteet

Tässä osiossa esitellään lyhyesti tutkielman tärkeimmät käsitteet. Suomenkielisen termin perässä on annettu myös englanninkielinen termi, sillä tutkimuskirjallisuus on usein kirjoitettu englanniksi.

Tietojärjestelmä (information system): Tietojärjestelmästä on esitetty useita määritelmiä eri tutkimuksissa. Tässä tutkielmassa lähdetään näkemyksestä, jonka mukaan tietojärjestelmä on sosiotekninen järjestelmä (Beynon-Davies, 2009). Varajäon ym. (2022) mukaan tietojärjestelmä muodostuu älykkäistä toimijoista (joita voivat olla niin ihminen kuin tekoälykin), prosesseista ja IT-laitteista, -ohjelmista ja -infrastruktuurista, ja sen tehtävänä on datan, informaation ja tietämyksen jako organisaatiossa.

Yeon (2002) määritelmän mukaan tietojärjestelmä tallentaa, käsittelee ja toimittaa organisaatiolle tärkeää tietoa siten, että tiedosta on hyötyä sen käyttäjille eli johtajille, työntekijöille, asiakkaille tai toimittajille. Hän esittää, että tietojärjestelmä on organisaation tukena päätöksenteossa, strategiatyössä, operatiivisessa toiminnassa ja hallinnon analyysissa. Sen avulla voidaan parantaa tehokkuutta, toimintoja tai prosesseja käyttämällä tietotekniikkaa, manuaalisia toimintoja, malleja ja tietämys- ja tietokantoja. (Yeo, 2002.)

Lingvistiseltä kannalta tarkasteltuna tietojärjestelmä käsittelee symboleja, joilla on vaikutus ihmisten sosiaaliseen käyttäytymiseen. Näin ollen tietojärjestelmien alaan voidaankin katsoa lukeutuvan niin teknologian, organisaatioiden, johtamisen että lingvistisen alan tutkimusta. (Hirschheim, Klein & Lyytinen, 1996). Tässä tutkielmassa keskitytään pääosin lingvistiseen puoleen.

Ontologia (ontology): Sanakirjamääritelmän mukaan ontologia on ”olevaisen perimmäistä olemusta tutkiva filosofian osa” ja tietojenkäsittelytieteessä ”koneluettavassa muodossa oleva kuvaus tarkasteltavan alueen käsitteistä ja niiden välisistä suhteista” (Kotimaisten kielten keskus & Kielikone Oy, 2024). Tieteen termipankki jakaa filosofian alalla määritelmän kahteen osaan: oppiin olevasta sekä käsitykseen siitä, ”minkä tyyppisiä olioita on olemassa” (Tieteen termipankki, 2024b). Yleisen tieteen osalta termipankki kuvailee ontologiaa yhteisen käsitteistön formaaliksi, eksplisiittiseksi määritelmäksi (Tieteen termipankki, 2024a). Wandin, Monarchin, Parsonsin ja Woon (1995) mukaan ontologia pohtii, mitä maailmassa on. Mallinnuskielen tapauksessa maailma on kielen käyttöympäristö. Näin ontologian avulla voidaan myös selvittää, mitä käsitteitä mallinnuskielessä pitää olla. (Wand ym., 1995).

Husáková ja Bureš (2020) viittaavat ontologian filosofiseen taustaan: filosofiassa ontologian avulla vastataan metafyyssisiin kysymyksiin ja pohditaan ihmisyyttä. Vastaavasti tietojenkäsittelytieteessä ontologioilla on kuvattu yleistä tietoa. Formaalisia ontologioita sen sijaan käytetään esimerkiksi tietojärjestelmien, henkilöstöhallinnon, UML-pohjaisen mallinnuksen, koodin ja dokumentaation yhteensovittamiseen. (Husáková & Bureš, 2020).

Malli (model): Myös mallista on lukuisia määritelmä kirjallisuudessa. Erityisesti tietojärjestelmien mallintamiseen soveltuu Kielitoimiston sanakirjan määritelmä: jotakin ”kuviona, kaaviona tm. havainnollistava esitys” (Kotimaisten kielten keskus & Kielikone Oy, 2024). Mallit ovat ”usein idealisoituja tai abstraktioita kohteestaan” (Tieteen termipankki, 2024).

Mallin voidaan sanoa olevan myös yksinkertaistettu versio todellisuudesta (Schichl, 2004), järjestelmän abstraktio, jonka pohjalta voidaan tehdä ennusteita tai päätelmiä (Kühne, 2006), ymmärtää maailmaa ja helpottaa osapuolten välistä viestintää (Greenspan, Mylopoulous & Borgida, 1994). Kellnerin, Madachyn ja Raffon (1999) määritelmän mukaan mallin avulla kuvataan kohdejärjestelmä, jotta voidaan parantaa ymmärrettävyyttä ja tehostaa kehittämistyötä. Mallien eri toiminnoiksi Schichl (2004) luettelee ilmiöiden selittämisen, ennusteiden ja päätösten tekemisen ja viestinnän – tiedon välittämisen.

Keller ym. (1999) määrittävät, että malli voi olla visuaalinen tai tekstuaalinen, mutta visuaaliset mallit ovat vakiintuneet käyttöön ohjelmistoalalla. Visuaalinenkin malli sisältää silti tekstuaalista tietoa muun muassa eri komponenttien välisistä vuorovaikutuksista (Kellner, ym. 1999.) Erona matemaattisiin malleihin on se, että tietojärjestelmätieteen malleilla pyritään kuvaamaan järjestelmää, ei niinkään tulkitsemaan teoriaa (da Silva, 2015). Tässä tutkielmassa mallilla tarkoitetaan tietojärjestelmätieteen mallia.

Mallintaminen/mallinnus (model[ling]): Tietojärjestelmien suunnittelussa käytetään mallintamista. Mallintamisessa suunniteltavana olevasta järjestelmästä laaditaan kaavio, josta nähdään järjestelmän osat, roolit ja niiden suhteet toisiinsa. Se sujuvoittaa viestintää, tukee ymmärrystä ja auttaa dokumentoinnissa. (Wand ym., 1995.) Mallintaminen on iteratiivinen prosessi, jossa reaali maailman ongelmaan muodostetaan vaiheittaisesti ratkaisu rakentamalla malli, ke-

räämällä tietoa ja arvioimalla ratkaisu (Schichl, 2004). Mallintamista taustoitetaan lisää myöhemmin tässä luvussa.

Mallinnuskieli (mode(l)ling language): Mallintamisessa käytetty kieli. Mallinnuskielen voidaan katsoa sisältävän symbolijärjestelmän sekä symbolien yhdistelyä koskevat säännöt, joilla määritetään, millaisia malleja kielellä voidaan esittää (Bjeković 2017, s. 30). Da Silvan (2015) määritelmän mukaan mallinnuskieli on (vapaasti suomennettuna) ”kaikkien mallinnuskielen abstraktin syntaksin mukaisten mahdollisten mallien joukko, jota edustaa yksi tai useampi konkreettinen syntaksi ja joka noudattaa tiettyä semantiikkaa”. Mallinnuskielen tulos mahdollistaa kommunikointi, analyysi ja suunnittelu (Frank, 2002).

Kielitiede eli lingvistiikka (linguistics): Häkkisen (1995, s. 9) mukaan kielitiede tutkii ihmisten puheen järjestelmiä sekä kirjoitettuja vastineita niistä. Kielellä voidaan tarkoittaa myös tietokonekieliä ja muita vastaavia symbolijärjestelmiä. Erotukseksi niistä ihmisten kielijärjestelmiin viitataan termillä *luonnollinen kieli* (Häkkinen, 1995, s. 9). Kielitieteen voidaan myös sanoa tutkivan, miten ihmiset esittävät tietoa ja viestivät siitä keskenään (Wand ym., 1995). Tietojärjestelmien analysoinnissa on hyödynnetty etenkin kielitieteen syntaktisia, semanttisia ja pragmaattisia näkökulmia (Lyytinen, 1985).

Nämä voidaan määrittää seuraavasti:

- syntaksi: sääntöjen yhdistelmät (Noran 2000).

Da Silva (2015) jakaa syntaksit abstraktisiin ja konkreettisiin rinnastaen abstraktit syntaksit luonnollisten kielten kielioppiin tai mallinnuskielten osalta metamallinnustekniikoihin.

- semantiikka: symbolien merkitys (Noran 2000).

Da Silvan (2015) mukaan semantiikan avulla voidaan tulkita tietyn kielen syntaktisesti oikeat ilmaisut tai mallit. Ohjelmointikielten tapauksessa semantiikka kertoo, mitä tietokoneen pitäisi tehdä, kun kyseisellä kielellä kirjoitettu ohjelma suoritetaan (da Silva, 2015).

- pragmatiikka: ohjeet kielen käyttöön (Noran 2000). Da Silva (2015) määrittää pragmatiikan tutkimuksen olevan kiinnostuneen luonnollisen kielen tapauksessa viestintätilanteista sekä sosiaalisista, kulttuurisista jne. tekijöistä. Mallinnuskielistä puhuttaessa sillä viitataan käytännön käyttötilanteisiin, etenkin käyttäjien ja roolien määrittäisiin, tehtäviin toimiin ja esimerkiksi sosiaaliin tai psykologisiin seikkoihin (da Silva, 2015).

Sovellusaluekohtaiset kielet (Domain-Specific Languages, DSL): sovellusaluekohtaiset kielet ovat erityiselle sovellusalueelle kohdennettuja kieliä, joista saatuja etuja ovat juuri kyseiselle kohdealueelle sopiva ilmaisuvoimaisuus ja helppokäyttöisyys (Mernik, Heering & Sloane, 2005). Suomenkielisissä julkaisuissa käytetään myös termejä *toimialakohtainen* tai *täsmäkieli*. Näitä kieliä käsitellään enemmän osiossa 3.4

Suuret kielimallit (large language models): luonnollisen kielen käsittelyn (*Natural Language Processing, NLP*) kehitys on kulkenut tilastollisista malleista neuroverkkoihin ja esikoulutetuista kielimalleista (PLM) suuriin kielimalleihin (LLM) (Naveed ym., 2023). Buscemin (2023) määritelmän mukaan suuret kielimallit ovat edistyneitä tekoälyjärjestelmiä, joita on koulutettu suurella määrällä

dataa ymmärtämään ja tuottamaan aidontuntuista kieltä. Ne perustuvat tokenisointiin, jossa teksti jäsenellään saneisiin (*token*) – näitä voivat olla merkit, sanat, symbolit tai sanojen osat (Naveed ym., 2023). Mallien koulutuksessa käytetään suuria tekstiaineistoja, ja niitä voidaan hyödyntää monissa luonnollisen kielen käyttöä edellyttävissä tehtävissä (Liu ym., 2023). Kielimalleja ja tekoälyä käsitellään lisää luvussa 3.

2.2 Mitä mallintamisella tarkoitetaan

Malli on siis abstraktio tai yksinkertaistettu versio kohteena olevasta järjestelmästä, kuten edellä tärkeimmissä käsitteissä on määritelty. Tämä kohdemaailma (*Universe of Discourse*) voi olla jo olemassa tai vasta suunnitteilla (da Silva, 2015). Malli kuvaa kohteen tärkeimpiä toimintoja tai osa-alueita (Kellner ym., 1999) siten, että tietojärjestelmän vaatimukset täyttyvät (Jaakkola ym., 2016).

Mallin tarkoitus on siis auttaa kehittäjiä kuvaamaan järjestelmän rakennetta ja toimintaa. Se helpottaa osapuolten välistä viestintää sekä mahdollistaa näkemysten ja tiedon jakamisen teknisten asiantuntijoiden ja muiden osapuolten välillä. (Subahi & Alotaibi, 2018.) Siten mallintaminen tukee kehitysprojektin kaikkia vaiheita ja auttaa muodostamaan yhteisen näkemyksen kehitettävästä järjestelmästä (Jaakkola ym., 2016).

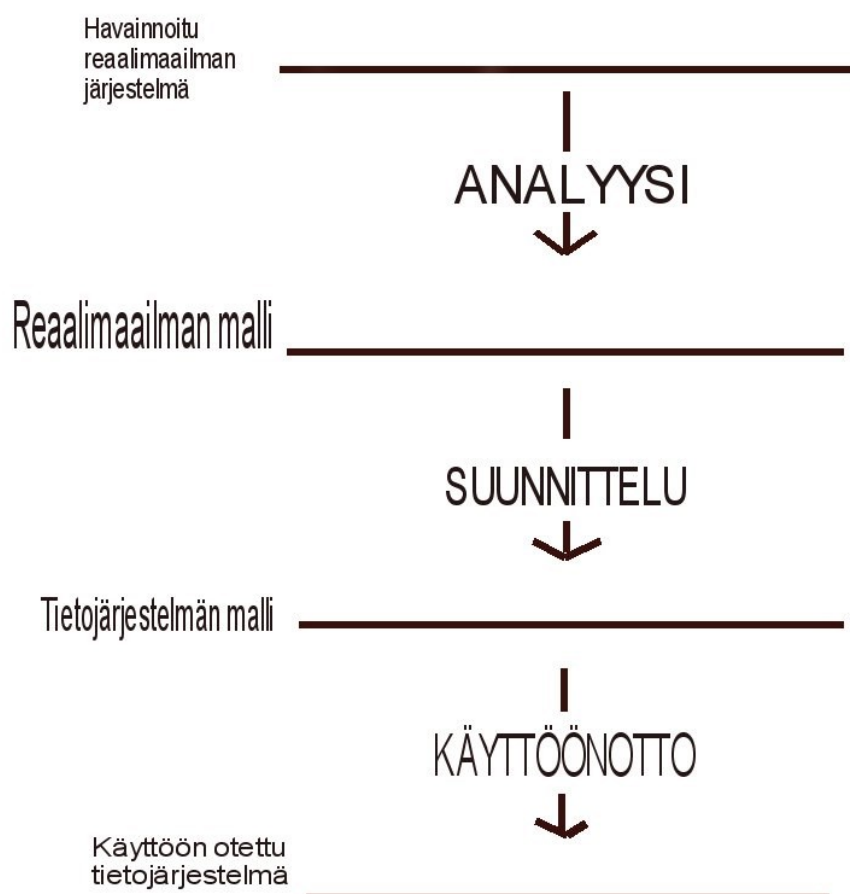
Schichl (2004) kuvaa, että mallien rakentaminen alkaa tarkasteltavasta reaali maailman kohteesta. Reaali maailman kohteiden tilalla käytetään mallissa yksinkertaisempia, usein samalla tavalla nimettyjä kohteita. Hänen määritelmänsä mukaan mallissa jäsenetään reaali maailmaa koskeva tieto ja pelkistetään se tärkeiksi katsotuiksi ilmiöiksi ja osa-alueiksi. Tähän tarvitaan usein monia iteraatioita. (Schichl, 2004.)

Mallintamisen avulla voidaan seuloa tarkasteltavasta järjestelmästä olennaisimmat osat ja keskittyä niihin (Giaglis, 2001). Giaglis (2001) tekee eron liiketoimintaprosessien mallintamisen (Business Process Modeling, BPM) ja tietojärjestelmien mallintamisen (Information Systems Modeling) välillä. Hän kuitenkin kokee niiden tukevan toisiaan ja arvioi mallintamisen mahdollistavan liiketoiminnan ja tietotekniikan yhteensovituksen (Giaglis, 2001).

Yritysmallintamisessa kuvataan mallilla yrityksen arkkitehtuuri, johon sisältyvät sekä liiketoiminnalliset että IT-toiminnot, käsitteet, näkökulmat ja työkalut (Fayoumi & Loucopoulos, 2016). Bjekovićin (2017, s 91) määritelmän mukaan yritysmalli on konseptuaalinen malli jostakin organisaation alueesta. Mallien tarkoituksena on antaa lisää tietoa yrityksestä, parantaa yhteistyötä, viestintää ja auttaa yrityksen muutoksessa. Siksi mallien pitääkin sopia yrityksen näkökulmiin ja kontekstiin myös kielsensä osalta. (Bjeković, 2017, s. 91.)

Kuviossa 1 esitetään käsitteellinen malli tietojärjestelmien kehittämisessä Wandin ym. mukaan (1995). Liikkeelle lähdetään havainnoidusta reaali maailman järjestelmästä, jota seuraa analyysi. Sitten todellisen maailman järjestel-

mästä laaditaan malli, ja lopuksi tietojärjestelmä otetaan käyttöön. (Wand ym., 1995.)



KUVIO 1. Mallit tietojärjestelmien kehittämisessä (Wandin ym., 2005, mukaan)

Bézivin ja Mullerin (1998) mukaan mallintamisella tarkoitetaan myös usein analysointia, jolloin ongelma jaetaan pienempiin, helpommin sisäistettäviin osa-alueisiin. Mallintamisessa laadittavan mallin muoto määritetään metamallissa, ja mallin sisältö riippuu kohteena olevasta ongelmasta (Bézivin & Muller, 1998). (Metamalleista lisää myöhemmin luvussa 3.3.) Funktionaalisessa mallinnuksessa tehtävät on jaettu funktioihin, olioperusteinen puolestaan perustuu olioiden käyttöön (Bézivin & Muller, 1998).

Lyytisen (1987) mukaan mallintaminen voidaan jakaa todellisuuden kartoittamiseen ja formaaliin kielen kehittämiseen. Todellisuuden kartoittamisen näkökulman mukaan tietojärjestelmä on ennakoitavissa oleva, formaali järjes-

telmä, joka peilaa kohdemaailmaa. Tähän käytetään formaalisti määritettyä kieltä. Toinen näkökulma Lyytisen (1987) jaottelussa on formaalisen kielen kehittämisen, jonka mukaan tiedon mallintaminen on organisaation sääntöjen laatisemisesta, kehittämisestä ja käyttöönottamisesta muodostuva prosessi.

Mallintamisen tarkoitus on Giaglisin (2001) mukaan tukea vuorovaikutusta ja viestintää, auttaa parantamaan prosesseja, tukea niiden hallintaa ja kehitystä sekä olla tukena prosessien toteutuksessa. Mallin pitää myös kyetä välittämään käyttäjälleen tarvittavat tiedot (Giaglis, 2001). Mallintamistyössä näiden tavoitteiden saavuttamisesta huolehditaan mallinnuskielten avulla.

Lyytisen (1987) mukaan kieli muokkaa aktiivisesti puhujien käsityksiä maailmasta. Sillä ei siis vain kuvata objektiivisesti todellisuutta, vaan se vaikuttaa myös siihen, millainen malli maailmasta laaditaan (Lyytinen, 1987). Siksi myös kielinäkökulman huomiointi mallintamisessa on paikallaan. Siihen perehdytään seuraavassa aliluvussa.

2.3 Kielet mallintamisen välineenä

Luonnollisissa kielissä ja tietojärjestelmien mallintamisessa voidaan nähdä monia yhtäläisyyksiä. Malli kuvastaa maailmaa kuten kieli, se huolehtii merkityksen välittämisestä ja ylläpitää sosiaalista vuorovaikutusta. Lisäksi kieliä käytetään mallintamisen välineenä, sillä mallin pitää kyetä välittämään käyttäjälleen tarvittavat tiedot (Giaglis, 2001). Tähän tarvitaan mallinnuskieliä. Subahin ja Alotaibin (2018) mukaan mallinnuskielet ovatkin yksi tärkeimmistä mallipohjaisen kehittämisen osa-alueista.

Kielen suhde varsinaiseen maailmaan voidaan rinnastaa mallintamisen ja maailman suhteeseen. Häkkisen (1995) mukaan ikoninen käsitys katsoo kielen jäljittelevän suoraan todellisuutta, jolloin kielenulkoinen maailma olisi suoraan määritelty kielen kategorioissa. Sen sijaan kognitiivis-psykologisen teorian mukaan kieliyhteisöt tulkitsevat maailmaa omalla tavallaan. Samalla voi syntyä erilaisia tulkintoja samasta asiasta, jolle myös annetaan eri nimityksiä. (s. 171–172). Mallintamisessa tietojärjestelmän suunnittelijat ja järjestelmää käyttävä organisaatio muodostavat omat kieliyhteisönsä, jotka tulkitsevat omaa toimintaympäristöään mallin muodossa.

Mallinnuskielten syntaksina on Harelin ja Rumpen (2000) mukaan kielen notaatio, kaaviot mukaan lukien. Da Silva (2015) jakaa syntaksin vielä konkreettiseen, notaation sisältävään syntaksiin sekä abstraktiin, metamallia vastaavaan syntaksiin. Kielitieteessä syntaksi tutkii puolestaan rakenteellisen kokonaisuuden muodostavia, sanaa suurempia kokonaisuuksia (Häkkinen, 1995, s. 146). Harel ja Rumpe (2000) määrittävät, että kielen syntaksiin voivat kuulua sanat, lauseet, kaaviot, mallit, moduulit jne. Heidän mukaansa tekstiin perustuvissa kielissä syntaksin perusteella muodostetaan lauseita, ikonisissa kielissä se tehdään kuvamerkeillä. Kaaviokielissä puolestaan kielen ilmaisut ovat esimerkiksi viivoja ja nuolia, ja ne ovat erityisen hyödyllisiä järjestelmän- ja ohjelmistonkehityksessä. (Harel & Rumpe, 2004).

Pragmatiikka puolestaan pohtii kielenkäytön merkitystä kielenkäyttötilanteen kontekstissa (da Silva, 2015). Mallinnuskielten pragmatiikka on da Silvan (2015) mukaan keskittynyt kielten käyttöön käytännössä, siinä ilmeneviin rooleihin ja toimintoihin sekä sosiaalisiin tai psykologisiin seikkoihin. Pragmatiikkaa huomioimalla pyritään usein parantamaan mallinnuskielten tehokkuutta ja toimintaa (da Silva, 2015).

Semantiikalla tarkoitetaan merkitysoppia, ja merkitykseen nojaututaan usein myös kielen muiden tasojen tutkimisessa, esimerkiksi lauseiden osien analysoinnissa (Häkkinen, 1995, s. 165). Semantiikka keskittyy kielen syntaktisesti oikeiden ilmaisujen merkitykseen (da Silva, 2015), josta sopiminen on sosiologinen prosessi (Harel & Rumpe, 2004), Häkkisen (1995) sanoin ”kielilyhteyden sisäiseen sopimukseen perustuva arvo” (s. 165).

Bjeković (2017) tähdentää, että merkkien ymmärtämisen kannalta on olennaista hahmottaa, mistä osa-alueista semanttinen merkitys muodostuu. Mallintamisen välineenä käytettäessäkin kielten semantiikkaa voidaan tarkastella de Saussuren merkkiopin avulla, jossa merkki on jaettu kahteen osaan, merkittävään ja merkittävään. (Bjeković, 2017, s. 27–29.) Peirce puolestaan käytti semanttista kolmiota, jonka kulmat muodostavat objekti eli tarkoite, interpretantti eli tulkinno ja merkitysväline eli ikoni, indeksi tai symboli (Mingers & Willcocks, 2014). Mingersin ja Willcocksin (2014) mukaan tämä jaottelu vastaa Fregen jakoa mieleen ja merkitykseen (*Sinn* ja *Bedeutung*), joita käsitellään osiossa 4.3.

Semantiikan ymmärtäminen, merkityksen sisäistäminen ja kuvaaminen auttavat Harelin ja Rumpen (2004) mukaan kehittämään parempia työkaluja ja menetelmiä nykyisille ja uusille mallinnuskielille. Fayoumi ja Loucopoulos (2016) nostavat esiin semantiikan merkityksen IT:n ja yrityksen liiketoimintatavoitteiden sovittamisessa. Heidän mukaansa semantiikan huomioinnilla pystytäisiin varmistamaan, että malli kuvastaa realistisesti kielen käyttöympäristön todellisuutta. Silloin IT:n ja yrityksen liiketoimintatavoitteiden yhteensovittaminen onnistuu helpommin, ja mallintamisen avulla voidaan kuroa umpeen hallinnon ja tietojärjestelmän välinen kuilu. (Fayoumi & Loucopoulos, 2016).

Hirschheim ym. (1996) korostavatkin kielen tehtävää organisaatioiden yhteisymmärryksen ja yhteistyön ja sitä kautta työn koordinoimisen mahdollistajana. Tässä tietojärjestelmien kielellä on tärkeä tehtävä, sillä se mahdollistaa viestien tallentamisen ja välittämisen. Ilman kieltä tietojärjestelmä ei voisi suoriutua tehtävistään. (Hirschheim ym. 1996.)

Kieliteorioita voidaan hyödyntää myös vaatimustenmäärittelyssä. Vaatimustenmäärittelyn tarkoituksena on selvittää tietojärjestelmän käyttötarkoitukset ja käyttökonteksti, ja se perustuu pitkälti luonnollisiin kieliin. Siksi olisi tärkeää, että kaikki osapuolet puhuvat samoista asioista samoilla sanoilla. (Chakraborty, Rosenkranz & Dehlinger 2012). Vaatimusmäärittelykielet (*Requirements Modeling Language*) ovat kuitenkin monessa mielessä eri asia kuin varsinaiset mallinnuskielet, joilla pyritään puolestaan kuvaamaan ihmisen tietoa ja käsityksiä maailmasta (Greenspan ym., 1994).

Mallipohjaisen kehittämisen arvellaan auttavan lisäämään tuottavuutta, vähentämään kustannuksia, parantamaan ohjelman uudelleenkäyttämävyyksiä ja auttamaan havaitsemaan vikoja ja väärinkäsityksiä ajoissa (Liddle, 2003). Mallipohjaista suunnittelua ja arkkitehtuuria on kuitenkin Liddlen (2003) mukaan kritisoitu monimutkaisuudesta ja käytön vaikeudesta, ja monimutkaisten mallien luominen ja käyttäminen koodin generoimiseen on ristiriidassa ketterän kehittämisen asiakastytyvyyden ja jatkuvan toimituksen periaatteiden kanssa. Tähän näkökulmaan tutustutaan seuraavaksi.

2.4 Mallintaminen ja ketterä kehitys

Ohjelmistonkehityksessä oli 1980- ja 1990-luvuilla yleisesti käytössä prosessi, jossa edettiin huolellisen suunnitelman mukaan. Menetelmään liittyi paljon ylimääräistä suunnittelusta ja dokumentaatiosta aiheutuvaa painolastia, ja muutosten tekeminen vaikeutui. Näiden ongelmien ratkaisemiseksi 1990-luvulla kehitettiin ns. ketteriä menetelmiä ("*agile methods*"). (Sommerville 2011, s. 58.)

Ketterän kehittämisen periaatteet on kuvattu Agile Manifesto -manifestissa. Sommerville (2011) kuvaa, että näiden periaatteiden mukaisesti menetelmässä asetetaan etusijalle muun muassa yksilöt ja vuorovaikutus prosessien ja työkalujen sijaan, ohjelmiston toiminta dokumentoinnin sijaan, asiakkaan kanssa vuorovaikutus sopimusneuvottelujen sijaan sekä muutokseen reagointi suunnitelman noudattamisen sijaan (s. 59). Sommervillen em. julkaisussa mainitsemia esimerkkejä ovat Extreme Programming ja Scrum.

Ketterän kehityksen periaatteiden voi nähdä sotivan mallipohjaisia menetelmiä vastaan, ja ketterän kehittämisen ja mallintamisen yhteensopivuutta onkin epäilty. Etukäteen tehtävä mallinnustyö ei ehkä taivu ketteriin muutoksiin, ja ketterään kehittämiseen tottuneet voivat kokea mallinnustyön vieraaksi (Sommerville 2011, s. 141).

Ambler (2003) epäilee niin ikään mallipohjaisen kehittämisen mahdollisuuksia nykyisen kehittäjäskupolven keskuudessa. Golra, Beugnard, Dagnat, Guerin ja Guychard (2016) siteeraavat myös omassa artikkelissaan Amblerin ajatuksia. He kirjoittavat, että mallipohjaisen kehittämisen vastustajien mukaan on vain ajan haaskausta liikkua korkeammalla abstraktiotasolla ennen koodausta, ja siihen kuluva aika kannattaisi sen sijaan käyttää itse kehitystyöhön. Yleisesti ottaen asiakkaat eivät myöskään halua monimutkaisia malleja vaan toimivan järjestelmän (Liddle, 2003).

Vastaavasti sovellusaluekohtaisten kielten kehittämisessä (ks. luku 3.4) noudatetaan tyypillisesti perinteisiä menetelmiä, joissa kehittämisessä edetään vaiheittaisesti metatason määrittämisestä mallien kehittämiseen. Tässä vaikeutena voi Golran ym. (2016) mukaan olla kielen määrittelyn lisäksi käyttäjän vaatimusten muuttaminen käsitteiksi. Ketterissä menetelmissä asiakasta määriteltäviä kuullaan kehityksen ohella, ja järjestelmää kehitetään vaiheittain (Golra

ym., 2016). Ambler (2003) kuitenkin kannattaa ”ketterää mallipohjaista mallintamista”, jolla syntyvä malli olisi joustava eikä sitoisi liikaa.

Sommervillen (2011) mukaan mallipohjaisen arkkitehtuurin kehittäjät sanovat mallinsa tukevan myös iteratiivista lähestymistapaa. Menetelmät voitaisiin saada sopimaan yhteen, jos koko ohjelman voisi generoida mallista eikä erillistä koodausta tarvittaisi. Siinä tapauksessa mallinnustyökalun pitäisi tukea myös regressiotestausta ja testilähtöistä kehitystä (Sommerville, 2011, s. 141).

Ketterän kehityksen ja mallintamisen yhdistämiseen uskovat omassa tutkimuksessaan myös Golra ym. (2016). Kommunikointiongelmia onnistuttiin vähentämään kehittämällä graafista sovellusaluekohtaista mallia interaktiivisesti ja iteratiivisesti yhdessä asiakkaan kanssa. Ketterä lähestymistapa voikin auttaa myös sovellusaluekohtaisten mallien kehittämisen ongelmissa, jos ongelmien taustasyynä on riittämätön kommunikaatio kehittäjien ja asiakkaiden välillä (Golra ym., 2016).

Tässä luvussa on käyty läpi, mitä mallintamisella tarkoitetaan, miten kieliä käytetään mallintamisen välineenä ja miten mallintaminen mahdollisesti sopii yhteen ketterän kehityksen periaatteiden kanssa. Seuraavassa luvussa katsotaan tarkemmin mallinnuskieliä, niiden historiaa, erilaisia mallinnuskielten tyyppejä, metamallintamista ja sovellusaluekohtaisia kieliä. Lisäksi luodaan lyhyt katsaus tekoälysovelluksissa hyödynnettyihin kielimalleihin.

3 MALLINNUSKIELTEN KEHITTÄMINEN JA KEHITTÄMINEN

Kyky kehittää abstrakteja malleja on ollut ihmiselle ominainen taito, josta on ollut hyötyä selviytymisen kannalta (Schichl, 2004). Hyödyksi on ollut myös kyky luokitella samanlaisia asioita yhteen ja erottaa erilaiset toisistaan – taito, johon perustuvat myös käsitteellinen mallintaminen ja tietojärjestelmien toiminta (Wand ym., 1995). Schichlin (2004) mukaan malleina voidaan pitää jo ensimmäisiä ihmisen käyttämiä numeroita, ja laskeminen johti vähitellen matemaattisten mallien syntyyn.

Liddle (2003) arvioi, että ohjelmointikielten ja käyttöjärjestelmien historiassa on siirrytty aina ylemmälle abstraktiotasolle. Seuraavassa luvussa tarkastellaan mallinnuskielten syntyhistoriaa ja käydään läpi joitakin yleisimmistä mallinnusmenetelmistä. Omissa alaluvuissaan käsitellään UML-kieltä, metamallintamista, toimialakohtaisia kieliä sekä viimeaikaisen tekoälyn hyödyntämiä suuria kielimalleja.

3.1 Mallinnuskielten historia

Mallintaminen on alkanut varhain ihmiskunnan historiassa: jo luolamaalausten voidaan katsoa olevan abstrakteja representaatioita reaali maailmasta, ja ensimmäisinä numeroina voidaan pitää luhin tehtyjä merkkejä (Schichl, 2004). Varsinainen mallintaminen kehittyi kuitenkin muinaisen Lähi-idän ja antiikin Kreikan kulttuureissa, ja 2000 eaa. Babyloniassa, Egyptissä ja Intiassa käytettiin matemaattisia malleja ongelmien ratkaisemiseen. Mallien hyödyllisyyden vuoksi vastaavia menetelmiä alettiin vähitellen kehittää eri puolilla maailmaa. (Schichl, 2004).

Schichlin (2004) mainitsemia merkittäviä hahmoja matemaattisten mallien historiasta ovat kreikkalaiset Eukleides (300 eaa.) ja Pythagoras, arabialainen Abu Abd-Allah ibn Musa al-Khwārizmī (800-luvulla), jonka nimestä on johdettu sana *algoritmi*, sekä Fibonacci (n. 1170–1240), joka havaitsi arabialaisten nu-

meroiden edut ja toi nollan Eurooppaan. Vähitellen mallien käyttö levisi matematiikasta myös muille aloille. (Schichl, 2004.)

Nykyään mallintamisessa käytettyjä tekniikoita ovat tietovuot (*data flow*), ER (Entity-Relationship) -kaaviot, tilanmuutoskaaviot (*state-transition*), IDEF-tekniikat ja UML (*Unified Modeling Language*) (Giaglis, 2001). Liiketoimintaprosessien mallintamisessa käytetään laajalti BPMN-kieltä (*Business Process Model and Notation*). Lisäksi käytössä on SBVR-standardi, jolla voidaan Fayoumin ja Loucopoulosin (2016) mukaan määrittää toimialan käsitteisiin perustuvia sääntöjä, sanastoja ja käytäntöjä liiketoiminnan tavoitteiden mukaisesti. MEMO puolestaan tulee sanoista Multi-Perspective Enterprise Modeling. Se auttaa sovellusaluekohtaisten kielten kehittämisessä kolmen abstraktiotasonsa (Fayoumi & Loucopoulos, 2016) ja erityisten visuaalisten mallinnuskielten avulla (Frank, 2002).

Yksi vanhimpia mallinnustekniikoita on vuokaavio. Vuokaavioiden kehittäjinä pidetään Frank ja Lilian Gilbrethiä, jotka esittelivät kaavion vuonna 1921 (Krajewski, 2000). Krajewski (2000) tosin huomauttaa, että varsinaisen keksijän määrittäminen voi olla vaikeaa, sillä vastaavanlaisia graafisia menetelmiä oli käytetty jo aiemmin.

IBM:n (1969) määritelmän mukaan vuokaaviolla esitetään tiedon vuo eli virtaus, jolla kuvataan järjestelmässä tehtävät toimet niiden suoritusjärjestyksessä. Vuokaavioissa tiedot kuvataan symboleilla, esimerkiksi erimuotoisilla laatikoilla ja nuolilla (IBM, 1969). Giaglis (2001) kuvaa, että aluksi vuokaavioilla esitettiin tietokoneohjelmien logiikkaa, mutta kaaviotyypin käyttö levisi myös muihin mallinnussovelluksiin. Hän määrittää, että vuokaavioita käytetään etenkin monimutkaisten prosessien esittämiseen yksinkertaisemmassa graafisessa muodossa (Giaglis, 2001).

Tietovuokaaviot (*data flow diagramming*, DFD) kehitti puolestaan DeMarco (1979). Menetelmä soveltuu tietovirtojen dokumentointiin prosessin eri tahojen tai vaiheiden välillä. Tietovuokaaviot ovat paljolti tietovuomallien kaltaisia, mutta niissä keskitytään toimintojen sijasta dataan (Giaglis, 2001). Tietovuokaavio ei myöskään kerro, mitä polkua seurata tai mikä prosessin käynnistää, vaan se esittää joukon mahdollisia polkuja (DeMarco, 1979). Giaglisin (2001) mukaan pelkkään dataan esimerkiksi ajallisten suhteiden sijasta keskittyminen on ollut myös tietovuokaavioiden heikkous. Siksi niitä ei ole helppo käyttää analyysiin tai päätöksentekoon (Giaglis, 2001).

Petri-verkot kehitti vuonna 1962 C. A. Petri, joka halusi kuvata tietokoneen komponenttien vuorovaikutusta ja kausaalisia suhteita tapahtumien välillä (Peterson, 1981, s. 3). Petri-verkot ovat matemaattisia/graaafisia esityksiä järjestelmästä ja koostuvat tiloista (*states*) ja siirtymistä (*transition*). Ne eivät soveltu monimutkaiseen korkean tason mallinnustyöhön, joskin niihin on lisätty laajennuksina ns. korkean tason Petri-verkkoja ja väritettyjä Petri-verkkoja (Giaglis, 2001).

Entity-Relationship eli ER-kaaviot kehitti Chen (1976). Niissä entiteeteillä (*entity*) viitataan johonkin tunnistettavissa olevaan asiaan, esimerkiksi henkilöön. Suhde (*relationship*) puolestaan kuvaa entiteettien välistä yhteyttä. Semant-

tisesti merkitsevä tieto valitaan perusavaimeksi (*primary key*). Lisäksi kaaviolla voidaan ilmaista entiteettien lukumäärä, esim. 1:n tai 1:1. Chen (1976) kuvaa, että ER-kaavioilla voidaan esittää tärkeää semanttista tietoa reaali maailmasta entiteettien ja niiden suhteiden välillä, ja niitä voidaan käyttää tietokantojen suunnitteluun.

Tilakaavio (*state transition diagram*) on ER-kaavioiden ja tietovuokaavioiden kaltainen graafinen notaatio, jossa solmut ovat järjestelmän tiloja (*state*) ja nuolet puolestaan siirtymiä (*transition*) tilojen välillä. Tilakaaviot kehittivät G.H. Mealy ja E. F Moore 1950-luvulla äärellisen automaatin parissa tehtyjen tutkimusten pohjalta. (Desharnais, Frappier & Mili, 1998.) Giaglisin (2001) mukaan tilakaavioiden tarkoitus oli ratkaista muiden DFD- ja ER-kaavioiden staattisuudesta aiheutuvat ongelmat. Nekin kuitenkin keskittyvät enemmän dataan ja laiminlyövät mm. työnkulun ja päätöksenteon. Näin ollen ne soveltuvatkin juuri tietojärjestelmiin paremmin kuin liiketoiminnan mallinnukseen. (Giaglis, 2001.)

Tilakaavioita on käytössä myös UML-kielessä (Sommerville 2011, s. 135), josta lisää seuraavaksi.

3.2 UML

UML on olioperustainen, visuaalinen mallinnuskieli, jossa käytetään notaatioita ja tekstikomponentteja (Grossman ym., 2005). Se koostuu symboleista eli notaatiosta ja säännöistä eli semantiikasta (Noran, 2000). UML on mallinnuskieli, ei menetelmä, sillä yleensä menetelmissä on mukana myös prosessi, joka kuvaa suunnittelun etenemisvaiheet. UML:n voi kuitenkin yhdistää Rational Unified Processin käyttöön. (Fowler & Scott, 2000).

Bézivin ja Muller (1998) taustoittavat, että UML syntyi 1990-luvulla oliopohjaisuuden alkaessa vallata alaa. Heidän mukaansa markkinoilla oli aluksi useita kilpailevia menetelmiä, mutta niiden väliset erot alkoivat kaventua ja lopulta kilpailu ei enää hyödyttänyt ketään osapuolta. Jim Runbaugh ja Grady Booch päättivät kehittää yhteisen menetelmän, ja lopulta käyttötapauksia kehittänyt Ivar Jacobson tuli mukaan. (Bézivin & Muller, 1998.) Universaalien kielten toivottiin mm. ratkaisevan eri kielten väliset yhteensopivuusongelmat (Giaglis, 2001).

UML:n kehittäjäkolmikkoa alettiin kutsua nimellä "Three Amigos" (Fowler & Scott, 2000). He kehittivät uuden mallinnuskielen pääasiallisiksi kilpailijoiksi nousseiden Booch Methodin, Object Modeling Techniquen ja Objectory Methodin parhaiksi katsottujen ominaisuuksien perusteella. Kielen kehitystä hallinnoi Object Management Group (OMG). (Grossman ym., 2005.) Lopputuloksena syntynyt UML-standardi hyväksyttiin joulukuussa 1997 (Bézivin & Muller, 1998).

Uuden Unified Methodiksi nimetyn menetelmän ensimmäinen versio esiteltiin lokakuussa 1995. Universaaliksi olioperustaiseksi kieleksi kehittyttyään sen nimeksi tuli UML (*Unified Modeling Language*). (Bézivin & Muller, 1998.)

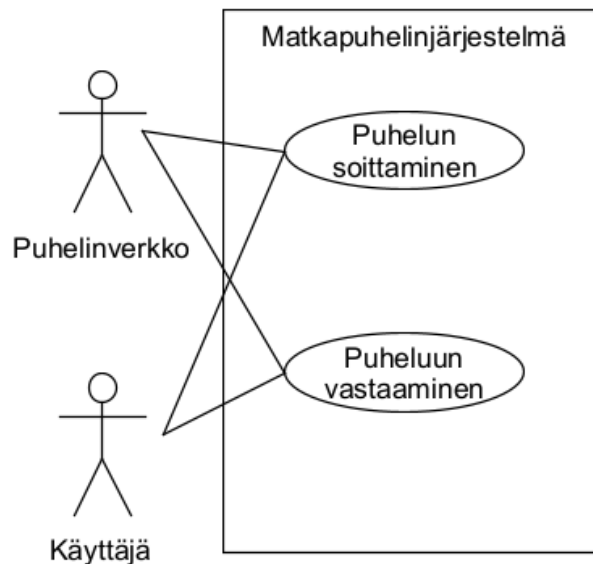
UML määrittää kielen notaation ja metamallin - notaatio on graafinen muoto, mallinnuskielen syntaksi. (Fowler & Scott, 2000.)

Giaglis (2001) perustelee kielen kehitystä sillä, että vaikka useita merkin-
täkieliä oli ollut, useimmat oli tarkoitettu johonkin tiettyyn analyysi- ja suunnit-
telumenetelmään. Kielet eivät olleet yhteensopivia. UML:n tarkoitus oli ratkais-
ta ongelmat universaaliudellaan ja soveltua niin tietokantoihin, ohjelmistokom-
ponentteihin kuin liiketoimintaprosessienkin kuvaamiseen. (Giaglis, 2001.)

UML:ssä oli alkujaan 9 kaaviotyyppeä (Bézivin & Muller, 1998). UML 2.0 -
versiossa kaaviotyyppejä on 13, ja niitä voidaan käyttää eri näkymien ja abst-
raktiotasojen mallintamiseen (Lange & Chauldron, 2006). Näistä viideksi tär-
keimmäksi voidaan mainita seuraavat (Sommerville, 2011, s. 120):

- aktiviteettikaavio (*activity diagram*), joka kuvaa prosessin toimintoja
- käyttötapauskaavio (*use case diagram*), joka osoittaa vuorovaikutuk-
sen järjestelmän ja sen ympäristön välillä
- sekvenssikaaviot (*sequence diagram*), jotka näyttävät toimijoiden ja
järjestelmän komponenttien väliset vuorovaikutukset
- luokkakaaviot (*class diagram*), jotka esittävät järjestelmän luokkia ja
niiden välisiä yhteyksiä sekä
- tilakaaviot (*state diagram*), jotka kuvaavat järjestelmän reaktiota si-
säisiin ulkoisiin ja tapahtumiin

Siitä jäljempänä esitetyssä kuvassa tikku-ukko on toimija ja ympyrät toi-
mintoja, jotka ovat neliöön rajatun järjestelmän sisällä.



KUVIO 2 Yksinkertaistettu esimerkki UML-käyttötapauskaaviosta (Giaglisia, 2001, muka-
ellen)

Noranin (2000) luettelemia muita luokkia ovat esimerkiksi oliokaavio (*object diagram*), joka esittää luokkakaavion tietyn tilanteen, tai tilakaavio, josta näkee järjestelmän mahdolliset tilat. Erilaisten kaavioiden avulla on tarkoitus kuvata järjestelmän rakennetta, käyttäytymistä ja toimintoja (Noran, 2000.)

UML:n tärkeimpiä etuja on viestinnän helpottaminen: Fowler ja Scott (2000) katsovat, että sen avulla voi viestiä monimutkaisista käsitteistä täsmällisemmin kuin luonnollisella kielellä mutta yksinkertaisemmin kuin koodilla. UML auttaa saavuttamaan sopivan tarkkuuden menemättä liiaksi yksityiskohtiin, ja sen avulla voi korostaa juuri tärkeitä seikkoja (Fowler & Scott, 2000).

UML:n on koettu tuovan useita etuja perinteisiin prosessipohjaisiin ratkaisuihin verrattuna (Grossman ym., 2005). Myös mallipohjainen kehittäminen on lisännyt UML:n merkitystä (Lange & Chauldron, 2006). Kieltä on pidetty johdonmukaisena ja joustavana, mutta sitä on silti myös kritisoitu mm. monimutkaiseksi ja vaikeaksi oppia ja semanttisesti epätarkaksi (Grossman ym., 2005).

Yhdeksi UML:n ongelmista on mainittu sen etujen liiallinen korostaminen. Silloin käyttäjät odottavat kieleltä liikaa, eikä se pystykään käytännössä täyttämään kaikkia odotuksia. (France, Ghosh & Dinh-Trong, 2006.) Lisäksi olioperusteisuutensa vuoksi UML:n voidaan katsoa soveltuvan heikommin käytettäväksi jonkin toisenlaisen mallinnustavan kanssa (Giaglis, 2001).

Tutkimuksissa on todettu, että UML-kieltä on käytössä kaikenkokoisissa projekteissa eri puolilla maailmaa (Grossman ym., 2005). Siinä ei ole virallista semantiikkaa, mikä onkin koettu sekä sen vahvuudeksi että heikkoudeksi – toisaalta se voi aiheuttaa kommunikaatio-ongelmia, toisaalta tuo vapautta (Lange & Chauldron 2006).

Mallinnuskielten kehitys ei ole myöskään päättynyt tähän, vaan se on jatkuvasti käynnissä. UML:kään ei ole ollut täysin ongelmaton, joten ongelmien ratkaisemiseksi on kehitetty versio 2.0. France ym. (2006) arvioivat, että versiossa pyritään tiivistämään turhan laajaksi paisuneita osa-alueita ja helpottamaan automatisointia. Siinä huomioidaan myös sovellusaluekohtainen mallinnus paremmin kuin aikaisemmassa versiossa (France ym., 2006). Sovellusaluekohtaisesta mallinnuksesta ja metamallinnuksesta lisää seuraavissa luvuissa.

3.3 Metamallintamisen merkitys

Sommervillen (2011) määritelmän perusteella mallipohjainen suunnittelu perustuu mallipohjaiseen arkkitehtuuriin (*Model-Driven Architecture*, MDA), jonka Object Management Group (OMG) kehitti vuonna 2001 uudeksi kehitysparadigmaksi (s. 138). Mallipohjaisella suunnittelulla (MDE) ja arkkitehtuurilla tarkoitetaan usein samaa asiaa, mutta Sommervillen mukaan MDE on laajempi käsite, joka koskee kaikkia ohjelmistonsuunnitteluprosessin vaiheita (Sommerville, 2011, s. 138–139).

Malleissa käytettävät mallinnuskielet määritetään metamalleissa, ja metamallien määrittämisessä käytetään meta-metamallia (Frank, 2002). Metamallia voidaan siis sanoa malliksi, jolla määritetään mallinnuskielen rakenne ja

malli, ja vastaavasti se tarvitsee oman mallinnuskielensä määrittämiseen metamallin (da Silva, 2015). Seidewitzin (2003) määritelmän mukaan metamallissa kerrotaan, mitä tietyllä mallinnuskielellä tehdyillä malleilla voidaan ilmaista. Metamalli on malli, joten se ilmaistaan mallinnuskielellä (Seidewitz, 2003). Tähän tarkoitukseen OMG on kehittänyt Meta Object Facility -tason (da Silva, 2015).

Malleja on myös alettu pitää pelkän dokumentaation sijasta ohjelmiston suunnittelun varsinaisena artefaktina (da Silva, 2015). Tällöin mallinnuksesta syntyy siis malli, ei ohjelmaa (Sommerville, 2011, s. 138). Tässä voidaan hyödyntää metamallinnusta ja koodin generointia, jossa ohjelmistojärjestelmiä voidaan suorittaa automaattisesti mallien pohjalta. Tällaisia menetelmiä ovat esimerkiksi model-driven engineering (MDE), model-driven development (MDD) tai model-driven software development (MDSD). (Da Silva, 2015.) Mallipohjaisessa suunnittelussa koodi pitäisi pystyä tuottamaan suoraan mallista. Se kuitenkin edellyttää graafisten mallien semantiikan tarkkaa määrittämistä. (Sommerville 2011, s. 142).

Atkinsson ja Kühne (2003) arvioivat, että metamallinnus helpottaa abstraktiotason nostamista. Sen ansiosta suunnittelijat voivat keskittyä miettimään koodin yksityiskohtien sijaan sitä, mitä toimintoja ja millaista arkkitehtuuria tarvitaan. Viime kädessä se auttaa parantamaan tuottavuutta kahtalaisesti: lisäämällä ohjelmistoartefaktin käyttökelpoisuutta sekä pidentämällä ohjelmistokoodin mahdollista käyttöikä (Atkinsson & Kühne, 2003). Vastaavasti virheiden vaara vähenee ja suunnittelu nopeutuu (Sommerville 2011, s. 139).

Atkinsson ja Kühne (2003) havainnollistavat metamallinnusta eri mallitasoja esittelevällä infrastruktuurirakenteella. Siinä jokainen taso toimii edeltävän instanssina, ja abstraktiossa edetään käyttäjän tiedoista malliin, metamalliin ja lopulta meta-metamalliin (Atkinsson & Kühne, 2003). He kuitenkin kritisoivat perinteistä infrastruktuuria vaikeudesta määrittää, mille tasolle mikin elementti kuuluu tai miten se liittyy reaaliin maailmaan, ja ratkaisuksi he esittävät lingvistisen ja ontologisen määritelmän yhdistävän lähestymistavan.

Seuraavassa taulukossa on annettu esimerkki mallien ja metamallien eri tasoista (taulukko 1).

TAULUKKO 1 Mallien eri tasot suhteessa toisiinsa. (Neton & Neton, 2011, mukaan)

Taso 2	Metamalli	Tason 1 käsitteiden luokittelu
Taso 1	Malli	Edustaa ylemmän tason instansseja ja luokittelee alemman tason käsitteet
Taso 0	Reaaliin maailman kohde.	Edustaa 1-tasolla määritettyä käsitettä

UML 2.0 painottaa lingvististä ontologiaa, mutta Atkinssonin & Kühnen (2003) mukaan sovellusalueen käsitteitä ja niiden ominaisuuksia on kuvattava

ontologisella metamallinnuksella. Esimerkiksi "Lassie" olisi "Collien" ontologinen ilmentymä, ja "Collie" puolestaan ilmentäisi "rotua". "Rodun" alle voidaan sitten jakaa eri metatyyppit, tässä tapauksessa eri koirarodut, ja määrittellä niiden ominaisuuksia. (Atkinsson & Kühne, 2003.) Lingvistiksi "Lassie" voidaan luokitella substantiiviksi, kun taas semanttiselta kannalta "Lassie" on "koiran" tyyppi tai elokuvan eläinhahmo (Neto & Neto, 2011).

Tasomäärittelyistä Liddle (2003) esittää myös nelitasoisen rakenteen:

- taso 0: datainstanssin taso, jolla on oliot, tietueet, data ja niihin liittyvät artefaktit.
- taso 1: käyttäjän malli tai mallin instanssi: luokkakaaviot, tilakaaviot ja vastaavat
- taso 2: metamallin taso – käsitteet, joista metamalli koostuu
- taso 3: meta-metamalli, UML:n tapauksessa OMF. (Liddle, 2003).

Tasot ovat yhteydessä toisiinsa. Neto ja Neto (2011) arvelevat, että lopputuotteen ongelmat juontavat usein juurensa suunnitteluvaiheisiin ja käsitteiden määrittelyyn, ja käsitteellisten mallien parantaminen voisi siis parantaa koko tuotteen laatua.

3.4 Sovellusaluekohtaisten kielten kehitys

Mallinnuskielet voidaan luokitella yleiskäyttöisiin (*general-purpose*) ja sovellusaluekohtaisiin (*domain-specific*). Esimerkki yleiskäyttöisestä mallinnuskielestä on UML, joka soveltuu monenlaisiin käyttökohteisiin. (Da Silva, 2015.) Tunnettuja sovellusaluekohtaisia kieliä ovat SQL, BNF ja HTML (van Deursen, Klint & Visser, 2000). Van Deursenin ym. (2000) määritelmän mukaan sovellusaluekohtainen kieli (DSL) on ohjelmointikieli tai suoritettava määrittelykieli, joka keskittyy ja yleensä rajoittuu johonkin tiettyyn ongelma-alueeseen.

Sovellusaluekohtaiset kielet eivät ole uusi keksintö, ja Mernik ym. (2005) viittaavatkin jo 1950-luvulla kehitettyyn APT:hen sekä vuodelta 1959 peräisin olevaan BNF-muotoon. Van Deursen ym. (2000) kirjoittavat Cobolin, Fortranin ja Lispin syntyneen tiettyihin ongelmiin keskittyneinä kielinä, mutta siitä ne ovat kehittyneet edelleen yleiskieliksi.

Merrik ym. (2005) huomauttavat, että kielten luokittelu ei ole aina selvärajaisista, vaan kielet voidaan sijoittaa asteikoille, jonka toisessa laidassa ovat puhtaasti yleiskäyttöiset ohjelmointikieliset ja toisessa täysin sovellusaluekohtaiset. Sovellusaluekohtainen kieli voi myös hyödyntää toista, valmista kieltä esimerkiksi laajentamalla sitä (Mernik ym., 2005). Sovellusaluekohtaiset kielet ovat kuitenkin yleensä tekstuaalisia tai graafisia tai niiden yhdistelmiä. Graafisissa kielissä käytetään symboleista ja viivoista ja kuvioista koostuvia kaavioita. Lisäksi voidaan käyttää avainsanoja ja luonnollisen kielen termejä. (Subahi & Alo-taibi, 2018.)

Liddlen (2011) mukaan sovellusaluekohtaiset kielet liittyvät läheisesti mallipohjaiseen kehittämiseen. Sovellusaluekohtaisessa mallintamisessa (DSM) kuvataan järjestelmää käyttökohteen tarpeisiin mukautetun kielen avulla. Tähän liittyy usein mahdollisuus generoida koodia järjestelmän ohjelmistoihin. (Liddle, 2011.)

Sovellusaluekohtaiset kielet on kohdennettu käyttökohteeseensa, minkä vuoksi niiden yleiskäyttöisyys kärsii, mutta vastaavasti tehokkuus ja ilmaisykyky kyseisessä käyttökohteessa paranevat (Mernik ym., 2005). Niitä on myös usein helpompi tulkita ja hyödyntää kehittäjien ja toimialan asiantuntijoiden välisessä vuorovaikutuksessa (da Silva, 2015).

Schmidt (2006) määrittää, että sovellusaluekohtaiset mallinnuskielet kuvaavat tietyn käyttökohteen rakennetta, käyttäytymistä ja vaatimuksia. Sovellusalueen käsitteet ja niiden suhteet, semantiikka ja rajoitteet kuvataan meta-malleissa. Metamallinnuksella voidaan siten muokata kieliä niin, että ne soveltuvat juuri kyseisen sovelluskohteen semantiikkaan ja syntaksiin – yleiskäyttöiset notaatiot harvoin yltyvät samaan tarkkuuteen. (Schmidt, 2006).

Sovellusaluekohtaisilla kielillä kehitetyissä ratkaisuihin voidaan käyttää ongelma-alueen kieltä ja abstraktiotasoa (van Deursen ym., 2000). Sovellusalueelle sopivien oikeiden käsitteiden ja määritysten käyttö tehostaa mallinnusprosessia (Luoma, Kelly & Tolvanen, 2004) ja helpottaa vuorovaikutusta eri alojen ammattilaisten välillä, sillä sovellusalueen käsitteitä on helppo lukea ja ymmärtää (da Silva, 2015).

Samoin Luoman ym. (2004) mukaan myös sovellusalueen vaatimukset pystytään huomioimaan paremmin, virheiltä voidaan välttyä ja ohjelmistonkehitysprosessin vaiheita automatisoida. On siis syytä arvella, että tässä käytössä lingvistiset lähtökohdat ja niiden huomiointi olisivat erityisen merkittävässä asemassa.

Sovellusaluekohtaisten kielten käyttö ei silti tuo pelkkiä etuja. Mernik ym. (2005) korostavat, että kielten kehittämisessä tarvitaan sekä käyttökohteen että kielen kehittämisen tuntemusta. Ne ovat vaativia taitoja, joita kaikilla ei ole. Kehittämisessä käytettävät tekniikat ovat monimutkaisia, ja vastaavasti koulutus, tuki ja ylläpito saattavat osoittautua vaativiksi. (Mernik ym., 2005.)

Van Deursen ym. (2000) mainitsevat haitoiksi myös kustannuspuolen: niin suunnittelun, käyttöönoton ja ylläpidon kuin koulutuksenkin aiheuttamat kustannukset. Kehittämisen ensimmäinen vaihe onkin päätöksen tekeminen sen suhteen, kannattaako kieli kyseisessä tilanteessa ja organisaatiossa kehittää, ja miten kehittäminen tehdään (Mernik ym., 2005). Analyysivaiheessa ongelma-alue määritetään ja kerätään kaikki siihen liittyvät tiedot (van Deursen ym., 2000) ja tarvittavat tiedot sovellusalueesta, sen sanastosta ja käsitteistä. (Mernik ym., 2005). Mernikin ym. (2005) mukaan suunnitteluvaiheessa kieli voidaan perustaa jo olemassa olevan kielen pohjalta tai kehittää täysin itse. Sitten kieli toteutetaan suunnittelemalla semanttiset käsitteet implementoiva kirjasto sekä DSL-komennot kirjallisuuskutsuiksi kääntävä ohjelma ja kieltä aletaan käyttää ohjelmissa (van Deursen ym., 2000).

Käyttäjiltä ei välttämättä edellytetä ohjelmointitaitoja: jos mallintajat eivät ole ohjelmistonkehittäjiä, kielen visualisoinnista voi olla hyötyä (Luoma ym., 2004). Voitaisiinkin ajatella, että semantiikan, syntaksin ja pragmatiikan huomiointi olisi erityisen tärkeässä asemassa juuri sovellusaluekohtaisten kielten kehittämisessä.

Sovellusaluekohtainen mallintaminen voi mahdollistaa niin sanotun low-code- tai no-code-kehityksen. Tällä viitataan ohjelmistokomponenttien generoinnissa käytettyihin alustoihin tai lähestymistapoihin, joissa liikutaan koodia ylemmällä abstraktiotasolla, eikä käyttäjä välttämättä näe koodia lainkaan. (Curty, Härer & Fill, 2023.) Silloin sovellusaluekohtaiset mallit helpottavat ohjelmistoartefaktien ymmärtämistä, sillä mallia on helpompi tulkita kuin koodia. Lisäksi mallien avulla voidaan huolehtia tarkastuksista ja luoda ohjelmistoartefakteja automaattisesti. (Curty ym., 2023.) Myös osa haastatelluista viittasi tähän mahdollisuuteen haastatteluissa, joita käsitellään empiirisessä osiossa.

Koodin generoinnista tarjoutuu mahdollisuuksia tekoälyn hyödyntämiseen, ja kielimallit tuovat uuden näkökulman lingvistiikkaan. Seuraavassa aliluvussa käsitellään tekoälyn hyödyntämistä tietojärjestelmien mallintamisessa.

3.5 Tekoäly ja kielimallit

Koneoppimisen sovelluksissa on viime aikoina tapahtunut paljon edistystä. Tässä osiossa luodaan lyhyt katsaus tekoälyn ja suurten kielimallien kehitykseen sekä sen mahdollisiin seuraamuksiin tietojärjestelmien mallintamisen ja koodin generoinnin alalla.

Generatiivinen tekoäly pystyy luomaan eli generoimaan sisältöä analysoimalla sen koulutukseen käytettyä dataa (Ooi ym., 2023). Tekoälyn luoma sisältö on aidontuntuista, ja kieli muistuttaa luonnollista ihmisen käyttämää kieltä (Liu ym., 2023).

Ooi ym. (2023) kirjoittavat generatiivisen, suuria kielimalleja käyttävän tekoälyn kehittyneen perinteisistä koneoppimisen muodoista – ohjattu oppiminen, ohjaamaton ja puoliohjattu oppiminen. Aluksi käytössä olleista N-grammeista on siirrytty transformer-pohjaiseen arkkitehtuuriin. Transformer-malli on puolestaan neuroverkkopohjainen arkkitehtuuri datan käsittelyyn ja generointiin. (Ooi ym., 2023.)

Esimerkki generatiivisen tekoälyn sovelluksista on OpenAI:n marraskuussa 2022 julkaisema ChatGPT-keskustelubotti. Se pystyy syntetisoimaan uutta tietoa, ja sen tuottama sisältö muistuttaa ihmisten välistä kommunikaatiota. (Ooi ym., 2023). Muita Ooin ym. (2023) mainitsemista vastaavista palveluista ovat Googlen Bard, Microsoftin 365 Copilot ja Anthropicin Claude. Tekstin lisäksi generatiivinen tekoäly voi olla vaikkapa kuvia luova DALL-E, audiota tuottava AudioLM, videosisältöön keskittyvä Imagen Video tai audiotiedoston transkriptioon tarkoitettu Whisper. (Ooi ym., 2023.)

Liu ym. (2023) ehdottavat ChatGPT:n käyttöä esimerkiksi koodin generointiin. Silloin tekoäly generoisi koodipätkiä kuvailun tai määritysten perus-

teella, jolloin säästetään aikaa ja vähennetään manuaalisesta työstä aiheutuvien virheiden vaaraa (Liu ym., 2023). Rajauksina Liu ym. (2023) toteavat, että menetelmä soveltuu toistaiseksi vain joillekin ohjelmointikielille, manuaalista työtä tarvitaan edelleen, ja lopputulos riippuu luonnollisella kielellä kirjoitetun kehotteen laadusta.

Tekniikan kehitys mahdollistaa luonnollisen kielen käytön ohjelmoinnissa. Shinin ja Namin (2021) mukaan tästä voitaisiin saada useita etuja: ohjelmointikielen opettelu voi olla työlästä, ihmisiltä saattaa puuttua abstraktin hahmotuksen kyky tai tarvittavat matemaattiset taidot, tai ohjelmointikielen koetaan rajoittavan ilmaisuja. Ohjelmointiin siirtymistä voisikin helpottaa, jos järjestelmä tuottaisi koodin luonnollisen kielen pohjalta. (Shin & Nam, 2021) Tässä pitää huomioida lingvistiikkaan liittyvät seikat, joita käsitellään myös tässä tutkielmassa: kieli voi yhtä paljon helpottaa kuin haitatakin kommunikointia, ellei siihen kiinnitetä huomiota.

Joitakin Dehaernen, Deyn, de Gendtin ja Meertin (2022) mainitsemissa koneoppimisen hyödyntämiskeinoja ovat ohjelmointikoodin tuottaminen luonnollisen kielen komennoista tai kuvasta, dokumentaation generointi koodista sekä ohjelmointikielten kääntämisen toiselle. Usein käytettyjä koneoppimisen menetelmiä ovat takaisinkytketyt neuroverkot (RNN), transformer-teknologia ja konvoluutioneuroverkot (CNN). (Dehaerne ym., 2022).

Generatiiviseen tekoälyyn liittyy myös ratkaistavia ongelmia. Näistä Ooi ym. (2023) mainitsevat muun muassa hallusinoinnin - tekoäly tuottaa uskottavalta kuulostavaa, mutta täysin keksittyä sisältöä. Lisäksi tekoälyn toimintaan voivat aiheuttaa ongelmia koulutusmateriaalin puutteet ja vinoumat. (Ooi ym., 2023). Liu ym. (2023) huomauttavatkin, että järjestelmät eivät pysty tekemään faktantarkastusta, ja koulutusmateriaali on voinut tulla epäluotettavistakin lähteistä.

Huomioitavia seikkoja ovat lisäksi plagiointi, tekijänoikeudet, tietosuojaja yksityisyys (Ooi ym., 2024). Ooi ym. (2023) arvelevat, että tekoäly aiheuttaa muutoksia työpaikkoihin ja edellyttää uusien työtapojen opettelua. Lisäksi käyttäjän antamien oikeiden kehoitteiden merkitys korostuu, jotta saadaan haluttu lopputulos. Tutkijoiden mukaan tarvitaankin koulutusta ja parempaa läpinäkyvyyttä. (Ooi ym., 2023).

Tekoälyn kehitys voi myös lisätä tunneälyn merkitystä: kun kone ottaa hoitaakseen teknisen puolen, ihmisille jää tunteisiin perustuva vuorovaikutus (Luong, ym., 2023). Luongin ym. (2023) mukaan tästä voi seurata ihmisiin liittyviä ongelmia, jos työntekijöiden kompetenssi ei ole vaatimusten tasalla.

Kirjallisuuden perusteella vaikuttaakin siltä, että generatiivisella tekoälyllä ja kielimalleilla on hyödyntämispotentiaalia tietojärjestelmien mallintamisessa nyt ja etenkin tulevaisuudessa. Tekoäly voi ottaa hoitaakseen osan nykyisistä tehtävistä, ja ihmiset voivat keskittyä tunneälyyn ja humanistisen puolen huomiointiin. Tämä vaatii kuitenkin sopeutumista ja uusia taitoja, ja mahdolliset haitat ja ongelmat on huomioitava ja ratkaistava.

Tekoäly, koneoppiminen, neuroverkot ja kielimallit muodostavat laajan aihepiirin, jota ei voida tämän työn puitteissa käsitellä kovin yksityiskohtaisella

tasolla. Se kuitenkin kokoaa yhteen mallintamisen, kielimallien ja lingvististen lähtökohtien teemat ja tarjoaa mahdollisia näkymiä tulevaisuuden kehitykseen.

Tässä luvussa tutustuttiin mallinnuskielten historiaan, nykytilaan ja tekoälyn tarjoamiin tulevaisuuden mahdollisuuksiin. Seuraavaksi tarkastellaan, mitä lingvistisiä näkökulmia on käytetty tietojärjestelmien mallintamiseen.

4 KIELITEOREETTINEN TARKASTELU

Aiemmissa osioissa on tarkasteltu tietojärjestelmien mallintamista eri näkökulmista. Mallintamisessa käytetään kieliä niin luonnollisilla kielillä käytävässä kommunikaatiossa esimerkiksi määrittämissä vaiheissa kuin myös mallinnuskieltenkin muodossa. Mallintamista voidaan käsitellä myös lingvivistisistä lähtökohdista, joihin tutustutaan tarkemmin tässä luvussa.

Kieliteoreettisten näkökulmien tarkastelussa käytetään pohjana Lyytisen (1985) artikkelissaan *Implications of Theories of Language on Information Systems* esittämiä teorioita. On mahdollista, että jokin muukin jaottelu olisi hyödyllinen. Tämä lähestymistapa kuitenkin soveltuu kirjallisuuskatsauksen tarkoituksiin, sillä samoja näkökulmia on käytetty tarkastelluista artikkeleista myös esimerkiksi Byrerin ja Jelassin (1989) sekä Basdenin ja Kleinin (2008) kirjoituksissa.

Kirjallisuuskatsauksen jälkeen empiirisessä osiossa pyritään vielä selvittämään mallintamisen käytännön asiantuntijoiden haastatteluilla, ovatko nämä lingvistiset lähtökohdat jääneet teorian tasolle, onko niistä ollut hyötyä käytännössä ja mitä tulevaisuudelta voi mahdollisesti olla odotettavissa niin kieliteorioiden kuin mallintamisenkin suhteen. Ensin kuitenkin katsotaan, minkälaisia kieliteorioita on ehdotettu sovellettaviksi tietojärjestelmätieteeseen.

Lingvistikseen tarkastelussa noudatetaan pitkälti kielen lajittelua osa-alueisiin syntaksi, semantiikka ja pragmatiikka (Lyytinen, 1985, ks. myös luku 2.3). Beynon-Davies (2009) määrittelee osa-alueet seuraavasti: Pragmatiikassa keskitytään kommunikoinnin tarkoitukseen. Semantiikassa keskipisteenä on viesti, jota viestintätapahtumalla yritetään välittää. Syntaksilla puolestaan tarkoitetaan merkkejä, joilla viesti välitetään. Lisäksi empiriikan voidaan sanoa tutkivan fyysistä muotoa, jolla viesti välitetään tai tallennetaan. (Beynon-Davies, 2009). Osa-alueita voi kuitenkin käytännössä olla hankala pitää erillään toisistaan, sillä niiden piirteet sekoittuvat helposti toisiinsa eri kielissä tai kielenkäyttötilanteissa.

Kirjallisuudessa on käsitelty useita näkökulmia. Yleisimmin mainittuja ovat puheaktiteoria ja siihen osittain perustuva Language Action Perspective (LAP). Chomskyn ajatukset universaalista kieliopista soveltuvat tietojärjestelmien kehittämisen pohdintaan, mutta lingvistisen näkökulman ei tarvitse olla

pelkästään täysin kielitieteellinen, kuten käy ilmi Fregen matemaattis-loogisten ajatusten soveltamisesta tai Piaget'n kognitioon painottuvasta tai behaviorismiin ja Skinnerin näkemyksiin perustuvasta kielitieteestä. Näitä käydään läpi erillisissä aliluvuissaan. Lisäksi lopuksi luodaan katsaus eri artikkeleista löydettyihin muihin ehdotuksiin.

4.1 Puheaktiteoria

Puheaktiteoria keskittyy puhujan tarkoitusperiin. Siinä ajatuksena on, että puheella voidaan tehdä puheakteja eli puhetoimituksia. (Häkkinen, 1995, s. 77, 158). Puheaktiteoria ei tarkastelekaan niinkään varsinaisia sanoja vaan sitä, mitä puhuja on halunnut niillä saavuttaa (Basden & Klein, 2018). Pääpaino on sitoutumisessa johonkin tulevaisuuden toimintaan: "Tavataan huomenna". (Flores & Ludlow, 1976).

Puheaktiteorian peruserätyksenä on, että sanoilla saadaan aikaan asioita (Auramäki ym., 1992). Sen tausta on Austinin kirjoituksessa *How to Do Things With Words*, ja ajatuksia kehitti puheaktiteoriaksi Searle (Stamper ym., 2000). Teoriassa lajitellaan puheaktit kolmeen osaan: on *lokutionaarinen* eli varsinainen puheen tuottaminen, *illokutionaarinen* eli puhujan aikoma vaikutus sekä *perlokutionaarinen* puoli, joka kuvaa puhujan vaikutusta kuulijaan (Janson & Woo, 1995). Illokutionaarisella lausumalla on illokutionaarista voimaa, eli lupauksen lausuminen sitouttaa lausujan ja kastaminen antaa kohteelleen nimen (Koskensilta, 2016. s. 152).

Jonkinlaisen vaikutuksen aikaan saavat lausumat ovat perlokutionaarisia. "Jos esimerkiksi vakuuttaa toiselle jotain, tekee illokutionaarisen teon. Jos samalla vakuuttaa toisen asiasta eli onnistuu vakuuttamisessa, tekee myös perlokutionaarisen teon." (Koskensilta, 2016 s. 153). Auramäen ym. (1992) mukaan perlokuutio perustuu enemmänkin tunteisiin eikä niinkään sisälly puheaktiteoriaan. Sen sijaan illokutiiviset puheaktit voidaan lajitella assertiivisiin, kommissiivisiin, ekspressiivisiin, deklaratiiivisiin ja direktiivisiin seuraavan taulukon mukaisesti. Puheaktien lajittelu on koostettu seuraavan sivun taulukkoon (taulukko 2) on koostettu puheaktien lajittelu Abbasin ym. (2018) mukaan. Taulukossa on puheaktin tyyppi, kuvaus ja esimerkkejä käyttökohteista.

Puheaktiteoria huomioi myös kontekstin ja pyrkii siten tulkitsemaan puheaktia puhujan, ajan ja paikan perusteella (Auramäki ym., 1998). Lisäksi Auramäki ym. (1998) kuvailevat siihen sisältyvän "mahdollisen maailman" käsitteen – eli myös tulevaisuuden mahdollisuudet.

Nimestään huolimatta puheaktilla ei kuitenkaan viitata pelkkään puheeseen, vaan puheakteja tutkittaessa voidaan huomioida myös kirjallinen kieli ja symbolit, kuten myös kokonaiset lauseet tai pidemmät tekstit (Auramäki ym. 1992). Lisäksi Wand ym. (1995) katsovat, että puheaktiteorian avulla voidaan kuvata tarkasti vuorovaikutustilanteiden dynamiikkaa niin ihmisten kuin koneidenkin välillä. Näin puheakteja voidaan soveltaa myös tietojärjestelmiin ja muihin IT-alan teksteihin.

TAULUKKO 2 Puheaktien lajittelu, Abbasin ym. (2018) mukaan

Puheakti	Kuvaus	Esimerkkejä
Assertiivinen	Puhuja esittää faktoja todellisuudesta	Väittämät, jotka ovat totia tai epätotia
Kommissiivinen	Puhuja sitoutuu tekemään jotain tulevaisuudessa	Lupaukset, sopimukset
Ekspressiivinen	Puhuja ilmaisee tunteitaan tai asenteitaan	Onnittelu, kiittäminen, pahtoittelu
Deklaratiivinen	Puhuja aiheuttaa muutoksen maailmaan	Julistukset, tuomiot
Direktiivinen	Puhuja haluaa kuulijan tekävän jotain	Ehdotukset, ohjeet, kysymykset, toiveet

Puheaktien käsitteeseen perustuu myös Lyytisen (1985) artikkelissaan mainitsema *ordinary speaking* eli ns. säännönmukainen puhuminen, interaktionaalinen näkemys kielen ja toiminnan suhteesta. Sen mukaan luonnollisen kielen lauseet ovat puhe- tai sosiaalisia akteja (Lyytinen, 1985). Kuo ja Yin (2011) katsovat, että säännönmukaisen puhumisen teorian perusteella puhuminen on aina toimintaa, joten lingvististä rakennetta tutkimalla voidaan tutkia myös päätöksentekoprosesseja, normeja, vallan jakautumista ja muita sosiaalisia toimia. Lisäksi valtaa käytetään organisaatioissa usein kielen kautta, joten lingvistisen analyysin avulla voidaan päästä liiketoimintaratkaisujen ytimeen ja tutkia, kuka päättää, mitä ja miksi (Kuo & Yin, 2011).

Puheaktiteorialla on vahvasti sosiaalinen painotus: puheaktien merkitys muodostuu puhujan ja kuulijan yhteisessä sosiaalisessa kontekstissa, ja puheaktin ymmärtäminen edellyttää, että puheakti noudattaa sosiaalisia normeja. (Janson & Woo, 1995). Sosiaalinen aspekti korostuu etenkin, jos järjestelmää käytetään organisaatioissa, joten sosiaalisen puolen tarkastelusta on siten hyötyä myös tietojärjestelmien mallintamisen tapauksissa.

Byrer ja Jelassi (1989) katsovat, että jos järjestelmällä on lingvistinen funktio, järjestelmän kehittämisenkin pitäisi perustua lingvistiselle teorialle. Silloin lingvistiikan ymmärtäminen voi auttaa käyttäjän ja tietokoneen vuorovaikutuksen tutkimisessa (Byrer & Jelassi, 1989). Vuorovaikutusten tutkiminen puheaktien avulla voi myös auttaa tiedonkeruussa, vaatimusten määrittämisessä tai järjestelmän tehokkuuden analysointimenetelmien arvioinnissa (Janson & Woo, 1995).

Puheaktiteoriaa on kuitenkin kritisoitu siitä, että eri ihmiset voivat luokitella puheaktit eri tavoin. Vastaavasti kaikille puheakteille ei aina löydy sopivaa luokitusta. (Janson & Woo, 1996.) Pelkkä puheaktiteoria ei olekaan välttämättä yksioikoinen ratkaisu joka tilanteeseen.

Puheaktiteoriaan perustuu myös Language Action Perspective (LAP) -näkökulma, joka syntyi vastapainoksi tekniikkapainotteiselle ajattelulle. Siitä enemmän seuraavaksi.

4.2 Language Action Perspective

LAP (Language Action Perspective) -näkökulman juuret ovat 1980-luvun alussa. Ajatuksen esittivät alun perin Goldkuhl ja Lyytinen, mutta näkökulman perusteos oli Winogradin ja Floresin *”Understanding Computers and Cognition: A New Foundation for Design”* (Basden & Klein, 2008). Muita merkittäviä tutkimuksia ovat julkaisseet Flores ja Ludlow sekä Stamper (Lyytinen, 2004).

Lyytinen (2004) taustoittaa, että LAP kehitettiin eräänlaiseksi vastaliikkeeksi tietokantojen, järjestelmäsuunnittelun, tekoälyn ja tietojärjestelmien alalla silloin vallinneille teknologiaan painottuneille näkemyksille. Tarkoitus oli tutkia kielen sosiaalista puolta ja sen vaikutusta tietojärjestelmiin (Lyytinen, 2004).

LAP-näkökulma nojautuu puheaktiteoriaan. Se keskittyy etenkin pragmaatiikkaan ja tarkastelee kielen käyttöä tietyissä asiayhteyksissä käytännön tavoitteiden saavuttamiseksi. (Schoop, 2001). Erona puheaktiteoriaan on kuitenkin huomion kiinnittäminen kokonaisuun keskusteluihin ja sosiaaliseen toimintaan yksittäisten lausumien sijasta (Basden & Klein, 2008).

LAP korostaa tietokoneiden toimintaa viestinnän välineenä ja nostaa esiin lingvististen ja sosiaalisten sääntöjen/normien merkityksen kielen käytössä (Lyytinen, 2004). LAP-ajattelulla haluttiinkin pystyä tulkitsemaan kielen käyttöä sosiaalisissa yhteyksissä ja tutkia sen perusteella tietojärjestelmiä lingvistisinä ympäristöinä. Lyytisen (2004) mukaan näin voitaisiin hahmottaa kielen ja maailman vuorovaikutusta sekä kielen psykologisia ja sosiaalisia vaikutuksia.

LAP-näkökulmaa on käytetty myös menetelmien ja työkalujen kehitykseen (Schoop, 2001). Sitä on hyödynnetty käsitteellisessä mallintamisessa, organisaation kehityksessä, ihmisen ja tietokoneen vuorovaikutuksessa sekä tekoälysovelluksissa pohdittaessa, mitä kielen ymmärtämisellä tarkoitetaan (Lyytinen, 2004). Lisäksi LAP-ajattelun pohjalta on laadittu malleja ja tutkittu etnografisesti, miten kieltä käytetään organisaatioissa (esim. Schoop, 1999).

LAP-näkökulmaa hyödyntämällä on kehitetty esimerkiksi SAMPO-metodologia toimistojen vuorovaikutustilanteisiin. Tämä 1990-luvulla laadittu malli suhtautuu tietojärjestelmiin sosiaalisina ja lingvistisinä viestintäjärjestelminä ja painottaa sosiaalisen vuorovaikutuksen merkitystä. Mallin oli tarkoitus avustaa järjestelmien käytössä puheaktiteorian avulla, pääosin määritysvaiheessa. (Auramäki, Hirscheim & Lyytinen, 1992.) Myöhemmin haastatteluosiossa selvitetään, mitä suunnittelijat ajattelevat mallistaan nykyään.

Eniten LAP-näkökulmaa on kuitenkin sovellettu akateemisissa ympäristöissä eikä niinkään käytännön mallintamisen työssä, sen tarjoamista mahdollisuuksista huolimatta (Lyytinen, 2004). Lyytinen arvelee käytön vähäisyyden syyksi taloudellistakin puolta: uusien ratkaisujen omaksumisen pitäisi nimittäin olla kannattavaa. Hänen mukaansa LAP-näkökulman käyttämisestä pitäisikin

pystyä saamaan selkeämpää taloudellista hyötyä, jotta se kannattaisi omaksua riittävän laajamittaisesti (Lyytinen, 2004).

4.3 Frege

Kieliteoriassa voidaan tarkastella myös Fregen ajatuksia logiikasta ja totuusarvoista. Seuraavaksi pohditaan, miten ne mahdollisesti soveltuisivat tietojärjestelmätieteen tarkoituksiin.

Friedrich Ludwig Gottlob Frege (1848–1925) oli saksalainen matemaatikko, loogikko ja filosofi, ja hänen kielifilosofinen pääteoksensa on *„Über Sinn und Bedeutung“* vuodelta 1892 (Zalta, 2024). Fregeä kiinnosti matemaattisten symbolien merkitys ja hyödyntäminen kielitieteessä (Basden & Klein, 2008), ja hän yhdisti ontologiaa ja lingvistiikkaa merkityksen pohdintaan (Halttunen, 2019).

Frege pohti lauseiden totuusarvoja ja erotti toisistaan ns. **mielen** (*Sinn*) ja **merkityksen** (*Bedeutung*). Halttunen (2019) kuvailee, että *Bedeutung* on merkin tarkoite ja *Sinn* on subjektiivisen idean ja objektiivisen tarkoitteen välillä. *Ajattelu, kieli, merkitys* -teoksessa (1997) kuvaillaan erottelua seuraavasti: ”Merkkiin (nimeen, sanayhdistelmään, kirjoitusmerkkiin) liittyy sekä merkitty, jota voidaan kutsua merkin merkitykseksi että se, mitä nimittäisin merkin mieleksi.” (Frege, 1997, s. 42.) Mingers ja Willcocks (2014) yhdistävätkin Fregen jaottelun Peircen merkitysjakoon.

Fregen logiikassa ristiriidan muodostavat tilanteet, joissa useampi nimi viittaa samaan kohteeseen: usein käytetty esimerkki on ”aamutähti” ja ”iltatähti” – ne kumpikin viittaavat samaan taivaankappaleeseen, vain eri näkökulmasta (Zalda, 2024). Näin sanoilla on sama ”merkitys”, mutta eri ”mieli” (Frege, 1997, s. 42). Vastaavasti ristiriidan muodostaa tilanne, jossa lausuja uskoo asian olevan tosi, vaikka näin ei todellisuudessa olisi (Thomas, 2013).

Fregen kielifilosofiassa kielellä ilmaistaan asioita, jotka voidaan luokitella tosiksi tai epätosiksi (Lyytinen, 1985). Sen mukaan luonnollisen kielen lauseiden ja reaali maailman loogisesti tosien ja epätosien faktojen välillä on yhteys (Byrer & Jelassi, 1989). Näin kieli oli Fregelle loogisen päättelyn väline (Basden & Klein, 2008). Hänestä olisi siis mahdollista luoda formaali kieli, jossa jokaisella maailman faktalla on vastaavuus formaalin kielen rakenteessa, ja luonnollisen kielen lauseet voitaisiin kääntää näitä faktoja vastaavalle formaalille kielelle (Lyytinen, 1985). Lyytinen (1985) arvioi, että Fregen ajatuksia sovellettaessa nojaututaan semantiikkaan. Siinä kielen pääasiallinen funktio on asioiden nimeämisessä, mutta pragmaattista kontekstia ei niinkään huomioida (Lyytinen, 1985).

Burk (2024) on saanut Fregen mielen, merkityksen ja totuusarvojen käsitteistä välineitä suurten kielimallien olemuksen ja tekoälyn hallusinoinnin pohdintaan. Hänen analyysinsä mukaan tekoälyn hallusinoimaa sisältöä voi pitää epäsemanttisena. Syntaktisesti tällaisten todennäköisyyksiin perustuvien lauseiden ”merkitys” on oikein, mutta niiden ”mieli” ei viittaa mihinkään ulkoisen maailman tilaan, jota voisi pitää totena (Burk, 2024).

Fregen ajatuksia on hyödynnetty käyttäjän ja tietokoneen välisessä vuorovaikutuksessa mm. komentokielten käytön, nimeämisen ja sopivimpien lyhenneiden tutkimuksessa. (Byrer & Jelassi, 1989). Lisäksi Fregeä on sovellettu tietokannan ja organisaation välisten viittaussuhteiden tutkimiseen. Hänen ajatuksiin voidaan hyödyntää tietokantojen hallinnan, tietojen nimeämisen ja ylläpidon tehtävissä (Lyytinen, 1985). Sen sijaan vahvasti totuusarvoihin perustuvia näkemyksiä voi olla vaikea soveltaa fiktion ja synonyymeihin (Basden & Klein, 2008).

Thomas (2013) on hyödyntänyt Fregen näkemyksiä kielestä, logiikasta, mielestä ja merkityksestä tutkimuksissaan sosiaalisen arvioinnin muodostumisesta tietojärjestelmissä. Hän esittää, että Fregen kielen ja logiikan käsittelyllä on ollut suuri merkitys tietojenkäsittelytieteelle (Thomas, 2013). Suoraan Fregen ajatuksia käsitteleviä tutkimuksia vaikuttaisi kuitenkin löytyvän eniten logiikan ja filosofian aloilta.

4.4 Skinner

Behaviorismi on psykologian haara, joka tutkii käyttäytymistä (Alvarez, 2021). Vuosina 1940–1990 elänyt B.F. Skinner edusti niin sanottua radikaalia behaviorismia. (Graham, 2023).

Skinnerin voidaan sanoa olleen yksi viime vuosisadan merkittävimmistä psykologeista, joskin nykyään hänen ajatustensa voidaan sanoa joutuneen alan valtavirran ulkopuolelle. (Alvarez, 2021). Lingvistiikkaan liittyviä ajatuksia on kuvattu etenkin hänen vuonna 1957 julkaistussa teoksessaan *Verbal Behaviour*. Näitä ajatuksia, niiden sovellutuksia sekä kritiikkiä käydään lyhyesti läpi jäljempänä.

Teoksessaan Skinner (1948) esitti, että perinteinen lingvistiikka ei ole kiinnittänyt tarpeeksi huomiota ihmisen käyttäytymiseen, josta lingvistiset muodot ovat syntyneet (s. 2). Hänestä olisikin psykologian tehtävä selvittää, mitä tapahtuu ihmisen puhuessa tai vastatessa puheeseen, sillä se on käyttäytymistieteen alaa (Skinner, 1948, s. 5).

Behaviorismin peruseriaatteisiin kuuluvat ärsyke, reaktio, vahvistaminen ja ehdollistuminen: jos rotta saa ruokaa (*vahvistaminen*) painamalla vipua valon (*ärsyke*) palaessa, se painaa todennäköisesti vipua (*reaktio*) ollessaan taas nälkäinen (Graham, 2023). Tämän ajattelutavan mukaan tieteen pitää rajoittaa havaittavissa oleviin ilmiöihin, eli välttää mielensisäisiä asioita (Enos, 2024). Skinneriläisittäin kieltä tutkitaankin behavioristisesti, eli sen suhteessa havaittavissa olevaan käyttäytymiseen (Byrer & Jelassi, 1989). Kielen tapauksessa verbaalinen tuotos on siis reaktio ärsykkeeseen ja havaittavissa oleva käyttäytymisen (Enos, 2024). Tämä käyttäytyminen kertoo lauseiden merkityksen, ja näiden ärsyke-reaktio-parien avulla merkitystä voidaan tutkia (Lyytinen, 1985).

Skinnerin teoriat muodostettiin usein laboratoriossa ehdollistamalla eläimiä palkinnoilla tai rangaistuksilla, ja saman mallin arveltiin soveltuvan ihmisiin (Enos, 2024). Myös kielen oppimisen Skinner katsoi Enosin mukaan (2024)

tapahtuvan ehdollistamalla, kun lapset reagoivat ärsykkeisiin oikealla kielellisellä vastineella ja saavat siitä palkinnon.

Lyytisen (1985) luokittelussa skinneriläinen ajattelu perustuu pragmatiikkaan, ja tutkimuksesta saatu tieto on psykologista. Skinnerin ajatukset voivat soveltua menetelmien kehittämiseen ja päätöksentekijän käyttäytymisen kuvaukseen. Sen avulla voidaan huomioida päätökseen vaikuttavat ympäristön muuttujat tai ennakoita käyttäjän reaktioita syötteisiin. (Lyytinen, 1985.) Jos siis behavioristien mukaan kieli on havaittavissa olevaa käyttäytymistä, tietojärjestelmän on tarjottava sellaisia ärsykejä, jotka aiheuttavat toivotun reaktion (Byrer & Jelassi, 1989).

Alvarez (2021) katsoo, että Skinnerin ajatuksista olisi hyötyä nykypäivään tuotuna esimerkiksi pelien ja sosiaalisen median tutkimuksessa. Ns. Skinnerin laatikko, jossa testattava eläin altistettiin ärsykeille ja sen käyttäytymistä tutkittiin, rinnastuu helposti vaikkapa nykypäivän mobiililaitteisiin tai internetin ja sosiaalisen median myötä koko nykymaailmaan (Alvarez, 2021).

Behavioristisen näkökulman avulla voisi tutkia päätöksentekotilanteen tehokkuutta, mutta pelkkään yksilön käyttäytymiseen keskittyvä teoria ei kuitenkaan voi huomioida järjestelmiä sosiaalisina kokonaisuuksina (Lyytinen, 1985). Aikaisemman käyttäytymisen tutkiminen ei myöskään välttämättä kerro, mitä käyttäjä oikeasti haluaisi tehdä seuraavalla kerralla, sillä käyttäjä saattaa esimerkiksi haluta nimenomaisesti muuttaa käyttäytymistään (Ekstrand & Willemssen, 2016).

Skinnerin kuuluisa kritisoija on ollut Chomsky, joka julkaisi teorian kriittikkinsä *A Review of B. F. Skinner's Verban Behavior* vuonna 1959. Kirjoituksessaan hän esittää vastaväitteet Skinnerin teeseihin, käsitteisiin ja termeihin (Chomsky, 1997). Chomskyn ajatuksista lisää seuraavassa aliluvussa.

4.5 Chomsky

Chomsky (1928–) on amerikkalainen lingvisti ja filosofi, jolla on ollut merkittävä vaikutus myös kognitiotieteen alalla (Enos, 2024). Kielitieteellisen teorian kannalta tärkeitä julkaisuja ovat olleet *Syntactic Structures* ja *Aspects of the Theory of Syntax* (Häkkinen, 1995., 73).

Lingvistiikan alalla Chomskyn tavoitteena oli löytää kaikkia ihmisten kieliä koskevat yleiset periaatteet – universaali kielioppi. Universaaliuden teorian mukaisesti Chomsky päätteli lingvistisen kompetenssin, kyvyn oppia ja käyttää kieltä, olevan ihmisen synnynnäinen ominaisuus. (Truex & Baskerville, 1998). Kieli ei olisi pelkkä puhuttu, viitottu tai kirjoitettu ulkoinen viestintäjärjestelmä vaan ihmisten sisäinen taipumus (Enos, 2024). Truexin ja Baskervillen (1998) mukaan syvärakenteet olisivat silloin abstrakteja lingvistisen tiedon rakenteita, joita ei voi suoraan havainnoida. Pintarakenne puolestaan muodostuisi syvärakenteesta kumpuavista säännöistä (Truex & Baskerville, 1998) tai parametrien määrittelystä, josta Chomsky myöhemmin puhui sääntöjen sijasta (Enos, 2024).

Kielen rakenteisiin liittyy myös teoria generatiivisesta transformaatiokieliopista. Häkkisen (1995, s. 73) määritelmän mukaan siinä ”merkityksen kannalta olennaiset osat” kuvataan abstraktilla esityksellä, minkä jälkeen rakenteeseen lisätään symboleja ja sovelletaan transformaatioita. Generatiivinen transformaatiokielioppi selitti, miten lauseilla voi olla sama syvärakenne mutta eri pintarakenne: niihin on sovellettu eri transformaatioita (Häkkinen, 1995, s. 73.) Tämän mallin avulla pitäisi pystyä yhdistämään luonnollisen kielen lauseet formaalisti määritettyyn kielioppiin siten, että lauseita voidaan jäsentää tai generoida (Lyytinen, 1985).

Enos (2024) kirjoittaa, että Chomskyn mukaan universaali kielioppi olisi lapsilla valmiina jo syntyessään. Heidän pitäisi vain oppia soveltamaan oman äidinkiellensä ominaispiirteitä – kyseisten parametrien arvoja. Chomskyn näkemyksen mukaan kielen kehittyminen vastaisi muiden fyysisten elinten kehittymistä lasten kasvaessa. Kielen synnynäisyyden ajatuksia on kuitenkin myös kritisoitu. (Enos, 2024.) Häkkinenkin (1997, s. 76) toteaa Chomskyn kielimallin abstraktisuuden aiheuttaneen ongelmia: Universaalien syvärakenteiden olisi pitänyt olla sovellettavissa kielestä riippumatta, mutta sen taustalta oli havaittavissa englantiin viittaavia ominaisuuksia. Lisäksi on vaikuttanut siltä, että transformaatioiden avulla syvärakenteita voi muodostella useita tietämättä, mikä on oikea.

Lyytisen (1985) jaottelun mukaan Chomskyn ajatusten soveltamisessa pääpaino on syntaksissa ja semantiikassa, ja näkökulma keskittyy kieleen ideoiden tasolla. Hän arvioi, että analyysissa hyödynnetään kirjallista merkitystä, merkityssuhteita ja syntaktisia rakenteita. Siten tietojärjestelmän voidaan sanoa koostuvan formaalista kieliopista sekä kielioppia noudattavista lauseista. Generatiivinen kielioppi voi tuoda näkökulmia tietokantoihin, mutta ei niinkään järjestelmien käytön tutkimiseen (Lyytinen, 1985).

Lyytisen (1985) mukaan teorian sovellus tietojärjestelmätieteen tutkimuksessa on ollut vähäinen – artikkelin kirjoitushetkellä sitä oli sovellettu tekoälyn luonnollisen kielen käyttökohteissa. Se voisi kuitenkin soveltua tilanteisiin, joissa syntaktiset rakenteet ovat tärkeitä. (Lyytinen, 1985.) Myös Byrer ja Jelassi (1989) katsovat käyttökohteiksi tekoälyn ja luonnollista kieltä hyödyntävät dialogit sekä jäsenyspuut, joilla voidaan määrittää sanojen kieliopilliset kategoriat. Lisäksi Chomskyn formaalien kielten teoriaa on sovellettu kielitieteen lisäksi etenkin tietojenkäsittelytieteessä (Enos, 2024).

Wand ja Weber (1995) ovat soveltaneet syvärakenteen ajatusta tietojärjestelmien rakenteeseen. Tutkijoiden mukaan tietojärjestelmien suunnittelussa on tarkoitus noudattaa jäljennettävän reaali maailman syvärakennetta. Truex ja Baskerville (1998) varoittavat kuitenkin Chomskyn monisyisten ajatusten liiallisesta oikomisesta, joka voi olla vaarana siirrettäessä teoreettisia käsitteitä alalta toiselle, esimerkiksi sovellettaessa syvärakenteen ajatusta toisissa asiayhteyksissä kuin mihin se on alun perin tarkoitettu. Uusien ajatusten tuominen muilta aloilta voi tuoda hyödyllisiä uusia tuulia, mutta voi edellyttää liiallista yksinkertaistamista (Truex & Baskerville, 1998). Tämä onkin hyvä huomioida myös pohdittaessa lingvistiikan ja tietojärjestelmätieteen yhtymäkohtia.

4.6 Piaget

Jean Piaget (1896–1980) keskittyi tutkimuksissaan etenkin lasten kognitiiviseen kehitykseen ja siihen luokittelemaansa neljään kehitysvaiheeseen, ja hänellä on ollut merkittävä panos kehityspsykologiassa (Sanghvi, 2020). Hänen ajatuksiinsa perustuva kieliteoria keskittyy kognitiivisiin rakenteisiin ja prosesseihin, ja kieltä pidetään psykologisena faktana sekä kognitiota sen pääasiallisena funktiona (Lyytinen, 1985). Seuraavaksi selvitetään, millä tavoin Piaget'n ajatuksiin perustuvaa kognitiivista näkökulmaa voisi hyödyntää tietojärjestelmien mallintamisessa.

Piaget'n teorioissa olennaisessa osassa ovat skeemat. Sanghvi (2020) kuvaa niitä tiedon järjestelyn ja tulkinnan kehyksiksi, joiden avulla maailmaa ymmärretään. Uuden asian kohdatessaan lapsi yrittää tulkita sen aiemmin oppimiensa skeemojen perusteella ja pitää siksi esimerkiksi kaikkia nelijalkaisia karvaisia eläimiä koirina, kunnes hän saa tietää kissoista. Uuden tiedon kohdalla tapahtuu tällöin **adaptaatio**: tieto **assimiloidaan** mukauttamalla se aikaisempiin käsitelmiin tai **akkomodaatio**, jolloin vanhoja skeemoja muokataan tai ne vaihdetaan uusiin uuden tiedon perusteella. (Sanghvi, 2020.)

Rico Carranza, Huang ja Besems (2023) määrittävät, että skeema on tietoista ja kokemuksista muodostuva kognitiivinen malli, joka toimii muistin ja tiedon pohjana. Se voi toimia tiedon järjestämiseen, yleistämiseen, muokkaamiseen ja luomiseen käytettävänä järjestelmänä. Henkilö pystyy tunnistamaan kohteen vasta, kun kognitiivinen rakenne on tunnistanut sen. (Rico Carranza ym., 2023.) Samoja mentaalisen maailman malleja voidaan hyödyntää tietojärjestelmien tutkimuksessa.

Piaget'n näkökulma ei keskity varsinaisesti kieleen vaan kognitioon, mutta sitä kautta se koskee myös kieltä (Basden & Klein, 2008). Teoria liittyy läheisesti kognitiiviseen lingvistiikkaan: Piaget'ta voidaan myös pitää nykyisen kognitiivisen lingvistiikan edelläkävijänä (Sinha, 2010). Bjekovićin (2017) määritelmän mukaan kognitiivis-lingvistisestä näkökulmasta luonnollista kieltä tarkastellaan ihmisen yleisten kognitiivisten kykyjen osana (s. 86).

Piaget'n psykologisen tulkinnan mukaan ihminen nähdään kognitiivisena toimijana, joka samalla tuottaa kieltä (Basden & Klein, 2008). Kielentutkimus kiinnittää huomion psykologiseen todellisuuteen, käyttäjän kognitiivisiin prosesseihin, joiden kautta luonnollisen kielen lauseet muodostuvat (Lyytinen, 1985). Kielen avulla voidaan myös päästä käsiksi taustamotivaatioihin ja käyttäytymisen syvärakenteisiin (Byrer & Jelassi, 1989).

Byrer ja Jelassi (1989) arvioivat, että Piaget'n skeemoihin perustuvaa lähestymistapaa voidaan hyödyntää käyttäjien erilaisten ajattelumallien ja kognitiivisten tyylien huomioimisessa. Piaget'lle on löytynyt toteutuksia UX-puolelta etenkin lapsille suunnatuissa järjestelmissä (ks. Kaur, Kalid & Sugathan, 2021). Käyttäjien skeemat voidaan huomioida myös virtuaaliympäristöjen suunnittelussa luonnollisen kokemuksen varmistamiseksi (Richir ym., 2015).

4.7 Muita mahdollisia näkökulmia

Kirjallisuuskatsauksessa noudateltiin Lyytisen vuonna 1985 kirjoittamassa artikkelissa *Implications of Theories of Language for Information Systems* mainitsemia näkökulmia. Lyytinenkin toteaa kuitenkin artikkelissaan, että mikään yksittäinen näkökulma ei ole täydellinen saati soveltu joka tilanteeseen, vaan niistä mahdollisesti voidaan yhdistellä ja valita kulloinkin tarvittavia osa-alueita. (Lyytinen, 1985). Useissa kieliaihetta käsitelleissä artikkeleissa mainittiin kuitenkin lisäksi muitakin suuntauksia. Niihin luodaan tässä lyhyt katsaus.

Habermasin ajatukset: Habermasin ajatuksia käytetään myös LAP-teorian pohjana (Schoop, 2001). Schoop (2001) toteaa, että Habermas on puolestaan perustanut vuonna 1981 julkaisemansa *Theory of Communicative Action* -teorian Searlen ajatuksille ja hyödyntää samoja näkemyksiä puheakteista.

Habermas määrittelee kolme erilaista maailmaa: ulkoinen faktojen maailma, sisäinen subjektiivinen maailma ja kyseisen yhteisön yhteinen ulkoinen sosiaalinen maailma (Basden & Klein, 2008). Schoop (2001) määrittää viimeksi mainitun ”normatiiviseksi todellisuudeksi”, johon sisältyvät muun muassa arvot. Basden ja Klein (2008) luettelevat myös Habermasin määrittämiä sosiaalisia toimintoja: instrumentaalinen, strateginen, kommunikatiivinen, normatiivinen ja dramaturginen. Näihin Hirscheim ym. (1996) viittaavat ”orientaation” käsitteenä, jonka avulla voidaan perustella, miksi muutos pitää tehdä, mitä pitää muuttaa ja miten. Heidän mukaansa Habermasin kriittinen sosiaalinen teoria soveltuukin sosiaalisen toiminnan ja muutoksen tutkimiseen tietojärjestelmien kehittämisen yhteydessä (Hirscheim ym., 1996).

Wittgensteinin kielipeli: ”Kielipelillä” Wittgenstein viittasi tilanteeseen, jossa heimon kieli koostuu vain neljästä sanasta, ja samalla heimon koko todellisuus koostuu vain näistä neljästä asiasta (Dreiling, 2006). Samaa tilannetta voidaan Dreilingin (2006) mukaan laajentaa koskemaan koko yhteiskuntaa ja laajempaa kielijärjestelmää. Vastaavasti esimerkiksi tilausjärjestelmä ja sen muodostavat toiminnot ja säännöt voivat olla kielipeli (Lyytinen, 1985). Käytännön kokemukset ja tietoisuus eri kielipeleistä muodostaa todellisuuden, jota vasten ilmaisujen merkitykset muodostuvat (Hanseth & Monteiro, 1994).

Dooyeweerd (1894–1977) oli hollantilainen filosofi, joka kehitti kosmologista filosofiaa (Basden & Klein, 2008). Hän määritteli kosmonomisesta todellisuudesta 15 ”aspektia”, esimerkiksi lingvistisen aspektin. Kaikki ihmisen toiminta kattaa useampia аспекteja, mutta eri aspektit ovat tärkeitä eri tilanteissa. Kieli oli siis Dooyeweerdille monta aspektia sisältävä todellisuuden osa. (Basden & Klein, 2008).

Tutkitussa kirjallisuudessa mainitaan myös **Husserlin fenomenologia:** Hanseth & Monteiro (1994) yhdistävät Husserlin fenomenologian aiemmin käsitellyyn Fregen teoriaan. Fenomenologiassa kognitiivista, merkityksen aistimuksille antavaa rakennetta sanotaan *noemaksi*. Siihen liittyvät esimerkiksi havaittuun esineeseen liittyvät etukäteisoletukset ja viime kädessä se, millaiseksi koko maailma ymmärretään. (Hanseth & Monteiro, 1994). Hanseth & Monteiro

(1994) toteavat, että fenomenologian perusteella ei ole kehitetty mallinnustekniikoita, mutta teoriaa voitaisiin kuitenkin hyödyntää analysoimalla käyttäjien noemoja.

Yleisellä tasolla kirjallisuudessa on katsottu funktionaalisen lingvistiikan soveltuvan tietojärjestelmien mallintamisen tarkasteluun: Clarke ja Kautz (2014) toteavat, että funktionaaliset viestintäteoriat pitävät kieltä merkityksen muodostamisen välineenä ja ovat kiinnostuneita kielen käytännön sovelluksista. Funktionaalisen näkökulman mukaan käyttökonteksti on otettava huomioon lingvistisen rakenteen analyysissä (Bjeković, 2017 s. 31). Näin ollen se soveltuukin järjestelmänkehityksen tilanteisiin.

Tässä osiossa on käsitelty useita lingvistisiä näkemyksiä, joiden on kirjallisuudessa esitetty voivan soveltua käyttöön tietojärjestelmien mallintamisesta. Kielinäkemyksissä on myös paljon yhtäläisyyksiä, ja usein näkökulmat perustuvat toistensa ajatuksille ja kehittävät niitä eteenpäin. Näin ollen Basdenin ja Kleinin (2008) mukaan puheaktiteoria perustuukin Husserlin fenomenologian ajatuksiin merkityksistä, Habermas muokkasi kielenkäytön sosiaalista puolta ja sitten LAP sovelsi Habermasin ajatuksia tietojärjestelmien kehitykseen.

Seuraavaksi kirjallisuuskatsauksessa esiin tulleista teemoista esitetään lyhyt yhteenveto.

4.8 Kirjallisuuskatsauksen yhteenveto

Tässä osiossa on käyty läpi tietojärjestelmien kehittämisen lingvistisiä näkökulmia käsittelevää kirjallisuutta. Luvussa on sukkellettu Chomskyn syvärakenteeseen, tutustuttu Searlen ja Austinin puheakteihin, katsottu maailmaa Piaget'n skeemojen läpi ja kurkistettu Skinnerin laatikkoon.

Pohjana on ollut Lyytisen vuoden 1985 artikkelissa *Implications of Theories of Language for Information Systems* esittämä viiden lingvistisen näkökulman luettelo. Samaa luokittelua on noudatettu myös Byrerin ja Jelassin vuoden 1989 artikkelissa *The Impact of Language Theories on DSS Dialog* sekä Basdenin ja Kleinin vuoden 2008 artikkelissa *Philosophy of Language Approach*. Näiden kieliteoreettisten näkökulmien peruspiirteet on koostettu seuraavan sivun taulukkoon (taulukko 3). Lisäksi käsitellyssä kirjallisuudessa tuli esiin joitakin muita mahdollisia kieliteoreettisia näkökulmia, joihin luotiin lyhyt katsaus lopuksi yhdessä aliluvussa.

Artikkelissa mainittujen lähteiden lisäksi on tehty lisähakuja, joilla on tarkasteltu teorioiden muita käyttökohteita ja kartoitettu mahdollisia uudempia tutkimuksia löytämiseksi. Lisäksi teorioiden esittelyjä on pohjustettu hakeamalla perustietoja luotettaviksi katsotuista, vertaisarvioituista internetlähteistä *Stanford Encyclopedia of Philosophy* ja *Internet Encyclopedia of Philosophy*.

TAULUKKO 3 Kieliteoreettisten näkökulmien peruspiirteet. Byrerin ja Jelassin (1989) mukaan.

Näkökulma	Pohjateoria	Kielen funktio	Tiedon luonne	Rakenne	Tutkijat
Frege	Lingvistinen strukturalismi	Deskriptiivinen	Matemaattinen	Formaali logiikka, faktojen kar-toitus	L. Bloomfield E. Sapir G. Frege
Chomsky	Generatiivinen kielitiede	Generatiivinen	Käsitteellinen	Formaali kielioppi	N. Chomsky J. Lyons
Skinner	Behavioristinen psykologia	Behavioraalinen	Empiirinen	Toisto/vahvistus, ärsyke/reaktio	B. F. Skinner C. Morris
Piaget	Kognitiivinen psykologia	Kognitiivinen	Psykologinen	ei tärkeä	J. Piaget P. Lindsay
Puheakti/LAP	Diskurssikielitiede	Interaktionaalinen	Sääntöpohjainen	Edistää sosiaalista vuorovaikutusta	J. L. Austin J. R. Searle

Yleisesti ottaen kielitieteen ja tietojärjestelmätieteen yhtäläisyyksistä on korostunut järjestelmien toiminta kommunikaation välineenä sosiaalisessa kontekstissa. Tämä näkökulma soveltuu etenkin IT-järjestelmien yleisiin käyttötilanteisiin organisaatioissa, joissa sekä kieli että malli mahdollistavat viestinnän ja yhteisymmärryksen projektin eri osapuolten kesken.

Lisäksi esiin nousi ajatus merkityksen luomisesta ja muodostumisesta reaali maailman ja sen tulkitsijan välillä. Tätä tapahtuu sekä kielenkäyttötilanteissa että järjestelmän mallintamisessa. Kielen avulla koettu todellisuus voidaan tulkita, käytettiin siinä sitten mahdollisesti skeemaa, noemaa tai mallia, ja tarpeelliseksi katsottu tieto välittää muille.

Erilaisten lingvististen seikkojen huomioinnin järjestelmien mallintamisessa on katsottu voivan parantaa tietojärjestelmän käyttökokemusta, tehostaa järjestelmänkehitystä, helpottaa järjestelmän määrittelyn onnistumista sekä auttaa käyttäjän ja teknologian vuorovaikutuksen ymmärtämisessä ja parantamisessa. Näin voidaan saada parempia malleja ja sitä kautta parempia järjestelmiä.

Yhteenvedo artikkeleissa mainituista kieliteorioista on esitetty seuraavan sivun taulukossa (taulukko 4). Taulukossa on esitetty teorian näkemysten mukainen tietojärjestelmän määritelmä, sen sovelluskohde tai projektin vaihe sekä tutkimuskohteet, joihin teoriaa voi kirjallisuuden perusteella soveltaa.

Tämän yhteenvetoluvun taulukot on muokattu Lyytisen (1985) sekä Byrerin ja Jelassin (1989) artikkeleista, joissa LAP-näkökulman ja puheaktin tilalla esiintyy *ordinary speaking*. Samankaltaisuuksien vuoksi kyseiset tiedot on esitetty näissä taulukoissa yhdistettyinä.

TAULUKKO 4 Kieliteoreettisten näkemysten yhteenvetoa Lyytisen (1985) mukaan.

Näkökulma	Tietojärjestelmän määritelmä	Sovelluskohde/-vaihe	Tutkimuskohteet
Puheakti/LAP	Kommunikatiivisen toiminnan järjestelmä, organisaation sopimusten luonti/ylläpito, niiden tilan raportointi	Vaatimusanalyysi / organisaatiomuutoksen merkityksellistämisen	Organisaatioviestinnän mallit, sitoumukset, merkityksen muodostuminen
Frege	Formalisoitu tietojärjestelmä tallentaa, välittää ja käsittelee kohdealueeseen liittyviä tietoja	Vaatimusanalyysin vaihe	Kohdemaailman analyysi
Skinner	Tietojärjestelmä on tarkoitettu tiedon keruuseen, luokitteluun ja noutamiseen johtotehtävien hoitamisen yhteydessä	Tietovaatimusten analyysi / päätöksenteon analyysi	Reaktiot lingvistisiin ärsykkeisiin
Chomsky	Tietojärjestelmä on kokoelma kielipililla generoituja ajan myötä vaihtelevia lausekokonaisuuksia	Vaatimusanalyysi / luonnollisella kielellä laadittujen käyttöliittymien suunnittelu	Oikeiden lauseiden määrittely, merkittävien lauseiden määrittely
Piaget	Tietojärjestelmässä on lingvistisiä rakenteita, jotka käynnistävät kognitiivisia prosesseja päätöksenteon yhteydessä	Tietovaatimusten analyysi / tiedon ja kognitiivisen tyylin yhteensopivuus	Päätöksentekijän kognitiiviset skeemat ja tyyli, kognitiivinen kehitys

Moni kirjallisuuskatsauksen artikkeleista oli kirjoitettu ja niissä käsitellyt teoriat kehitetty jo joitakin vuosikymmeniä sitten. Näin ollen niiden ajatukset teorioiden soveltuvuudesta ja käyttökohteista pitää suhteuttaa ajalliseen viitekehykseen. Organisaatioissa käytetyt järjestelmät, ratkaisut ja teknologiat ovat muuttuneet ajan myötä, ja ihmisten suhde tietotekniikkaan on nykyään toinen. Samoja teorioita on toki voitu siirtyä soveltamaan uusiin kohteisiin sitä mukaa, kun organisaatioissa otetaan käyttöön uusia ratkaisuja. Toisinaan kuitenkin vanhat käytännöt unohtuvat muutoksen myötä. Näin on voinut käydä mallin-

tamisellekin ketterän kehityksen yleistyessä, mallintamisen kieliteorioista puhumattakaan.

Vastaavasti uudet tekoälysovellukset luonnollisen kielen käsittelyn käyttökohteineen avaavat uusia ovia ja yhteyksiä kielitieteen ja tietojärjestelmätieteen välille. Tulevaisuuden tarpeisiin voidaan tarvita uutta teoriaa, jota ei ole vielä kehitetty, mutta myös aiemmista teorioista voi löytyä uutta ja hyödyllistä näkökulmaa.

Kielitiedettä ja tietojärjestelmätiedettä yhdistelevät näkemykset saattavat perustua enemmän teoreettisiin oletuksiin hyötyjen mahdollisuuksista kuin käytännön kokemuksiin niistä. Mahdollisesti ajan myötä myös kirjoittajien näkemykset asioista ovat voineet muuttua. Empiirisessä osiossa onkin tarkoitus siirtyä tarkastelemaan käytännön näkökulmaa ja pohtia ajallista muutosta.

5 EMPIIRISEN TUTKIMUKSEN LÄHTÖKOHDAT JA TOTEUTUS

Tietojärjestelmien mallintamisesta ja kielitieteestä voidaan löytää useita yhtäläisyyksiä. Kumpaakin käytetään todellisuuden kuvaamisen ja sosiaalisen vuorovaikutuksen välineenä. Lisäksi kielet ovat mukana tietojärjestelmien suunnittelussa, toteutuksessa ja käyttötilanteissa. Tietojärjestelmien mallintamiseen onkin sovellettu useita lingvistisiä teorioita, joita käsiteltiin edellisessä luvussa. Aiheesta julkaistun kirjallisuuden tutkimisesta saattaa kuitenkin syntyä yksipuolisesti pelkkään teoriaan perustuva kuva. Kokonaiskuvaa täydennettiin vielä asiantuntijahaastatteluilla.

Tässä luvussa käydään läpi haastattelujen tavoitteet, esitellään teemahaastattelun ominaispiirteitä ja kerrotaan yleisellä tasolla haastateltavista sekä haastattelujen toteutuksesta. Tuloksia pohditaan tarkemmin seuraavassa luvussa.

5.1 Haastattelujen tavoitteet

Työhön valittiin kirjallisuuskatsauksen tueksi asiantuntijahaastattelut. Haastattelujen avulla oli tarkoitus saada uusia näkökulmia teorioihin ja saavuttaa menetelmätriangulaatio: "Menetelmätriangulaatio voi tarkoittaa kahta seikkaa: samaa menetelmää käytetään eri tilanteissa tai eri menetelmiä käytetään samassa tutkimuskohteessa" (Hirsjärvi & Hurme, 2022, s. 38). Tässä tutkielmassa triangulaatiolla on tarkoitus tarkastella samaa tutkimuskohdetta eri menetelmien avulla.

Tutkimukseen yhdisteltäviksi menetelmiksi valittiin kirjallisuusanalyysi ja haastattelut. "Menetelmävalintoja tehtäessä joudutaan ratkaisujen perusteena käyttämään sellaisia kriteerejä kuin tehokkuus, taloudellisuus, tarkkuus ja luotettavuus." (Hirsjärvi & Hurme, 2022, s. 32.) Tämän tutkielman yhteydessä menetelmien yhdistelmä parantaa etenkin luotettavuutta ja tarkkuutta. Tehokkuuden ja taloudellisuuden parantamiseksi haastattelut suoritettiin videohaastatteluina.

Kirjallisuusanalyysin tarkoituksena oli selvittää, miten lingvistisiä näkökulmia oli lähestytty kirjallisuudessa. Haastattelut puolestaan oli tarkoitettu kirjallisuusanalyysin täydentämiseen käytännön näkemyksillä, sillä pelkkä teoria olisi voinut antaa yksipuolisen ja vääristyneen kuvan todellisesta tilanteesta. Haastattelujen etuina on se, että haastateltava pääsee tuomaan asioita esille vapaasti, aihe voidaan sijoittaa laajempaan kontekstiin ja haastattelujen avulla voidaan syventää saatavia tietoja (Hirsjärvi & Hurme, 2022, s. 33). Nämä tavoitteet soveltuivat myös tämän tutkielman tarpeisiin. Tarkoitus oli sisällyttää kieli-teorioiden kontekstiin uusia näkökulmia sekä taustoittaa ja syventää teoriaosuudessa kerättyjä tietoja.

Samalla haastatteluissa voitiin tarkastella osin jo monen vuoden takaisissa kirjallisuuslähteissä esitettyjä näkemyksiä nykytilanteen valossa ja selvittää, minkälaisia muutoksia mallintamisen käytössä tietojärjestelmien kehittämisessä on mahdollisesti tapahtunut ajan myötä.

5.2 Teemahaastattelu

Käytetty haastattelutyyppi oli teemahaastattelu. Se on strukturoimattoman ja lomakehaastattelun välimuoto, joka kohdistuu tiettyihin teemoihin (Hirsjärvi & Hurme, 2022, s. 46–47). ”Teemahaastattelusta puuttuu strukturoidulle lomakehaastattelulle luonteenomainen kysymysten tarkka muoto ja järjestys, mutta se ei ole täysin vapaa niin kuin syvähaastattelu” (Hirsjärvi & Hurme, 2022, s. 48).

Alastalo, Åkerman ja Vaittinen (2017, s. 186) toteavat, että asiantuntija-haastattelut ovat usein ”teemahaastattelun eli puolistrukturoidun haastattelun muunnelma”. Valittu haastattelutyyppi mahdollisti aiheen käsittelyn melko vapaasti mutta kuitenkin siten, että tarvittavat teemat käytiin läpi.

Kaikkien haastateltavien kanssa käsiteltiin samoja teemoja, mutta kysymyksiä hieman vaihdeltiin sen mukaan, oliko kyseessä asiantuntija työn vai tutkimuksen kautta vai ehkä molempia. Tarkka sanamuoto, järjestys ja mahdolliset lisäkysymykset voivat myös hieman muuttua haastattelun etenemisen mukaan. Kuten Alastalo ym. (2017, s. 189) toteavat, asiantuntijahaastatteluissa on usein tarpeen räätälöidä kysymyksiä ja muokata runkoa. Vertailtavuuden vuoksi huolehdittiin, että samat asiat kuitenkin käsiteltiin kaikissa haastatteluissa.

5.3 Haastateltavat

Haastateltavat valikoitiin tutkielman ohjaajan avustuksella siten, että heillä oli joko käytännön mallinnuskokemusta tai tutkimustaustaa tai molempia. Kutsu lähetettiin sähköpostitse kuudelle haastateltavalle, jotka kaikki suostuivat.

Haastateltavia oli siis yhteensä kuusi. Lukumäärä oli melko suppea, mutta sen katsottiin olevan soveltuva tämän tutkimuksen tarkoituksiin, eli täydentä-

mään ja ajantasaistamaan kirjallisuuskatsauksesta välittyntä kuvaa. Hyvärinen (2017, s. 29) kirjoittaa haastattelujen määrästä: ”Laadullisen tutkimuksen tekijä ja ohjaaja joutuvat jatkuvasti pohtimaan, mikä on kulloinkin riittävä määrä haastatteluja. Kysymykseen ei ole yksiselitteistä ja yleispätevää vastausta.” Lopputulokseksi tuleekin ”riippuu tilanteesta” (Hyvärinen, 2017, s. 29). Yhtäältä sanotaan: ”Jos aineistossa on paljon laajoja haastatteluja, analyysin tarkkuus ja haastattelujen yhtäläinen hyödyntäminen väistämättä kärsivät” (s. 32). Pie-nempi määrä haastattelumateriaalia onkin helpompi huomioida tasapuolisesti.

Hirsjärvi ja Hurme (2022, s. 59–60) määrittävät: ”Haastattele niin monta kuin välttämätöntä, jotta saat tarvitsemasi tiedon”. Heidän mukaansa haastatteluvien määrä riippuu aina tutkimuksen tarkoituksesta. Tähän pro gradu -työhön soveltuu myös määritelmä: ”*harkinnanvaraisesta näytteestä*”, jolla pyritään syventämään ymmärrystä, saamaan tietoa tai uusia näkökulmia: ”Muutama henkilö haastatteleamalla voidaan saada merkittävää tietoa”. Mahdollisimman merkittävän tiedon saamiseksi haastatellut oli valittu huolellisesti taustansa perusteella.

Haastattelut olivat asiantuntijahaastatteluja. Aiheesta ovat kirjoittaneet Alastalo ym. (2017, s. 180): ”...asiantuntijat ovat haastattelujen kohteena erityinen ryhmä ja heidän haastattelemisessaan on tiettyjä erityispiirteitä.” Vaikka kyseessä on erityinen ryhmä, kyseessä ei ole heidän mukaansa kuitenkaan varsinaisesti erillinen haastattelumenetelmä.

Mitä ”asiantuntijoilla” sitten tarkoitetaan? (Alastalo ym., 2017) määrittelevät asiantuntijan henkilöksi, ”jolla on tietystä aihealueesta sellaista tietoa ja mahdollisesti sellaisia taitoja, joita maallikolla ei ole” (2017, s. 183) tai ”joilla on sellaista erityistä tietoa tutkittavasta asiasta, jota ei ole kenelläkään toisella tai jota on vain hyvin harvoilla” (2017, s. 183).

Tässä tapauksessa kaikilla haastateltavilla oli monen vuoden tausta erilaisen mallintamiseen liittyvien tehtävien ja/tai tutkimuksen parista. Taustojen perusteella oli syytä olettaa, että kaikilla valituilla henkilöillä voisi olla tutkielman kannalta hyödyllistä käytännön tietoa ja kokemuksia aiheesta.

Kooste haastateltavien taustoista:

- logististen toimintaketjujen mallintaminen, Petri-verkot ja kommunikaatiokanavat.
- toiminnan mallintaminen ja softaan liittyvä mallinnus, jossa generoidaan myös ohjelmakoodia
- konsulttina tekemässä mallinnuskieliä ja generaattoreita
- tutkimusta ja julkaisuja, teollisuusyhteistyötä, yritystoimintaa
- CASE-työkalut, omat mallinnuskielet, tuotekehittelyssä ja konsultoinnissa asiakkaille ja osittain teoreettistakin tutkimusta aiheesta
- tutkimustyötä
- metamallinnustutkimusta, monenlaista menetelmien mallinnusta ja menetelmien kehittämistä
- opettajana ja teollisuudessa tekemässä mallintamistyökalua, kaikissa töissä keskeistä ollut mallien teko.

Osa haastateltavista suhtautui varauksella nimensä käyttöön. He kokivat pystyvänsä vastaamaan vapaammin anonyymeina, etenkin kun valmista tutkielmaa ei ollut haastatteluhetkellä luettavana. Yhdenmukaisuuden ja tasapuolisuuden vuoksi muidenkin nimet on muutettu koodeiksi. Poikkeuksen muodostavat Lyytinen ja Auramäki, sillä heidän LAP- ja SAMPO-lähestymistapansa esiintyivät teoriaosiossa vahvasti, eikä heillä ollut myöskään mitään nimen käyttöä vastaan.

5.4 Haastattelukysymykset

Haastateltaville esitettiin seuraavat haastattelukysymykset:

1. Minkälaista tutkimus- tai mallinnustyötä olet tehnyt käytännössä?
 Minä vuosina / kuinka kauan?
 Oletko hyödyntänyt käytännön työssäsi mallintamisesta tehtyä tutkimusta?
2. (Jos aiemmin tutkimusta) Mitä mieltä olet aikaisemmista tutkimuksistasi nyt?
 Entä onko se edelleen ajankohtaista?
3. Kuinka paljon hyötyä nimenomaan lingvistisistä teorioista yleensä arvioit olleen käytännön työlle?
4. Millaiseksi arvelet ylipäätään käytännön ja tutkimustyön suhteen toisiinsa nykyään?
 Eli noudatetaanko tutkimusten suosituksia käytännössä, onko teorialla ja käytännöllä yhtymäkohtia vai ovatko ne vieraantuneet toisistaan?
5. Mallinnuskieliä kehitettiin paljon 90-luvulla, jolloin syntyi myös UML.
 Millaiseksi arvioit mallinnuskielten tulevaisuuden kehityksen?
 Väheneekö niiden käytännön soveltaminen, jääkö se vallan pois vai lisääntykö?
 Minkälaisia taitoja tarvittaisiin?
6. Miltä kannalta mielestäsi mallintamista tulisi vielä lähestyä tutkimuksissa?
 Olisiko kielitieteestä hyötyä vai tuleeko mieleesi jokin muu lähtökohta? Miksi?
 Vai olisiko tutkimuksen hyödyllisempää keskittyä muihin aiheisiin?
 Tuleeko mieleen jotain muuta lisättävää?

5.5 Toteutus

Haastattelut toteutettiin touko- ja kesäkuussa 2020 videoneuvotteluohjelmien Skype ja GoToMeeting välityksellä. Videohaastatteluihin päädyttiin sekä vallinneen koronavirusepidemian että maantieteellisten etäisyyksien vuoksi.

Videohaastattelujen käyttö soveltuu erityisesti asiantuntijahaastatteluihin, sillä ”Asiantuntijatehtävissä työskentelevät ihmiset ovat myös usein tottuneita videoneuvotteluihin, mikä tekee videoteknologiasta luontevan asiantuntija-haastattelun apuvälineen” (Alastalo ym., 2017, s. 195). Videototeutus myös helpotti haastatteluista sopimista ja sopivan ajan löytämistä, koska kenenkään ei pitänyt erikseen matkustaa minnekään.

Toisaalta etätoteutus saattoi vaikeuttaa haastatteluun orientoitumista, koska haastattelija ei ollut samassa tilassa. On mahdollista, että henkilökohtaisessa tapaamisessa esiin olisi saattanut tulla muita tai enemmän asioita. Henkilökohtaiset haastattelut olisivat kenties voineet kestää pidempään, sillä videoneuvottelu on helpompi lopettaa milloin tahansa yhteys katkaisemalla, ja toisinaan haastattelulle on ehkä raivattu vain lyhyt väli muiden kiireiden keskelle.

Haastattelun valinnassa menetelmäksi on huomioitava, että se ei ole ongelmatonta. Haastattelutilanteen ongelmallisia puolia kuvataan seuraavasti:

Haastattelu on tilanne, jossa kaksi toisilleen vierasta ihmistä tapaa haastattelijan aloitteesta satunnaisissa olosuhteissa ... Haastattelijan tehtäviin kuuluu olla utelias ja tehdä haastateltavalle tästä ehkä outoja kysymyksiä. Haastateltava havaitsee pian, että hän voi vastata lähes mitä vain, jopa valehdella, koska haastattelijaa vain nyökkäilee, hymyilee ja hymisee ja saattaa vaikuttaa ajoittain jopa ikävystyneeltä. (Hirsjärvi & Hurme, 2022, s. 43)

Tässä haastattelutilanteessa haastateltavat saivat valita sopivan tilanteen ja olosuhteen, joten häiriöitä ei siksi ollut paljon. Haastattelun aika oli joka tapauksessa pitänyt mahdollistaa muiden velvollisuuksien väliin, mikä saattoi vaikeuttaa tilanteeseen keskittymistä. Lisäksi etähaastattelussa osapuolet voivat jäädä etäisemmiksi toisilleen kuin henkilökohtaisessa tapaamisessa.

Haastatteluja käytetään tässä tutkielmassa pääosin lisänäkökulman tuomiseen kirjallisuusanalyysiin. Niiden avulla kerättiin käytännön kokemusta, joka ei käy ilmi teoriasta. On kuitenkin huomattava, että ”Haastattelu on kontekstija tilannesidonnaista. Tuloksiin sisältyy aina tulkintaa, ja tulosten yleistämistä on tarkkaan punnittava.” (Hirsjärvi & Hurme, 2022 s. 12.) Näin ollen haastatteluissa kerätyt näkemykset ovat yksittäisten henkilöiden näkemyksiä, eikä niitä välttämättä voi yleistää koskemaan muita asiantuntijoita tai koko tietojärjestelmien kehittämisen aluetta.

Haastattelujen kestot olivat 7:49–28:20 minuuttia. Kestot, ajankohdat ja haastattelujen olosuhteet on eritelty seuraavassa taulukossa.

Taulukko 5. Yhteenveto haastattelujen toteutuksesta

Haastattelu	Aika ja paikka	Kesto
Auramäki (H5)	18.5.2020 / Skype	15:51
Lyytinen (H6)	20.5.2020 / Skype	28:20
H1	20.5.2020 / Skype	16:27
H2	15.6.2020 / Skype	10:16
H3	9.6.2020 / Skype	7:49
H4	18.5.2020 / GoToMeeting	18:37

6 TUTKIMUSTULOKSET

Tämä pro gradu -tutkielma muodostuu luvussa 4 käsitellystä teoriaosuudesta sekä empiirisestä osiosta, jonka lähtökohdat ja toteutus kuvattiin edellisessä luvussa. Tässä luvussa esitellään empiirisen tutkimuksen keskeisimpiä tuloksia.

Kirjallisuuskatsauksessa käytiin läpi artikkeleja, joissa oli käsitelty kieli-teorioita etenkin tietojärjestelmätieteen viitekehyksessä. Niiden mukaan lingvististen lähtökohdien, esim. LAP-näkökulman, huomioinnilla voitaisiin saavuttaa monisyisempi käsitys symbolien ja todellisuuden vuorovaikutuksesta sekä hahmottaa paremmin kielen psykologisia ja sosiaalisia vaikutuksia (Lyytinen, 2004).

Teoriassa esiin tuodut näkemykset eivät kuitenkaan aina toteudu käytännössä, ja lisäksi näkemykset ja käytännöt muuttuvat usein ajan myötä. Siksi tilannetta kartoitettiin myös haastattelujen muodossa. Haastatteluissa selvitettiin kuuden asiantuntijan käytännön näkemyksiä teoriassa esiin tulleisiin aiheisiin sekä heidän kokemuksiaan alalla ajan myötä tapahtuneista muutoksista. Tulokset käytiin läpi ja käsitellyistä aiheista kerättiin esiin nousevia teemoja. Näitä käsitellään seuraavaksi.

6.1 Mallintamisen kannattavuus

Aluksi pitää korostaa, että niin tässä tutkielmassa kuin haastatteluissakin käsiteltävinä ovat käsitteelliset, tietojärjestelmien suunnitteluun käytettävät mallit eivätkä matemaattiset mallit. Lisäksi kielistä puhuttaessa tarkoitus on käsitellä mallinnus- eikä ohjelmointikieliä. Tämän vuoksi haastateltaviksi olikin valittu asiantuntijoita, joilla oli vuosien kokemus käytännön mallintamistyöstä tai aiheeseen liittyvästä tutkimuksesta. Heiltä kysyttiin tutkimusta pohjustavana kysymyksenä, kuinka paljon ylipäättään mallintamista käytetään ja koetaanko se kannattavaksi.

Tutkimusten mukaan mallintaminen kannattaisi ja parantaisi tuottavuutta, mutta silti mallintamista ei käytännössä aina käytetä (H4) tai käytetään harvoin.

Sen sijaan H2:n sanoin "aika paljon vain koodataan ja katsotaan mitä tulee". Tämän hän koki olevan sääli.

Yrityksillä todettiin olevan erilaisia lähtökohtia mallintamiseen: joissakin yrityksissä ns. citizen developer tekee koodeja low code- tai no code - ympäristöön mallintamalla ja generoimalla, mutta sitten taas moni perinteinen projektitalo "piirtää ruutupaperille jotain ja siinä on se kaikki mallinnus", ja "molemmat on omasta mielestään oikeassa" (H4). Myös yrityksen toimintakenttä vaikuttaa toimintatavan valintaan: jos asiakkaat vaihtuvat projektikohtaisesti, mallinnuksesta ei nähdä vastaavaa etuja, mutta "tuotetalo on todennäköisemmin mallintaja" (H2).

Joskus yritys voi huomata, että mallintaminen pakottaa toimimaan tavalla, joka ei itselle sovi. H1 mainitsi tilanteesta, jossa kehitys oli johtanut mallinnuskielen muokkaamiseen tai oman kielen keksimiseen. Jotkut mallinnusta kokeilevat puolestaan toteavat, ettei se sovi heille ja hylkäävät sen (H4). Mallinnuksen sopivuutta ja tarpeellisuutta pidettiin jonkin verran tapauskohtaisena. Tietämissä tehtävissä ne koettiin välttämättömiksi, toisissa taas ei.

Hyvin mallinnukseen katsottiin soveltuvan etenkin sellaisten tehtävien, joissa ei tarvita kovin paljon muutoskykyä ja joustavuutta. Tällaisesta mainittiin esimerkkinä tietokantasuunnittelu. (H3) Mallintaminen voisi soveltua myös tilanteisiin, joissa ei ole käytettävissä prototyyppisiä. Tällaisia järjestelmiä voi olla vaikea hahmottaa ilman malleja. (H2)

Malleilla siis arveltiin olevan paikkansa joissakin tilanteissa. Sen sijaan toisilla alueilla mallien ei katsottu olevan välttämättömiä, ja hyödyn sijaan malleista voikin aiheutua vain ylimääräistä työtä (H3). Malleista ei ole nimittäin hyötyä, ellei niitä pidetä ajan tasalla, mutta se vaatii oman vaivannäkönsä. Hankalia ovat etenkin tilanteet, joissa tapahtuu nopeita muutoksia, ja samalla myös toteutusta täytyy pitää ajan tasalla. (H3) Mallien ei koettu olevan useinkaan riittävän joustavia muuttuviin tarpeisiin nähden.

Käyttökelpoisiksi ei koettu ainakaan kovin tarkkoja mallinnuskieliä (H2). Hyötyjä tulisikin arvioida tilannekohtaisesti sen sijaan, että mallintamista pidettäisiin yleismaailmallisena ratkaisuna jokaiseen ongelmaan. H3 tiivistä tilanteen sanomalla: "Mallintamisella on oma käyttönsä. Mutta se on rajoittuneempi kuin luullaan." Ehkä automaattisen soveltamisen sijaan kannattaakin pohtia, soveltuuko mallintaminen juuri kyseiseen käyttökohteeseen – kuitenkin hylkäämättä sitä suoralta kädeltä.

Mallintamisen hylkäämisen syynä on usein siitä aiheutuva ylimääräinen vaiva, eikä aina ole mitenkään itsestään selvää, milloin vaivannäkö kannattaa. Tämän katsottiinkin olevan aihealue, josta tarvittaisiin lisää tutkimusta: millaisessa tilanteessa ja minkätyyppinen mallinnus tuo lisäarvoa, kuinka paljon työtä pitää tehdä etukäteen – ja kuinka paljon kärsii jälkikäteen, jos sitä työtä ei ole tehty (H6). Pitäisi siis selvittää, mikä mallinnuksessa on tehokasta ja toimivaa (H2) ja voitaisiinko siinä huomioida esimerkiksi liikettä. (H4)

Haastatteluissa katsottiin, että mallintamista ei aina käytetä, mutta sitä pitäisi käyttää ja järjestelmiä miettiä etukäteen enemmän kuin mitä käytännössä nykyään tapahtuu. Mallintamisen uskottiin voivan parantaa testausta ja suun-

nittelua. (H5) Mallit pakottavat pohtimaan järjestelmää ja ajattelemaan abstraktisti, mistä voi aiheutua lisää työtä kyseisellä hetkellä. Myöhemmissä vaiheissa malli voisi kuitenkin helpottaa työtä, vähentää kustannuksia (vrt. Jaakkola ym., 2016) ja parantaa lopputulosta.

6.2 Teorian ja käytännön ristiriita

Haastateltavilta kysyttiin myös kokemuksia teorian ja käytännön suhteesta. Näin voitaisiin kartoittaa, missä määrin kirjallisuudessa esiin tulevien näkemysten voidaan olettaa toteutuvan käytännössä. Mallintamisen käyttämisessä vaikuttikin vallitsevan kuilu käytännön ja teorian välillä, ja niiden katsottiin vieraantuneen toisistaan.

Haastateltavat kertoivat, että tutkijat eivät tutki käytännön kannalta tarpeellisia asioita vaan keskittyvät teoreettisempiin pohdintoihin kuin mitä käytännön tarpeet edellyttäisivät. (H1) Tähän vaikuttaa myös open source -kehityksen ja yrityksen omien ohjelmien välinen ristiriita. H1 arvioi, että yliopistoissa valitaan mieluusti maksuton open source -vaihtoehto, mutta yritykset mieluummin maksavat siitä, että voivat luottaa tukeen ja välineen olemassaoloon jatkossakin – ja IT-puolella muutenkin mielellään rakennetaan itse. Vastaavasti yritykset ovat haluttomia päästämään tutkijoita tutkimaan oikeaa työtä. Tämän vuoksi lähtökohdat ajautuvat erilleen. (H1)

Osa haastateltavista suhtautui kuitenkin positiivisemmin teorian tarjoamiin mahdollisuuksiin. H5 mainitsi bitcoinin esimerkiksi tilanteesta, jossa teoria ja käytäntö kulkevat käsi kädessä. H1 kertoi huomanneensa asiakkaita auttaessaan, miten kannattaa toimia käytännössä, ja tutkimuksesta on puolestaan oppinut, miten voi auttaa asiakkaita.

Kovin hyödyllistä ei ole kuitenkaan lukea teoriaa ja siirtyä soveltamaan sitä mekaanisesti mihin tahansa kohteeseen käytännössä. Sen sijaan teoria pitää liittää konkreettiseen ongelmaan: muussa tapauksessa ”siitä tulee tällainen aika turhanpäiväinen älyllinen shakki. Sitä voi harrastaa loputtomiin” (H6). Kannattaisikin mieluummin pohtia, mihin käytännön tarpeeseen kyseinen teoria parhaiten soveltuu tai vastavuoroisesti harkita, minkälainen teoria voisi auttaa ratkaisemaan tietyn käytännön ongelman.

Mallintamisen käytäntöjen moninaisuudesta, mallintamisen hyötyjen häilyväisyydestä eri tilanteissa sekä teorian ja käytännön ristiriidasta aiheutuu epävarmuutta, joka heijastui myös haastatteluissa. ”Voi olla että jollain alueilla on [teoria vieraantunut käytännöstä] ja voi olla että joillain tutkimusalueilla ei ole”. (H3.) Yksioikoista vastausta voi siksi olla vaikea antaa. Kirjallisuuskatsauksen ja haastatteluvastausten perusteella vaikuttaa kuitenkin siltä, että mallintamisen käytännön todellisuus ei kaikilta osin vastaa artikkeleissa esitettyjä näkemyksiä.

6.3 Kielitieteen näkökulma

Jos sitten mallintamista käytetään, kielitieteen hyödyntämisestä mallintamisessa oltiin sinänsä kiinnostuneita: "Uskon että siitä olisi hyötyä ja varmaan sitä käytetäänkin" (H5); "On henkilökohtaisesti mietityttänyt ja miellyttänyt kielitieteen näkökulma" (H4). Lingvistiikan koettiin vaikuttaneen taustalla omaan ajatteluun (H5) ja kielitieteen näkökulman alkaneen kiinnostaa työn myötä eri tavalla (H4). Lisäksi monen alan toimijan arveltiin kokevan kielet läheisiksi, vaikka varsinaista kielitiedettä olisikin käytännön työssä mukana aika vähän (H1).

Mallintamisen ja kielitieteen risteämispisteenä pidettiin kumpaankin olennaisesti liittyvää kommunikaatiota. H5:n mukaan kommunikaatiota tarvitaan erityisen paljon järjestelmien suunnittelun yhteydessä käyttäjien todellisten tarpeiden, reunaehtojen ja rajoitusten selvittämiseksi. Esimerkkinä hän mainitsee elektronisen kaupankäynnin ja terveydenhuollon päivystysjärjestelmät, joissa toiminnan pitäisi olla nopeaa ja tehokasta ja kyseessä on nimenomaisesti kommunikointi.

Kielten ja mallien ymmärtäminen koettiin rinnasteiseksi. Niiden katsottiin olevan samantyyppistä työtä, ja kielitieteestä arveltiin olevan hyötyä mallinnuksen tutkimuksessa. (H3.) Työn myötä lingvistiikasta oli kiinnostuttu uudella tavalla, ja kielitieteellistä näkökulmaa alettu arvostaa (H4). Kielen rakenteiden ymmärtämisestä katsottiin olevan hyötyä metamallintamisessa (H2). Hyödylliseksi koettiin ylipäätään ajatus siitä, että kyseessä on kommunikaatiojärjestelmä (H5).

Välittömiä hyötyjä kielitieteestäkään ei silti välttämättä nähty. Huomautettiin, että kyseessä on vain yksi näkökulma, "se on sitten mitä siitä saa irti" (H2). Lisäksi kognitiotieteestä ja lingvistiikasta saatava oppi "ei ole heti sovellettava välttämättä". Siihen tarvitaan molempia maailmoja ymmärtävä henkilö (H1.)

Kielitieteen eduksi H1 mainitsi sen, että kielitieteessä humanistinen puoli on huomioitu paremmin kuin tekniseen puoleen keskittyttäessä, ja järjestelmää käyttää kuitenkin ihminen. Se voi auttaa myös ymmärtämään, miksi järjestelmä onnistuu tai miksi se ei toimi niin kuin suunniteltiin (H6). H6:n mukaan kieliteoriat ovat "hyviä käsitteellisiä tapoja pyrkiä ymmärtämään tietojärjestelmiä ja tietojärjestelmien roolia organisaatiossa tai oikeastaan laajemminkin sosiaalisessa vuorovaikutuksessa". Silti hän arvelee, että kieliteoriasta saatava rajahyöty voi olla aika pieni. Kieliteoria voi kuitenkin auttaa kehyksenä, jos mallintaja sen avulla ymmärtää, miten tietojärjestelmän taustalla olevat kategoriat ovat syntyneet sosiaalisen konsultoinnin ja neuvottelun kautta. Näin voitaisiin hahmottaa järjestelmän kehittämisen vaikeuksia ja haasteita ja siten saada selville, miksi järjestelmä onnistuu tai mikä siinä on mennyt pieleen.

H6 mainitsee erityisesti suositus-, huutokauppa- yms. järjestelmät, joiden toteutuksessa on pragmaattisia ongelmia. Ihmisillä voi olla erilaisia käsityksiä siitä, mitä viestittiin ja missä tarkoituksessa. Tällöin hänen mukaansa kieliteori-

oista voi olla hyötyä. Niitä ei pidä kuitenkaan pitää vasarana, jolloin ”koko maailma näkyy yhtenä naulana”. (H6)

Edellä luvussa 4.1 on käsitelty puheaktiteoriaa ja siihen perustuvaa SAMPO-lähestymistapaa. Haastateltaviin oli valittu myös SAMPO-projektiin osallistunut Auramäki (H5). Hän mainitsi itsekin SAMPO-hankkeen ja taustoitti sen aikoinaan vallinneeseen ajattelutapaan, jonka mukaisesti tietokoneiden ajateltiin olevan vain laskentaa varten. Auramäki kuitenkin koki, että hankkeessa käytännössä saavutetut hyödyt eivät täysin vastanneet teorian tavoitteita: ”Se puheaktijuttu oli vähän sellaista, että me kehitettiin vähän niinku mallintamiskieltä siihen, mutta eihän sitä pirukaan rupee sellaisia kuvauksia tekemään”. Hän arveli vivahteiden paljouden tekevän menetelmän käytöstä liian mutkikasta. Sen sijaan mallintamisen pitäisi olla nopeampaa ja yksinkertaisempaa, jotta sitä jakseltaisiin tehdä.

Lyytinenkin (H6) totesi käsityksen SAMPO-metodologian ja kielimallien käytännöllisyydestä muuttuneen ajan myötä. Alun perin kuviteltiin, että SAMPO-lähestymistapaa voitaisiin käyttää käyttöjärjestelmiin tai uudentyypisiin sovelluksiin. Kuitenkaan hänen mukaansa ne eivät ole kovin laajasti käytössä missään mallinnusmenetelmissä tai käytännön sovellutuksissa.

Mallien käytön vähäisyyden syyksi Lyytinen (H6) arveli hyvin yksityiskohtaisen mallintaminen työläyden ja sen vaatiman investoinnin suhteessa järjestelmän toteuttamisessa saataviin hyötyihin. Hänen mukaansa moni 80- ja 90-luvuilla samantyyppistä työtä tehnyt on havainnut saman ilmiön. Kuitenkaan ajan kulumisesta huolimatta tietojärjestelmien pragmaattinen puoli ja toiminta vuorovaikutuksen säätelyssä ei ole muuttunut, ja sitä näkökulmaa ei pidä unohtaa. Ajallista muutosta katsotaan tarkemmin seuraavassa aliluvussa.

6.4 Ajallinen muutos

Käsitykset niin tietojärjestelmän luonteesta kuin eri teorioiden käytännöllisyydestäkin ovat muuttuneet ajan myötä. Kirjallisuuskatsauksessakin osoittautui, että käytetyt kielet ovat muuttuneet, UML muutti mallinnuksen kenttää, ja teknologia sekä toimistoissa käytettävät ohjelmat ja työn käytännöt ovat kehittyneet vuosien varrella. Lisäksi tulevaisuus tuo mukanaan uusia haasteita ja ilmiöitä. Siksi haastateltavia pyydettiin esittämään oma näkemyksensä aiheesta.

Haastatteluissa mainittiin alalla vallitsevan muuttuvia muoti-ilmiöitä, jotka tulevat ja menevät ja vaikuttavat esimerkiksi mallintamisen tutkimuksen suosioon. Toisinaan asioiden pitäisi kuitenkin ”palata taas muotiin, koska ne ovat tärkeitä.” (H2) Uuden teknologisen muutoksen myötä tietty aihe tulee kiinnostavaksi, kunnes innostus taas hälvenee. Toisinaan uuden menetelmän ajatellaan tekevän mallintaminen turhaksi. Tulee kuitenkin vaikeuksia, kun useita laajoja järjestelmiä pitää saada toimimaan yhteen, ja malleja tarvitaan taas. H4 viittasi 30 vuotta sitten tutkittuihin Expert Systems- eli asiantuntija ja

tekoälyjärjestelmiin, jotka unohdettiin vuosikymmeniksi, mutta nyt ne ovat taas tapetilla.

Käsi kädessä muoti-ilmioiden kanssa kulkee tutkimushankkeille myönnettävä rahoitus. EU ei esimerkiksi nykyään tue mallinnushankkeita, mutta 10–15 vuotta sitten oli toisin. (H4) Tutkimus ohjautuu sitten herkästi aiheisiin, joihin rahoitusta on saatavilla. Näin ollen mallintamisen tutkimuspäätösten taustalla voikin olla muitakin syitä kuin vain mallintamisesta teoriassa tai käytännössä saadut edut.

Tutkimusaiheen arveltiin olleen hyvin keskeinen ”joskus 80-luvun loppupuolella”, joskin sittemmin sen suosio on jäänyt – ”Mutta ehkä joku vielä palaa siihen lähiaikoina ja toteaa, että täällähän on peltoa kyntämättä”. (H3) Arveltiin, ettei aiheen tutkimuksesta saatu hyöty vastannut odotuksia, minkä vuoksi sen suosio on vähentynyt. Enää ei uskota mallintamisen ratkaisevan kaikkea, toisin kuin vielä 20–30 vuotta sitten (H3).

Asenteet ovatkin jonkin verran muuttuneet ajan myötä. Kuten H5 totesi SAMPO-lähestymistavasta: ”ei se ehkä ole tätä aikaa enää sitten ehkä”. Kaiken mallintaminen vaatii usein paljon työtä, ja mallit sekä niiden ajan tasalla pitäminen koetaan ylimääräiseksi huolettavaksi asiaksi, kuten edellä osiossa 6.1 on todettu.

Domain-spesifisten kielten (ks. luku 3.3) katsottiin olevan hyvä tutkimusaihe, sillä ne ovat parantaneet tuotavuutta enemmän kuin mikään vuosikymmeniin. Siirtyminen Assemblerista kolmannen sukupolven kieliin oli suuri harppaus, mutta sittemmin kielissä on ”poljettu paikallaan”. (H1)

UML kehitettiin alun perin yleiskieleksi (Giaglis, 2001). Sen on koettu ajan myötä kehittyneen yhä monimutkaisemmaksi, mutta samalla yleiskäyttöisyyden katsottiin heikentävän sen ilmaisukykyä (H2). Kieleen on lisätty ”tauhaa”, mikä nähtiin ongelmallisena. (H2). Sen sijaan sen kehityksessä ei ole huomioitu kieliteoreettisia näkökulmia, ja H1 mainitseekin kielen kehittäjien harmitelleen asiaa. Esimerkiksi Physics of Notation -periaatteilla lopputuloksesta olisi voinut tulla parempi. Huomioimalla myös kieliteoreettisia näkökulmia olisi voitu parantaa ilmaisuvoimaa ja siten UML:n käyttökelpoisuutta.

Tulevaisuudessa UML:n arveltiin pysyvän omana yleisenä kielenään, jota ”moni ymmärtää, mutta harva käyttää – oikein, sanotaan vaikka niin, tai kovin laajasti.” (H1). Kielestä arveltiin tulevan lingua franca -tyylinen väline. Uusien vastaavien kielten ilmestymiseen ei uskottu, vaan UML:n arveltiin aiheuttaneen ”allergiaa”. (H1.)

Ohjelmistokehityksen painopisteen arveltiin muuttuneen (H6). Kehittämisessä on siirrytty nopeaan kehittämiseen ja valmiiden resurssien ja kirjastojen hyödyntämiseen. H6 pohtikin, mihin mallinnusspesifikaatiota tarvitaan, jos semantiikka löytyy vaikkapa Azurelta nopeammin ja käytännöllisemmin.

Tähänastisen ajallisen muutoksen lisäksi haastateltavia pyydettiin myös esittämään ajatuksiaan tulevaisuudesta. Siltä osin useampi haastateltava arveli abstraktiotason nousevan ja siihen liittyvien abstraktisuustaitojen olevan tarpeen. Mallien merkitys saattaa korostua etenkin virtualisoitumisen myötä. ”Siihen nämä mallit on aika hyviä, jos ne vaan on oikeanlaisia.” (H2).

Kommunikaatiotaitoja uskottiin tarvittavan paljon myös tulevaisuudessa, jotta asiakkaan todelliset tarpeet ja reunaehdot voidaan selvittää (H5). Tässä kohdin korostuvatkin kielitieteen näkökulma ja mahdolliset kieliteorioiden tuomat edut.

Haastateltavista H3 oli skeptinen tulevaisuuden kehityksen suhteen. Hän ei ollut nähnyt tulevaisuuden kehitystä tai muutakaan kehitystä viime aikoina. H2 sen sijaan pohti, että mallinnusta käytettäneen myös tulevaisuudessa pohjana. Hän ennusti sovellusaluekohtaisten kielten pysyvän tarpeellisina. Niitä voidaan käyttää silloinkin, kun yleiskielen ilmaisuvoima ei riitä, sillä ne on juuri kehitetty parhaaksi mahdolliseksi kieleksi kyseiseen alueeseen.

Kuitenkaan kovin järeää mallintamiskalustoa ei enää uskottu tarvittavan (H6), sillä koko järjestelmän tarkka mallintaminen on tehotonta ja siitä saatavat hyödyt pieniä. Abstraktin mallintamisen H6 kuitenkin arvelee säilyvän tarpeellisenä sovellusten välillä ja organisaation tasolla.

H2 arveli, että mallintamista tarvitaan vielä tulevaisuudessakin jonkin aikaa, kun ihmiset viestivät keskenään, ”kunnes koneet tekee itseksensä kaikki”. Tulevaisuudessa mallintamista tekeväksi ammattiryhmäksi arveltiin kuitenkin muita kuin nykyisiä koodaajia – he eivät lähde mallintamaan, vaan isoin kasvu tapahtuu muiden ammattilaisten keskuudessa. ”Mallinnuksen kohderyhmät pitäisi tulla ja tulla muualta kuin niistä, jotka tänä päivänä naputtelee Javaa tai JavaScriptiä ruudulle”. (H4)

6.5 Haastattelujen yhteenveto

Haastatteluihin kutsuttiin sähköpostitse kuusi tietojärjestelmälän ammattilaisista, joilla oli pitkä ja monipuolinen käytännön kokemus mallintamisesta ja/tai sen tutkimuksesta. Taulukkoon 6 on koostettu yhteenveto haastateltavien taustoista, heidän mainitsemiaan kokemuksia hyödyistä sekä ajatuksia mallintamisesta yleensä.

TAULUKKO 6 Haastattelujen yhteenveto 1/2

Haastateltu	Tausta	Hyödyt	Yleistä
H1	Vuodesta 1992 tutkimusta, mallinnuskieliä, tuotekehittelyä ja konsultointia	Tutkimus ja työ olleet samaa ja tukeneet toisiaan.	Ihmisaspekti: usein fokus tekniikassa ja unohdetaan, että on ihminen käyttäjänä. Kielitieteessä se puoli on paremmin esillä.

(jatkuu)

Taulukko 6 (jatkuu)

H2	90-luvulla metamallinnus-tutkimusta, menetelmien mallinnusta ja kehittämistä 2012 saakka	Puheaktiteoria oli muotia taannoin, representaatioon perustuvaa tutkimusta	Säännöllisesti kuvitellaan, että on niin tehokas kehitysväline, ettei tarvitse mallintaa. Se aiheuttaa vaikeuksia, kun tehdään tarpeeksi suuria asioita.
H3	14 vuotta professorina, 9 vuotta opettajana, 5 teollisuudessa, mallinnustyökälun tekoa	Kumpi on tärkeämpää, että mallit kertoo mallit jotain reaali maailmasta vai kielenkäytöstä. Usein kommunikaatiossa kielenkäyttö on paljon tärkeämpää kuin yhtäpitävyys todellisuuden kanssa yhtäpitävyys.	"Mallintamisella on oma käyttösä. Mutta se on rajoittuneempi kuin luullaan."
H4	Kaupallisella tai soveltavalla puolella vuodesta 91/92 alkaen, konsultointia	On pystynyt elättämään itsensä	Mallinnus on vain työkalu, kuten kielikin. Ei itseisarvo
H5 (Auramäki)	Vuodesta 1978, tutkimusta, yritystoimintaa, SAMPO	On ohjannut ajattelua	"Pakkohan prosesseja on mallintaa ja miettiä, miten asiat kulkee"
H6 (Lyytinen)	80- ja 90-luvulta tutkimusta, LAP	Hyviä käsitteellisiä tapoja ymmärtää tietojärjestelmiä ja niiden roolia organisaatiossa	Enemmän tietoa tarvitaan siitä, milloin mallinnuksesta on todella hyötyä ja milloin ei.

Haastattelut järjestettiin etätoteutuksena, ja niissä käsitellyjä aiheita olivat mallintamisen käytännön ja teorian ristiriidat, tietojärjestelmien mallintaminen käytännössä, mallintamisesta saatavat hyödyt, kieliteorioiden hyödyntäminen sekä ajallinen muutos ja tulevaisuuden näkökulmat.

Mallintamisen hyötyjen pohdinta on monisyinen kysymys, joka osittain punoutuu muiden kysymysten ja otsikoiden alle. Yhtä ja kaikille joka tilanteessa soveltuvaa ratkaisua ei vaikuttanut olevan olemassa, ja vastauksia voi olla yhtä monta kuin tilanteita. Toisinaan yritykset ovat päätyneet kehittämään oman mallinnuskielen tai todenneet, että mallintaminen ei sovikaan heille. Jos yritys ei tällä hetkellä mallinnusta käytä, voi silti olla mahdollista, että siitä voisi jossain muodossa olla hyötyä juuri omassa toiminnassa.

Kieliteoreettisten näkökulmien lupaamia hyötyjä ei lopulta ajan myötä ole sovellettu käytännössä etenkin siinä mittakaavassa kuin kirjallisuuden perusteella olisi voinut arvella. Haastattelujen perusteella monet tämän tyyppistä työtä 80- ja 90-luvulla tehneet olivat joutuneet havaitsemaan, ettei näin tarkka mallintaminen tuonutkaan riittävästi hyötyjä järjestelmän toteuttamiseen. Nykyisin on puolestaan siirrytty valmiisiin resurssikirjastoihin eikä järeää mallintamista koeta yhtä hyödylliseksi. Kieliaihe tuntui silti läheiseltä ja kiinnostavalta: moni haastateltava suhtautui lingvistisiin näkökulmiin positiivisesti ja rinnasti luonnollisen kielen osaamisen mallinnuskieliin. Osa puolestaan katsoi kieliteorioiden johtavan vain filosofisiin pohdintoihin ja ”älylliseen shakkiin”.

Tulevaisuudessa katsottiin tarvittavan etenkin abstraktiokykyä. Mainittakoon, että haastattelut tehtiin ennen viimeaikaista tekoälyn kehitystä, joten sen mahdollinen vaikutus ei ole myöskään näkynyt vastauksissa. Sovellusaluekohtaisten kielten ja no code -ympäristöjen käytön arveltiin tai toivottiin lisääntyvän. Yleisesti ottaen alalla pintaan nousevat aiheet vaihtelevat teknologian kehityksen ja rahoituksen saatavuuden mutta osittain myös muoti-ilmiöiden mukaan. Tietyt asiat pysyvät kuitenkin ennallaan: esimerkiksi kommunikaation koettiin pysyvän tärkeässä osassa niin kauan kuin ihmisiä on mukana myös käyttäjinä ja koneet eivät tee kaikkea itsekseen. Kielitieteen hyötynä sen arveltiin pystyvän huomioimaan paremmin ihmisnäkökulman.

Yhteenvedo haastatteluissa käsitellyistä teemoista on koostettu taulukkoon 7.

TAULUKKO 7 Haastattelujen yhteenvedo 2/2

Haastateltu	Teoria	Kielinäkökohdat	Ajallinen muutos	Tulevaisuus
H1	Hankala saada tutkijat tutkimaan oikeaa käyttöä: yrityssalaisuudet, halu rakentaa itse.	Voi oppia paljon, muttei välttämättä suoraan sovellettavaa.	40-50 vuotta sitten mentiin eteenpäin Assemblerista, nyt on poljettu paikallaan kielissä, domain-spesifiset kielet lupaavia	Toivoo domain-spesifisten kielten käytön lisääntyvän

(jatkuu)

Taulukko 7 (jatkuu)

H2	Erikoistunutta, ei tekemisissä käytännön kanssa. Tutkijat innostuvat asioista, jotka eivät ole käyttökelpoisia käytännön ihmisten mielestä.	Tärkeä ymmärtää, mitä kielet tekee ja mitä niillä voi tehdä. Auttaa ymmärtämään kielen rakenteita.	Muutosta ollut pitkällä aikavälillä - ennen uskottiin UML:n ratkaisevan ongelmat, mutta ei niin käytnytkään. Nykyään enemmän koodataan ja katsotaan, mitä tulee.	Ehkä UML on pohjana, ja sovellusaluekohdaisilla kielillä voidaan tehdä sellaista, jota yleiskielillä ei.
H3	Jollain alueilla vieraantunut, joillakin ei.	Mallintaminen samanlaista kuin kielen ymmärtäminen.	Enää ei uskota, että mallintaminen ratkaisisi kaiken. Ei varmaankaan saatu odotettua hyötyä ja osoittautui aikaa vieväksi	Ei ole nähnyt kehitystä mallinnuskielten alueella.
H4	Tutkimuksen mukaan mallinnusta kannattaa tehdä, mutta käytännössä niin ei tehdä.	Ajan myötä alkanut arvostaa kielipillistä näkökulmaa	Alalla tulee usein teknologinen murros ja aihe tulee taas kiinnostavaksi isoille massoille.	Tulee uusia abstraktiotasoja, mallinnuksen muut kohderyhmitä kuin koodaajat, tarvitaan abstraktin hahmottamisen kykyä
H5 (Auramäki)	Arvelee joissakin aiheissa teorian ja käytännön kulkevan käsi kädessä	Teorioilla on merkitystä	Ei ehkä enää tätä aikaa	Kommunikointitaitoja tarvitaan
H6 (Lyytinen)	Teorioihin voi mennä syvällekin, mutta perusongelma voi olla käytännöllinen.	Lisätyöstä saatava rajahyöty pieni, jos vain mallintaa pragmaattisia Aspekteja.	Käyttötarkoitus, merkitys ja arvo voivat olla toisentyypisiä kuin alun perin kuvitettiin	Järeää mallinnuskalustoa ei ehkä tarvita, mutta yritysarkkitehtuurin mallinnus säilyy merkittävänä.

Haastateltujen ammattilaisten keskuudessa vaikutti vallitsevan jonkin verran epävarmuutta siitä, missä tilanteissa ja mihin tarkoituksiin mallintamisesta on hyötyä, minkälaista hyötyä se on, ja mitä ajan myötä tapahtuvat muutokset tuovat mukanaan. Toisinaan mallintaminen kannattaa, mutta aina ei,

kieliteorioista voi olla hyötyä joskus, mutta ei välttämättä, ja teoria tukee käytäntöä toisinaan, mutta on silti myös vieraantunut käytännöstä. Teoria vaikuttaakin tältä osin jonkin verran yksiselitteisemmältä kuin käytäntö.

7 POHDINTA

Tässä työssä on käyty läpi muutoksia, joita tietojärjestelmien mallintamisen kentällä on tapahtunut vuosien varrella. Keskenään kilpailevista kielistä siirryttiin lopulta UML-yleisstandardin käyttöön (Sommerville, 2011), ketterä kehitys muutti asenteita mallintamis pohjaista järjestelmänkehitystä kohtaan (Ambler, 2003), ja sovellusaluekohtaiset kielet (Mernik ym., 2005) ja tekoäly (Ooi ym., 2023) ovat omalta osaltaan muuttaneet käytäntöjä. Kukin menetelmä on omalla tavallaan pyrkinyt parantamaan tietojärjestelmien suunnittelun onnistumista ja tehokkuutta, tai ainakin esittänyt teoriassa niin tekevänsä. Tässä työssä tarkoitus on ollut tarkastella, mikä paikka kieliteorioilla on tällä kentällä – millainen tilanne on ollut ennen, millainen se on nyt ja mitä tulevaisuudessa on odotettavissa.

Teorian ajatukset eivät aina toteudu käytännössä. Empiirisessä osiossa olikin tarkoitus peilata haastatteluissa esiin tulevia käytännön näkemyksiä kirjallisuudessa esiin tulleisiin aihealueisiin. Haastatteluihin valittiin kuusi asiantuntijaa heidän käytännön mallintamiskokemuksensa perusteella. Käsitellyillä aihealueilla pyrittiin selvittämään, mikä asiantuntijoiden kokemus on teorian ja käytännön ristiriidasta, mallintamisen todellisuudesta, kieliteorioiden hyödyllisyydestä sekä tulevaisuuden mahdollisuuksista.

Tässä luvussa käydään läpi tutkimuksen tuloksia. Lisäksi pohditaan tulosten hyödynnettävyyttä ja luotettavuutta sekä esitetään aiheita jatkotutkimukseen.

7.1 Johtopäätökset

Tässä pro gradu -tutkielmassa on tarkasteltu teorian ja käytännön vuorovaikutusta ja ristiriitoja tietojärjestelmien mallintamisen lingvististen näkökulmien osalta, mallintamisen ajallista muutosta sekä tulevaisuuden mahdollisuuksia. Seuraavaksi käydään läpi työssä asetetut

tutkimuskysymykset ja pohditaan niihin vastauksia sekä teorian että haastattelujen perusteella.

- *Miten kieliteoreettisia näkökulmia on hyödynnetty tietojärjestelmien suunnittelussa sekä teoriassa että käytännössä?*

Kirjallisuudessa hyödynnettyjä kieliteoreettisia näkökulmia käytiin läpi osiossa 4. Etenkin puheakteihin ja LAP-näkökulmaan perustuvaa ajattelutapaa on hyödynnetty tutkimuksissa vuosien varrella esimerkiksi SAMPO-metodologian kehittämisessä (Auramäki ym., 1992) sosiaalisen median tutkimuksessa (Abbasi ym., 2018) ja terveydenhuoltoalan kommunikaation tutkimuksessa (Schoop, 1999). Lisäksi tietojärjestelmien kehittämiseen on sovellettu Fregen loogis-matemaattisia näkemyksiä ja *Sinn/Bedeutung*-jaottelua (Halttunen, 2019), behavioristista ajattelua Skinnerin tapaan (Alvarez, 2021), Chomskyn generatiivista kielitiedettä (Truex & Baskerville, 1998) sekä Piaget'n tutkimuksia kielestä kognitiivisena toimintana (Sinha, 2010). Nämä näkökulmat oli alun perin valikoitu tarkasteluun Lyytisen artikkelista *Implications of Theories of Language on Information Systems* (1985). Lisäksi kirjallisuudessa mainittiin Wittgensteinin (Dreiling, 2006), Habermasin (Schoop, 2001) ja Dooyeweerdin (Basden & Klein, 2008) kielifilosofiset pohdinnat sekä fenomenologia (Hanseth & Monteiro (1994). Funktionaalisen lingvistiikan on nähty yleisesti ottaen soveltuvan tietojärjestelmien kehittämiseen alan sosiaalis-pragmaattisen painotuksen vuoksi (Bjeković, 2017).

Empiiristä osiota varten haastateltiin kuutta asiantuntijaa. Kaikilla oli monivuotinen kokemus mallintamisesta joko akateemisesta maailmasta, yritysmaailmasta tai molemmista. Haastatellut ilmaisivat kieliteorioista olleen hyötyä joko ajattelua ohjaavana tekijänä, käsitteellisenä ymmärryksen välineenä tai osana käytännön työtään. Havaittavissa oli myönteisyyttä kieliä yleensä ja kieli-teorian tarjoamia mahdollisuuksia kohtaan, mutta mallintamista koskevan teorian toteutuksiin käytännössä suhtauduttiin osittain varauksella.

Haastattelujen perusteella mallintamista ei hyödynnetä käytännössä niin paljon kuin voisi kannattaa. Malli auttaisi ajattelemaan järjestelmää etukäteen ja saamaan kuvan järjestelmästä myös silloin, kun käyttäjä ei osaa lukea koodia. Toisaalta myös esitettiin, että täysimittainen mallintaminen ei välttämättä kannata joka tilanteessa, ja mallien ylläpitämisestä sekä muutoksiin reagoimisesta aiheutuu lisää työtä.

- *Mitä hyötyä kielitieteestä on ollut tietojärjestelmätieteelle?*

Epäonnistumiset ja haasteet ovat yleisiä ongelmia tietojärjestelmäprojekteissa (ks. Varajão ym., 2022; Nelson, 2007, McManus & Wood-Harper, 2008). Yhtenä syynä epäonnistumisiin on nähty viestinnän ongelmat (McManus & Wood-Harper, 2008). Ongelmia syntyy, kun käyttäjät puhuvat toimialakohtaisilla sanoilla, kehittäjät teknisillä (Bostrom, 1989). Tällaisessa tilanteessa mallintaminen voi auttaa, sillä mallin avulla järjestelmää voi ymmärtää myös osaamatta koodia. Malli helpottaa osapuolten välistä viestintää sekä mahdollistaa näkemysten ja tiedon jakamisen teknisten asiantuntijoiden ja muiden osapuolten välillä (Subahi & Alotaibi, 2018). Kieli- ja viestintänäkökulmien huomiointi mallintamisessa voi parantaa mallinnuksen

osa-alueita ja vaiheita, vaatimusten määrittelystä aina järjestelmän käyttöönottoon ja testaukseen asti.

Haastatellut asiantuntijat ovat kokeneet kielitieteen näkökulman kiinnostavaksi. Sen on arveltu vaikuttavan taustalla ja auttamaan hahmottamaan esimerkiksi metamallintamista. Kielitieteen todettiin silti olevan vain yksi näkökulma muiden joukossa: hyöty riippuu pikemminkin siitä, mitä siitä saa irti. Toisaalta sen on sanottu tuoneen uutta näkemystä aiemmin pelkkään laskentaan keskittyneelle alalle. Lisäksi kielitiede korostaa ihmisenäkökulmaa, joka saattaa jäädä huomiotta pelkkään tekniseen puoleen keskityttäessä. Järjestelmään lopulta kuitenkin käyttävät ihmiset.

Kieliteoriat voisivatkin mahdollistaa ihmisenäkökulman sekä järjestelmän sosiaalisen ja pragmaattisen aspektin huomioinnin (esim. Auramäki ym. 1992). Tästä voisi olisi hyötyä myös ketterän kehityksen projekteissa, joissa keskusteluilla ja palautteella on suuri merkitys, tai tekoälyn käytössä, sillä kone ei osaa esimerkiksi tunneälyä (ks. Luong ym., 2021.)

Haastatteluissa kieliteorioiden mainittiin toisaalta mahdollistaneen itsensä elättämisen, ja teorian ja käytännön nähtiin tukevan toisiaan ja kulkevan toisinaan myös käsi kädessä. Toisissa tapauksissa puolestaan teoreettisen ajattelun koettiin vieraantuneen käytännöstä ja käytännön käyttäjien tarpeista, ja kieliteorioiden tutkiminen koettiin turhan filosofisten aiheiden pohdinnaksi. Teorian ja käytännön välille syntyy juopa, jos akateemisessa maailmassa ei tutkita työelämässä hyödynnettäviä asioita, eikä työelämässä päästetä tutkijoita seuraamaan oikeaa työtä. Voikin pohtia, pitäisikö tutkijoiden mahdollisesti kiinnittää enemmän huomiota siihen, että tutkimuksesta on saatavissa käytännön hyötyä. Aina hyötyjä ei tietysti voi tietää etukäteen, kuten H3 totesi haastatteluissa: "mutta useinkin ne hyödyt ovat ennakoimattomia".

Vastauksista kuului varovaisen myönteinen asenne kieliteorioihin silloinkin, kun mallintamiseen itseensä suhtauduttiin epäillen: jos mallintaminen ylipäätään kannattaa, kieliaspektinkin huomiointi voi kannattaa, mutta aina mallintaminen ei kuitenkaan välttämättä kannata. Teorian hyödynnettävyys käytännössä ei riipukaan pelkästään kieliteorioista itsestään, vaan myös käyttökohdeesta ja sen tarpeista.

- *Voisiko joistakin vielä hyödyntämättömistä lingvistisistä teorioista olla hyötyä tietojärjestelmien suunnittelussa, vai kannattaako lingvistisistä lähtökohdista luopua ja keskittyä johonkin muuhun lähestymistapaan?*

Empiirisessä osiossa esille nousi kiinnostus kognitiivisen tieteen hyödyntämiseen, ja Cognitive Dimensions of Notation-, Physics of Notations -lähestymistavat mainittiin. Muita mainittuja mahdollisia lisätutkimuksen aiheita olivat domain-spesifiset kielet ja pohdinta siitä, minkätyyppinen mallinnus tuo lisäarvoa. Haastattelujen perusteella tutkimuksessa voisi selvittää, miten mallit voitaisiin saada helpommin omaksuttaviksi, luettaviksi tai muistettaviksi. Tähänkin kielitieteeseen perustuva tutkimus voisi tarjota välineitä.

Kielitiedekin kehittyi jatkuvasti, ja uusia näkökulmia nousee esiin. Etenkin funktionaaliset, pragmatiikkaan ja sosiaalisiin konteksteihin liittyvät teoriat voisivat soveltua tietojärjestelmien kehittämisen ja käytön tutkimiseen.

Tekoälyn ja kielimallien kehitys voi tarjota uusia mahdollisuuksia mallintamisen ja kielitieteellisten näkökulmien yhdistämiselle, kun tekoäly mahdollistaa esimerkiksi luonnollisen kielen käytön ohjelmoinnissa (ks. esim. Shin & Nam, 2021). Generatiivisen tekoälyn mukanaan tuomat ongelmat pitää kuitenkin huomioida (ks. Ooi ym., 2023, Liu, 2023). Samalla vastaan voi tulla uusia tutkimusaiheita ja ratkaisuja.

Lingvistikä ajattelulla on puolensa niin pitkään kuin ihmiset ovat mukana järjestelmiä tekemässä, käyttämässä ja viestimässä keskenään niiden avulla. Kieliteorioiden avulla voidaan esimerkiksi pyrkiä kartoittamaan, miten sana viittaa kohteeseensa todellisuudessa, mikä vastaa mallin ja reaali maailman järjestelmän välisiä viittaussuhteita (Hanseth & Monteiro, 1994). Haastatteluissa kävi kuitenkin ilmi, että kieliteorioiden yksityiskohtainen pohdinta voidaan kokea liian filosofiseksi ja käytännölle vieraaksi.

Haastatteluissa todettiin, että mallien käyttö voi ylipäätään osoittautua käytännössä liian hankalaksi. Ylimääräistä työtä ei haluta, joten malli ei saisi olla liian monimutkainen tai hankala. Mallin ylläpidosta ei saisi aiheutua liikaa vaivaa, jotta se koettaisiin kannattavaksi. Tutkimusta voitaisiinkin tarvita etenkin siitä, milloin mallinnusta kannattaa tehdä ja milloin ei.

Sen sijaan sovellusaluekohtaisten kielten käyttö voi lisääntyä tulevaisuudessa ja mahdollistaa koodin generoinnin sekä no code -ympäristöjen käytön. Haastateltavien mukaan koodaustaitoa ei välttämättä tarvita mallinnukseen, mutta abstraktin hahmotuksen kyky ja viestintätaidot ovat tarpeen. Nämä seikat kannattaa huomioida mahdollisia uusia lähestymistapoja mietittäessä.

Lisäksi tekoäly ja kielimallien kehitys voivat tuoda uutta näkökulmaa, jonka mahdollisuuksiin kannattaa perehtyä myös tietojärjestelmien kehittämisen puolella. Kieliteorioista voisi mahdollisesti löytyä uutta näkökulmaa myös uuden teknologian mukanaan tuomiin tilanteisiin. Kaikista näkökulmista ei siis kannata yksioikoisesti luopua, mutta pitää pohtia, mikä on järkevää missäkin kontekstissa.

Eri näkökulmien huomiointi voi parantaa mallinnuksen osa-alueita ja vaihteita, vaatimusten määrittelystä aina järjestelmän käyttöönottoon ja testaukseen asti. Mikään yksittäinen ratkaisu tai keino ei ehkä sovellu joka tilanteeseen, mutta eri vaihtoehtoihin tutustumalla tai niitä yhdistelemällä voisi kirjallisuuden perusteella löytää eri tilanteisiin ja tarpeisiin sopivia ratkaisuja. Osa niistä on kenties jo unohtunut ja tilalle on tullut uusia ratkaisuja, mutta toisinaan vanhoista keinoista voi löytyä ratkaisu myös uusiin ongelmiin.

7.2 Tulosten hyödynnettävyys ja luotettavuus

Tämän pro gradu -tutkielman tuloksia voidaan hyödyntää tietojärjestelmien mallintamisen tutkimuksessa sekä käytännön työssä tietojärjestelmien kehittämisessä organisaatioissa.

Työ keskittyy kieliteorioihin, joten aihetta on lähestytty teorian pohjalta. Näin ollen työn tuomat hyödyt ovatkin mahdollisesti parhaiten saavutettavissa

tutkimuksen kautta. Suurimpana hyötynä tutkielmassa nostetaan esiin aiemmin tehtyä ja mahdollisesti tietojärjestelmätieteen alalla jo unohtunutta tutkimusta kieliteorioiden hyödyntämisestä ja niiden tuomista eduista sekä koostetaan useiden artikkelien tiedot kirjallisuuskatsaukseen. Lisäksi tuloksista on hyöttyä teorian ja käytännön yhteensovittamisessa, sillä siinä tuodaan esiin tällä alueella ilmeneviä ongelmia.

Kirjallisuushakuun liittyy rajoituksia – työssä on käsitelty lukuisia aiheita, joista osasta on julkaistu useita tutkimuksia vuosikymmenten ajalta. Onkin mahdollista, että joitakin hyödyllisiä lähteitä on voinut jäädä seulomatta esiin. Tätä ongelmaa on pyritty ratkaisemaan tekemällä useita hakuja, kokeilemalla läheisiä hakutermejä sekä kohdistamalla hakuja eri vuosille. Lisäksi lähteitä on otettu käsiteltyjen artikkelien kirjallisuusluetteloista.

Alastalo ym. (2017, s. 195) toteavat asiantuntijahaastattelujen yhteydessä, että haastattelijankin on oltava valmis toimimaan asiantuntijana, ” kysymään täsmäntäviä kysymyksiä, kyseenalaistamaan ja kannustamaan haastateltavaa esittämään perusteluita”. Tältä osin tehokkaammin toimimalla haastatteluista olisi voitu saada enemmän irti. Tälläkin laajuudella haastateltavat pääsivät kuitenkin tuomaan esiin tärkeimmiksi katsomiaan asioita.

Luotettavuuden kannalta haastatteluissa oli melko suppea otanta, kuusi asiantuntijaa. Pienen haastateltavien määrän vuoksi tulokset saattavat vaihdella jonkin verran sen mukaan, keitä on pyydetty haastatteluun. Toisaalta haastateltavat oli valikoitu tarkasti siltä kannalta, että heillä voisi olla eniten annettavaa tutkimusaiheeseen.

Lisäksi voidaan todeta, että haastateltavien vastaukset olivat linjassa keskenään. Vastauksissa oli toki jonkin verran vaihteluakin, mutta yleisesti ottaen ne täydensivät ja tukivat toisiaan. Voidaan siis olettaa, että tuloksia voidaan laajentaa koskemaan myös suurempaa määrää haastateltavia.

Kaksiosaisen menetelmän valinta paransi tutkimuksen luotettavuutta. Pelkkä kirjallisuusanalyysi olisi antanut kielitieteellisten menetelmien käytöstä toisenlaisen kuvan kuin valittu kokonaisuus asiantuntijahaastatteluihin yhdistettynä. Nykyaikaan päivitetty haastattelukatsaus toi arvokasta tietoa teorian ja käytännön eroavaisuuksista ja mallinnuksen sovellettavuudesta.

Tutkielman tekoon tuli välissä joidenkin vuosien tauko, ja haastatteluja olisi voinut olla perusteltua saattaa ajan vielä tasalle ja täydentää useammalla henkilöllä. Toisaalta tulokset eivät olisi enää olleet täysin verrannollisia keskenään, ja se olisi jälleen omalta osaltaan viivästyttänyt tutkielman valmistumista. Haastattelujen lisääminen jätetäänkin jatkotutkimuksen varaan.

Haastatteluosiosta saadaan tietoa käytännön ja teorian eroavaisuuksista ja ammattilaisten tulevaisuutta koskevista ajatuksista. Niiden perusteella voidaan harkita, mitä aiheita kannattaa tutkimuksessa ottaa esiin. Voidaan myös pohtia, missä määrin voidaan realistisesti odottaa teorioissa esitettyjen näkemysten pätevän myös käytännössä.

Tuloksia voidaan mahdollisesti hyödyntää viestinnän tukena mallintamisen lisäksi myös järjestelmäsuunnittelun tilanteissa, joissa mallintamista ei

käytetä. Samoin niitä voidaan hyödyntää pohdittaessa, kannattaako mallintamista käyttää vai ei.

Kieliteorioiden esittely jää siltä osin pintapuoliseksi, että aiheista olisi tarjolla tietoa ja tutkimusta paljon enemmän kuin tässä pro gradu -työssä pystyy käsittelemään, ja rajausta on väistämättä tehtävä. Suuri osa kirjallisuudesta ei myöskään suoranaisesti liity tietojärjestelmien mallintamiseen, jos aiheena on vaikkapa lingvistiikka, filosofia tai psykologia. Näitä kaikkia voidaan toki tarvita tietojärjestelmien mallintamisessa, mutta jokin yhtymäkohta silti tarvitaan. Työn tarkoitus olikin esitellä siihen mahdollisia vaihtoehtoja.

7.3 Jatkotutkimus

Tässä tutkielmassa on käyty läpi useita mallintamista ja kieliteorioita koskevia artikkeleja. Mainituista kieliteorioista on kuitenkin tehty tutkimusta enemmän kuin mitä yhden tutkielman sivuilla voi käsitellä. Jatkotutkimuksessa voisikin olla mahdollista syventyä hyödyllisimmiksi katsottuihin kieliteorioihin tätä laajemmin. Siten voi ehkä huomata myös muita mallintamiselle tarjoutuvia mahdollisuuksia, joita tähän mennessä ei ole vielä käsitelty.

Jatkotutkimuksessa kannattaa kiinnittää huomiota tekoälyn ja kielimallien vaikutukseen tietojärjestelmien mallintamisessa. Tälläkin alalla kannattaa huomioida myös kielitieteen puolelta mahdollisesti esiin tulevat uudet teoriat. Aihe on verraten uusi, joten siitä kertyy varmasti lisää tutkimusta ja tietoa tulevaisuudessa. Lisäksi nopeasti kehittyvä teknologia edellyttää jatkuvaa tutkimusta, jotta muutoksissa pysytään ajan tasalla.

Tutkimuksessa olisi myös syytä huomioida käytännön näkökulma, jotta haastatteluissakin havaitulta teorian ja käytännön ristiriidalta voitaisiin välttyä. Pelkkään teoriaankin keskittymällä voidaan saada hyödyllisiä tuloksia, mutta monipuolisemman lopputuloksen saa yhdistämällä tutkimuksen konkreettiseen käytännön hyötyyn.

Tutkimukseen voidaan ottaa mukaan lisää haastateltavia, jolloin saadaan lisää näkökulmaa eri tilanteista, eri ajankohtina. Haastatteluilla olisi hyvä myös tuoda esiin kansainvälistä näkökulmaa eri kielialueilta. Lisäksi haastattelujen voisi olla myös hyvä olla pidempiä, jatkokysymyksiä ja täydentäviä kysymyksiä olisi esitettävä lisää tarvittaessa. Aiempia haastatteluja voi hyödyntää pohjana.

Haastatteluissa esiin tulleita lisätutkimuksen aiheita olivat etenkin tuottavuutta parantavat domain-spesifiset kielet, liikkeen huomiointi mallinnuskielissä sekä kysymys siitä, minkätyyppinen mallinnus tuo lisäarvoa, missä tilanteessa ja mihin tarkoituksiin.

Kirjallisuuden perusteella hyödyllistä näkökulmaa voisi löytyä pragmaatiikkaan, funktionaaliseen kielitieteeseen ja sosiaaliin konteksteihin liittyvistä teorioista. Tämä pro gradu -työ on osoittanut, että myöskään vanhempia tutkimuksia ei kannata kaihtaa, vaan tuoretta näkökulmaa voi löytää myös tarkaste-

lemalla uudelleen ja nykyalan valossa joitakin aikaisemmin hyödynnettyjä teorioita.

8 YHTEENVETO

Tietojärjestelmät ovat käyttäjien ja teknologian vuorovaikutuksen keskipisteessä. Ne yhdistävät eri osapuolet ja auttavat tulkinnassa, ymmärtämisessä ja kommunikaatiossa. Näin tietojärjestelmät toimivat vuorovaikutuksen välineenä tavalla, joka voidaan rinnastaa kielellisiin järjestelmiin. Tietojärjestelmien mallintamisella ja kielitieteellä onkin monia yhtymäkohtia. Molemmissa pohditaan, mitä todellisuudessa on ja miten sitä voisi kuvata. Siitä viestimiseen käytetään kieltä. Näin ollen lingvistic-filosofiset näkökulmat voivat avata hyödyllisen näkökulman mallinnusprosessiin.

Tietojärjestelmissä tarvitaan tietotekniikkaa järjestelmän tehtävien hoitamiseen ja kieltä tärkeiden viestien välittämiseen, mikä on pohjimmiltaan tietojärjestelmän sosiaalinen tarkoitus. (Hirscheim ym., 1996). Lisäksi valtaa organisaatioissa käytellään usein kielen kautta. Siten lingvistisen analyysin avulla voidaan päästä liiketoimintaratkaisujen ytimeen ja tutkia, kuka päättää, mitä ja miksi. (Kuo & Yin, 2011.)

Aina kieliperusteiset lähestymistavat eivät ole silti saaneet kehittäjien enakoimaa kannatusta. LAP on saanut suosiota joillakin aloilla, mutta se ei ole kuitenkaan vaikuttanut huomattavasti tietojärjestelmien suunnittelun lähestymistapoihin (Beynon-Davies, 2009). LAP-näkökulman esteeksi on voinut muodostua se, että UML:n on koettu riittävän tietojärjestelmien kehittämisen tarpeisiin. Näin ollen syytä uuden menettelyn opetteluun ei ole ollut. (Lyytinen, 2004).

Uuden opettelussa on tietenkin aina hankaluutensa. Viestintämalleja saatetaan kuvata erilaisin notaatioin kuin mihin on totuttu, ja yhteydet sosiaalisen järjestelmän ja teknologian välillä jäävät heikoiksi (Beynon-Davies 2009). Uusien tapojen omaksumiseen tarvittaisiinkin motivaatiota, jota kenties kielilähtökohtien etuja esiin tuovan tutkimuksen muodossa pystytään tarjoamaan.

Jokaisella lingvistisellä lähestymistavalla on omat, toisiaan täydentävät ominaisuutensa, ja niistä on vaikea valita vain yksi (Byrer & Jelassi 1989). Kieliteoreettisista lähtökohdista voidaankin saada parhaiten lisänäkemystä perehtymällä niihin mahdollisimman monipuolisesti ja eri lähtökohtien eri puolia yhdistelemällä, tilanteen ja kontekstin tarpeet huomioiden.

Kirjallisuudesta esiin piirtyi kielen ja todellisuuden suhteen laaja kirjo Fregen totuusväittämistä universaaliin kielioppiin, tekoälyyn ja kielimalleihin. Haastatteluissa puolestaan kerrottiin käytännön puolesta, jossa kieli näyttäytyy kiinnostavana ja hyödyllisenäkin seikkana, mutta kielipohjaiset mallit eivät kuitenkaan ole osoittautuneet niin hyödyllisiksi kuin joskus on kuviteltu. Mallintaminenkin on työlästä ja aikaa vievää, eikä automaattisesti joka tilanteessa hyödyllistä. Mallintaminen tuo kuitenkin etuja etenkin tilanteissa, joissa projektin eri osapuolten on kommunikoitava keskenään ja osapuolten teknisen osaamisen taso vaihtelee (esim. da Silva, 2015). Mallintamisella voisi siis ajatella olevan paikkansa myös tulevaisuudessa tilanteissa, joissa pitää pystyä ymmärtämään järjestelmää osaamatta lukea koodia.

Tekoäly kielimalleineen vaikuttaa suuresti moneen alaan jo nyt sekä tulevaisuudessa. Nykyään ChatGPT ja kielimallit käyttävät luonnollista kieltä ihmistä vastaavasti (Ooi ym., 2023). Samalla voivat muuttua kommunikaatiotarpeet: teknologia paranee ja automaattiset toiminnot lisääntyvät. Toistaiseksi kuitenkin on hyvä pitää ihminen mukana kehityksessä, jos ihminen on myös järjestelmän käyttäjä.

Tietojärjestelmien suunnittelussa on pitkälti kyse muutoksesta – niin IT:n, organisaation kuin kielenkin tasolla. Tietojärjestelmien jatkuva parantaminen ja pitäminen alan nopeiden muutosten mukana voi tuoda nykypäivän yrityksille olennaisen edun kilpailussa (Varajão ym., 2022). Käsityksiä ja teorioita pitää olla valmis muokkaamaan. Ketterä kehitys voi tarjota tähän yhden mahdollisen väylän. Toisaalta joskus ratkaisut tulevaisuuden ongelmiin voivat löytyä myös menneisyydestä. Eräs haastateltava varoitti tilanteesta, jossa vasara kädessä kaikki näyttää nauiloilta – ongelmia tulee, jos vastaan tuleekin ruuveja. Jos silti nauloja on joka tapauksessa naulattavana, aina vasaraa ei kannata keksiä uudelleen.

Sekä kirjallisuudessa että haastatteluissa viitattiin kieliteorioita edeltäneeseen paradigmaan, jossa tietokoneita pidettiin vain teknisinä laskennan välineinä. Tutkitun aineiston perusteella vaikuttaa siltä, että tämä asetelma on muuttunut, ja nykyisin tietokoneita ja tietojärjestelmiä tulkitaan ja tutkitaan paljon laajemmalla kantilta. Lingvistiset näkökulmat ovatkin omalta osaltaan muuttaneet asenteita, tuoneet uutta näkemystä ja tarjonneet uusia ratkaisuja.

LÄHTEET

- Abbasi, A., Zhou, Y., Deng, S. & Zhang, P. (2018). Text Analytics to Support Sense-Making in Social Media: A Language-Action Perspective. *MIS Quarterly*, Vol., 42, No. 2, s 427–464.
- Alastalo, M., Åkerman, M. & Vaittinen, T. (2017) Asiantuntijahaastattelu. Teoksessa Hyvärinen, M., Nikander, P., Ruusuvuori, J. (toim.): *Tutkimushaastattelun käsikirja*. [E-kirja] (s. 181–197). Tampere: Vastapaino.
- Alvarez, M. P. (2021). Why should we care about Skinner, thirty years on?. *Papeles del Psicólogo*, 42(1), 10-20.
- Alves, L. M., Souza, G., Ribeiro, P., & Machado, R. J. (2021). Longevity of risks in software development projects: a comparative analysis with an academic environment. *Procedia Computer Science*, 181, 827–834.
- Ambler, S. W. (2003). Agile model driven development is good enough. *IEEE Software*, 20(5):71–73.
- Atkinson, C. & Kühne, T. (2003). Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, 20(5), 36–41.
- Auramäki, E., Hirscheim, R. & Lyytinen, K. (1992). Modelling Offices Through Discourse Analysis: The SAMPO Approach. *The Computer Journal*, Vol. 35, No. 4.
- Auramäki, E., Lehtinen, E. & Lyytinen, K. (1988). A Speech-Act-Based Office Modeling Approach. *ACM Transactions on Office Information Systems*, Vol. 6, No. 2, April.
- Austin, J.L. (2016). *Näin tehdään sanoilla. Harvardissa 1955 pidetyt William James -luennot*. Urmson, J. O. & Sbisá, M. (toim.), R. Koskensilta (suom.). Tampere: niin & näin.
- Basden, A. & Klein, H. K. (2008). New Research Directions for Data and Knowledge Engineering: A Philosophy of Language Approach. *Data & Knowledge Engineering* 67, 260–285.
- Beynon-Davies, P. (2009). The ‘Language’ of Informatics: The Nature of Information Systems. *International Journal of Information Management* 29, 92–103.
- Bézivin, J. & Muller, P. A. (1998) UML: The Birth and Rise of a Standard Modeling Notation. Julkaisussa *The Unified Modeling Language*. "UML" '98:

Beyond the Notation. (s. 1–8) First International Workshop, Mulhouse, France, June 3–4.

Bider, I. & Perjons, E. (2012). Reviving language/ action perspective in the era of social software: research in process. Julkaisussa *The 5th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling* (s. 72–89). Rostock, Germany, November 7–8.

Bjeković, M. (2017). *Pragmatics of Enterprise Modelling Languages: A Framework for Understanding and Explaining*. Radboud University Nijmegen.

Bostrom, R. P. (1989) Successful Application of Communication Techniques to Improve the Systems Development Process. *Information & Management* 17, 279–295.

Burk, D. L. (2024). Asemic Defamation, or, the Death of the AI Speaker. *First Amendment Law Review*, 22.

Buscemi, A. (2023). A Comparative Study of Code Generation using ChatGPT 3.5 across 10 Programming Languages. *arXiv preprint arXiv:2308.04477*.

Byrer, J. K. & Jelassi, M. T. (1989). The Impact of Language Theories on DSS Dialog. *European Journal of Operational Research*, Vol. 50, Issue 2, s. 113–126.

Chakraborty, S., Rosenkranz, C. & Dehlinger, J. (2012). A Grounded Theoretical and Linguistic Analysis Approach for Non-Functional Requirements Analysis. *Thirty Third International Conference on Information Systems, Orlando*.

Charaf, M., C., Rosenkranz, C. & Holten, R. (2013). The Emergence of Shared Understanding: Applying Functional Pragmatics to Study the Requirements Development Process. *Info Systems J* 23, 115–135.

Chen, P. P. S. (1976). The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1), 9–36.

Chomsky, N. (1997) B. F. Skinnerin teos *Verbal Behavior*. Teoksessa Raatikainen, P. (toim.) *Ajattelu, kieli, merkitys: Analyyttisen filosofian avainkirjoituksia*. (s. 279–310) Helsinki: Gaudeamus.

Clarke, A. (1999). A practical use of key success factors to improve the effectiveness of project management. *International Journal of Project Management* Vol. 17 No. 3, s. 139–145.

Clarke, R. J., & Kautz, K. (2014). What's in a user story: IS development methods as communication. *Proceedings of the 23rd International Conference on Information Systems Development* (s. 356–364). Croatia: University of Zagreb.

- Curty, S., Härer, F., & Fill, H. G. (2023). Design of blockchain-based applications using model-driven engineering and low-code/no-code platforms: a structured literature review. *Software and Systems Modeling*, 22(6), 1857–1895.
- Dehaerne, E., Dey, B., Halder, S., De Gendt, S., & Meert, W. (2022). Code generation using machine learning: A systematic review. *Ieee Access*, 10, 82434-82455.
- DeMarco, T. (1978). Structured Analysis and System Specification. *Original Historic Documents* 529. Yourdon, New York, s. 1-7 ja 37-44.
- Desharnais, J., Frappier, M., & Mili, A. (1998). State transition diagrams. *Julkaisussa Handbook on architectures of information systems* (s. 147-166). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Deursen, A. van, Klint, P., & Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM Sigplan Notices*, 35(6), 26–36.
- Dreiling, A. (2006). On the impact of the ‘linguistic turn’ on research in information systems. *European Conference on Information Systems*.
- Ekstrand, M. D., & Willemsen, M. C. (2016). Behaviorism is not enough: better recommendations through listening to users. *Julkaisussa Proceedings of the 10th ACM conference on recommender systems* (s. 221–224).
- Enos, C. A. (2024). "Noam Chomsky," *The Internet Encyclopedia of Philosophy*, ISSN 2161-0002, haettu 27.2.2024 osoitteesta <https://iep.utm.edu/chomsky-philosophy/>.
- Fayoumi, A. & Loucopoulos, P. (2016). Conceptual modeling for the design of intelligent and emergent information systems. *Expert Systems with Applications* 59, 174–194.
- Fernández, D.M., Wagner, S. & Kalinowski, M. ym. (2017). Naming the pain in requirements engineering. Contemporary problems, causes, and effects in practice. *Empir Software Eng* 22:2298–2338.
- Flores, F. & Ludlow, J. J. (1976). Doing and Speaking in the Office. Teoksessa G. Fick, H. Sprague Jr (toim.). *Decision Support Systems: Issues and Challenges* (s. 95–118), IIASA Proceedings Series. Pergamon Press, New York.
- Fowler, M. & Scott, K. (2000). UML distilled: a brief guide to the standard object modeling language. Second Edition. Addison Wesley. Noudettu 17.5.2024 osoitteesta <https://pja.mykhi.org/6sem/MAS/books/Addison%20Wesley%20-%20UML%20Distilled,%202nd%20Edition.pdf>

- France, R. B., Ghosh S. & Dinh-Trong, T. (2006) Model-Driven Development Using UML 2.0: Promises and Pitfalls. *Computer*, vol. 39, no. 2, s. 59–66.
- Frank, U. (2002). Multi-Enterprise Perspective Modeling (MEMO) – Conceptual Framework and Modeling Languages. Julkaisussa *Proceedings of the 35th Hawaii International Conference on System Sciences* (s. 1258–1267). IEEE.
- Frege, G. (1997). Mielestä ja merkityksestä. Teoksessa Raatikainen, P. (toim.) *Ajattelu, kieli, merkitys: Analytyttisen filosofian avainkirjoituksia* (s. 41–56). Helsinki: Gaudeamus.
- Giaglis, G. M. (2001). A Taxonomy of Business Process Modeling and Information Modeling Techniques. *The International Journal of Flexible Manufacturing Systems*, 13. 209–228.
- Golra, F. R., Beugnard, A., Dagnat, F., Guerin, S., & Guychard, C. (2016). Using free modeling as an agile method for developing domain specific modeling languages. Julkaisussa *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems* (s. 24–34).
- Graham, G. (2023). "Behaviorism", *The Stanford Encyclopedia of Philosophy* (kevään 2023 versio), Edward N. Zalta & Uri Nodelman (toim.), haettu 19.5.2024 osoitteesta <https://plato.stanford.edu/archives/spr2023/entries/behaviorism/>.
- Greenspan, S., Mylopoulous, J. & Borgida, A. (1994) On Formal Requirements Modeling Languages: RML Revisited. *Proceedings of 16th International Conference on Software Engineering* (135–147), Sorrento, Italy.
- Grossman, M., Aronson, J. E. & McCarthy, R. V. (2005). Does UML Make the Grade? Insights from the Software Development Community. *Information and Software Technology* 47, s. 383–397.
- Halttunen, V. (2019). Adding 'Sense' to Conceptual Modeling: An Interdisciplinary Approach. Julkaisussa J. Dietz, D. Aveiro, & J. Filipe (toim.), *IC3K 2019: Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. Volume 2: KEOD 2019: Proceedings of the 11th International Conference on Knowledge Engineering and Ontology Development* (s. 248–255). SCITEPRESS Science And Technology Publications.
- Hanseth, O., & Monteiro, E. (1994). Modelling and the representation of reality: some implications of philosophy on practical systems development. *Scandinavian Journal of Information Systems*, 6(1), 3.

- Harel, D. & Rumpe, B. (2000). *Modeling Languages: Syntax, Semantics and All That Stuff. Part I: The Basic Stuff. Technical Report (1-28.)* Weizmann Science Press of Israel. Jerusalem, Israel.
- Harel, D. & Rumpe, B. (2004) Meaningful Modeling: What's the Semantics of "Semantics"? *Computer*, vol. 70, no. 10, s. 64-72.
- Hirscheim, R., Klein, H. K. & Lyytinen, K. (1996). Exploring the Intellectual Structures of Information Systems Development: A Social Action Theoretic Analysis. *Accounting, Management and Information Technologies*, Vol. 6, No. ½, s. 1-64.
- Hirsjärvi, S, & Hurme, H. (2022). *Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö.* [E-kirja] Gaudeamus.
- Husáková M. & Bureš V. (2020) Formal Ontologies in Information Systems Development: A Systematic Review. *Information*. 11(2):66.
- Hyvärinen, M. (2017). Haastattelun maailma. Teoksessa Hyvärinen, M., Nikander, P., Ruusuvuori, J. (toim.): *Tutkimushaastattelun käsikirja* (s. 9-37). Tampere: Vastapaino.
- Häkkinen, K. (1995). *Kielitieteen perusteet*. (2. painos). Tampere: Tammer-Paino Oy.
- IBM. *Flowcharting Techniques*, GC20-8152-1 edition, 1969.
- Jaakkola, H., Henno, J., Welzer-Druzovec, T., Thalheim, B., & Mäkelä, J. (2016). Why Information Systems Modelling Is Difficult. Julkaisussa *SQAMIA* (s. 29-39).
- Janson, M. A., Woo, C. C. (1995). Comparing IS Development Tools and Methods: Using Speech Act Theory. *Information & Management* 28, 1-12.
- Janson, M. A., Woo, C. C. (1996). A Speech Act Lexicon: An Alternative Use of Speech Act Theory in Information Systems. *Info Systems J* 6, 301-329.
- Kaur, K., Kalid, K. S., & Sugathan, S. (2021, July). Exploring children user experience in designing educational mobile application. Teoksessa *2021 International Conference on Computer & Information Sciences (ICCOINS)* (s. 163-168). IEEE.
- Kellner, M. I., Madachy, R. J., Raffo, D. M. (1999). Software Process Simulation Modeling: Why? What? How? *The Journal of Systems and Software* 46, 91-105.

- Krajewski, M. (2020). The Structure of (Information) Infrastructure. Origins, History, and Theory of the Flow Chart. *BMCCT working papers*, (August 2020) No. 001. University of Basel.
- Kuo, F.-Y. & Yin, C-P. (2011) A Linguistic Analysis of Group Support Systems Interactions for Uncovering Social Realities of Organizations. *ACM Transactions on Management Information Systems*, Vol. 2, No. 1, Article 3.
- Koskensilta, R. (2016). Suomentajan jälkisanat. Teoksessa Näin J. L. Austin, *Näin tehdään sanoilla* (s. 151–164). Tampere: niin & näin.
- Kotimaisten kielten keskus & Kielikone Oy (2024). Haettu 17.5.2024 osoitteesta <https://www.kielitoimistonsanakirja.fi>.
- Kühne, T. (2006) Matters of (meta-)modeling. *Software and Systems Modeling* 5(4):369–385.
- Lange, C. F. J. & Chaudron, M. R. V. (2006). In Practice: UML Software Architecture and Design Description. *IEEE Software*, vol. 23, no. 2, s. 40–46.
- Liddle, S. W. (2011). Model-driven software development. Julkaisussa D. W. Embey & B. Thalheim (toim.), *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges* (17–54). Berlin: Springer-Verlag.
- Liebel, G., Tichy, M., Knauss, E., Ljungkrantz, O. & Stieglbauer, G. (2018). Organisation and communication problems in automotive requirements engineering. *Requirements Eng* 23:145–167.
- Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T. & Ge, B. (2023). Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 100017.
- Luong, T.T., Sivarajah, U. & Weerakkody, V. (2021). Do Agile Managed Information Systems Projects Fail Due to a Lack of Emotional Intelligence?. *Inf Syst Front* 23, 415–433.
- Luoma, J., Kelly, S. & Tolvanen, J-P. (2004). Defining Domain-Specific Languages: Collected Experiences. Teoksessa J.-P. Tolvanen, J. Sprinkle, M. Rossi (toim.): *Proceedings of the 4th OOPSLA Workshop on Domain-Specific Modeling (DSM04)* (s. 1–10), Computer Science and Information System Reports, Technical Reports, TR-33, Jyväskylän yliopisto.
- Lyytinen, K. (1985). Implications of Theories of Language on Information Systems. *MIS Quarterly*, Vol. 9, No. 1, s. 61–74.
- Lyytinen, K. (1987). Two Views of Information Modeling. *Information & Management* 12, 9–19.

- Lyytinen, K. (2004). The Struggle with the Language in the IT – Why is LAP not in the Mainstream? *Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004)*. Rutgers University, The State University of New Jersey, New Brunswick, NJ, USA, June 2-3.
- McManus, J. & Wood-Harper, T. (2008). Understanding the Sources of Information Systems Project Failure. *ITNOW*. 51. 38-43.
- Mernik, M., Heering, J. & Sloane, A. M. (2005). When and How to Develop Domain-Specific Languages. *ACM Computing Surveys*, Vol. 37, s. 316-344.
- Mingers, J., & Willcocks, L. (2014). An integrative semiotic framework for information systems: The social, personal and material worlds. *Information and Organization*, 24(1), 48-70.
- Nelson, R. R. (2007). IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices. *MIS Quarterly Executive* Vol. 6 No. 2, June.
- Neto, A. N. F. & Neto, J. S. (2011). Metamodels of Information Technology Best Practices Frameworks. *JISTEM - Journal of Information Systems and Technology Management*, Vol. 8, No. 3, s. 619-640.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N. & Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- Noran, O. S. (2000). Business modelling: UML vs. IDEF. *School of Computing and Information Technology, Griffith University*.
- Ooi, K. B., Tan, G. W. H., Al-Emran, M., Al-Sharafi, M. A., Capatina, A., Chakraborty, A., Dwivedi, Y. K., Huang T. L., Kar A. K., Lee V. H. Loh X. M., Micu A., Mikalef P., Mogaji E., Pandey N., Raman R., Rana N. P, Sarker P., Sharma A., Teng C-I., Wamba S. F. & Wong, L. W. (2023). The potential of generative artificial intelligence across disciplines: Perspectives and future directions. *Journal of Computer Information Systems*, 1-32.
- Peterson, J.L. (1981). *Petri Net Theory and the Modeling of Systems*. Yhdysvallat: Prentice-Hall.
- Richir, S., Fuchs, P., Lourdeaux, D., Millet, D., Buche, C. & Querrec, R. (2015). How to design compelling Virtual Reality or Augmented Reality experience?. *International Journal of Virtual Reality*, 15(1), 35-47.
- Rico Carranza, E., Huang, S. Y., & Besems, J. (2023). (In) visible Cities: What Generative Algorithms Tell Us About Our Collective Memory Schema. In *CAADRIA proceedings* (s. 463-472). CAADRIA.

- Sanghvi, P. (2020). Piaget's theory of cognitive development: a review. *Indian Journal of Mental Health*, 7(2), 90-96.
- Seidewitz, E. (2003). What models mean. *IEEE Software*, 20(5), 26-32.
- Schichl, H. (2004). Models and History of Modeling. *Julkaisussa Modeling Languages in Mathematical Optimization* (s. 25-36). Norwell, MA, USA: Kluwer Academic Publishers.
- Schmidt, D. C. (2006). Model-Driven Engineering. *IEEE Computer*, 39 (2), 25-31.
- Schoop, M. (1999). An empirical study of multidisciplinary communication in healthcare using a language-action perspective. *Julkaisussa Proceedings of the Fourth International Workshop on the Language Action Perspective on Communication Modelling (LAP 99)* (s. 59-72).
- Schoop, M. (2001). An introduction to the language-action perspective. *Acm Siggroun Bulletin*, 22(2), 3-8.
- Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press.
- Silva, A. R. da (2015). Model-Driven Engineering: A Survey Supported by the Unified Conceptual Model. *Computer Languages, Systems & Structures* 43, 139-155.
- Sinha, C. (2010). Cognitive Linguistics, Psychology, and Cognitive Science. *Julkaisussa Geeraerts, D., & Cuyckens, H. (toim.). The Oxford Handbook of Cognitive Linguistics. Oxford Handbooks* (2010; online, Oxford Academic, 2012), <https://doi.org/10.1093/oxfordhb/9780199738632.013.0049>.
- Shin, J., & Nam, J. (2021). A survey of automatic code generation from natural language. *Journal of Information Processing Systems*, 17(3), 537-555.
- Skinner, B. F. (1948). *Verbal Behavior*. William James Lectures. Harvard University Press.
- Sommerville, I. (2011). *Software Engineering*. (Ninth Edition). Boston, MA, USA: Pearson Addison-Wesley.
- Stamper, R., Liu, K., Hafkamp, M. & Ades, Y. (2000) Understanding the Roles of Signs and Norms in Organisations. *Journal of Behaviour & Information Technology*, Vol. 19 (1), s. 15-27.
- Subahi, A. F., & Alotaibi, Y. (2018). A New Framework for Classifying Information Systems Modelling Languages. *J. Softw.*, 13(1), 18-42.

- Thomas, N. (2013). Social computing and the manufacture of sense. *Proceedings of the Nineteenth Americas Conference on Information Systems*.
- Tieteen termipankki (2024b): Avoin tiede:ontologia. (Haettu 6.2.2024 osoitteesta [https://tieteentermipankki.fi/wiki/Avoin tiede:ontologia](https://tieteentermipankki.fi/wiki/Avoin_tiede:ontologia).)
- Tieteen termipankki (2024a): Filosofia:ontologia. (Haettu 6.2.2024 osoitteesta <https://tieteentermipankki.fi/wiki/Filosofia:ontologia>.)
- Truex, D. P., & Baskerville, R. (1998). Deep structure or emergence theory: contrasting theoretical foundations for information systems development. *Information Systems Journal*, 8(2), 99-118.
- Varajão, J. ., Trigo, A. ., Pereira, J. L. ., & Moura, I. (2022). Information systems project management success. *International Journal of Information Systems and Project Management*, 9(4), 62-74.
- Wand, Y., Monarchi, D. E., Parsons, J. & Woo, C. C. (1995). Theoretical Foundations for Conceptual Modelling in Information Systems Development. *Decision Support Systems* 15, 285-304.
- Wand, Y., & Weber, R. (1995). On the deep structure of information systems. *Information systems journal*, 5(3), 203-223.
- Winograd, T. (1980). What Does it Mean to Understand Language? *Cognitive Science* 4, 209-241.
- Yeo, K. T. (2002). Critical Failure Factors in Information System Projects. *International Journal of Project Management* 20, 241-246.
- Zalta, Edward N., "Gottlob Frege", *The Stanford Encyclopedia of Philosophy* (kevät 2024), Edward N. Zalta & Uri Nodelman (toim.), haettu 19.5.2024 osoitteesta <https://plato.stanford.edu/archives/spr2024/entries/frege/>.