

Timo Itälä, Juha Mykkänen, Hannu Virkanen, Tuija Tiihonen, Kari Hiekkanen,  
Irmeli Luukkonen, Ilkka Sammeltu, Ilkka Melleri, Yong Han

## **Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet**

SOLEA-hanke  
Itä-Suomen yliopisto  
Aalto-yliopisto





Timo Itälä, Juha Mykkänen, Hannu Virkanen, Tuija Tiihonen, Kari Hiekkänen, Irmeli Luukkonen, Ilkka Sammelvu, Ilkka Melleri, Yong Han

## **Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet**

Kehikot, jäsentämismallit, notaatiot ja niiden yhteensovittaminen arkkitehtuurityössä

Itä-Suomen yliopisto ja Aalto-yliopisto

Kuopio

2012



©Itä-Suomen yliopisto ja Aalto-yliopisto 2012  
SOLEA-hanke  
<http://www.uef.fi/solea>

ISBN 978-952-61-0723-3 (PDF)

Timo Itälä, Juha Mykkänen, Hannu Virkanen, Tuija Tiihonen, Kari Hiekkänen, Irmeli Luukkonen, Ilkka Sammelvuo, Ilkka Melleri, Yong Han. Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet. Kehikot, jäsentämismallit, notaatiot ja niiden yhteensovittaminen arkkitehtuurityössä. Itä-Suomen yliopisto ja Aalto-yliopisto. 2012. 77 s.  
ISBN 978-952-61-0723-3 (PDF)

## Tiivistelmä

Tässä dokumentissa esitellään kokonaisarkkitehtuuria sekä kokonaisarkkitehtuurityössä tarpeellisia arkkitehtuurikehikoita, jäsentämismalleja, työmenetelmiä notaatioita ja työvälineitä. Dokumentti on tarkoitettu yhteenvedoksi SOLEA-projektin aikana pidetyistä esityksistä ja tehdyistä selvityksistä, jotka liittyvät kokonaisarkkitehtuurin kehikoihin ja kuvauksiin sekä niiden tukemiseen käytettyihin jäsentämismalleihin, notaatioihin, menetelmiin ja työkaluihin. Lopuksi vielä pohditaan, mitä erilaista osaamista ja kyvykkyyksiä tarvitaan kokonaisarkkitehtuurityössä. Dokumenttia voivat hyödyntää myös muut kuin projektin osapuolet.

Eryteisesti dokumentti on suunnattu tietohallinnon, systeemisuunnittelun, projektityön, prosessien kehittämisen ja arkkitehtuurityön ammattilaisille. Sisältö on hyödynnettävissä toimialariippumattomasti, vaikka jotkin esimerkit liittyvät terveydenhuoltoon, julkiseen hallintoon tai yritystoimintaan. Dokumentissa on mukana myös kokonaisarkkitehtuurin tiettyihin näkökulmiin tai osa-alueisiin liittyviä malleja.

Dokumentissa tuodaan esille yleisluontoisia näkemyksiä kokonaisarkkitehtuurista ja palvelukeskeisyydestä "näin se voitaisiin tehdä" -tyyppisesti. Näkemykset ja kannanotot perustuvat kirjallisiin lähteisiin, SOLEA-hankkeen työpajojen, kokousten ja työkohteiden tuloksiin ja kirjoittajien omiin mielipiteisiin ja havaintoihin.

Luokitus: UDK 004 Tietotekniikka

Asiasanat: kokonaisarkkitehtuuri, tiedonhallinta, (YSA): tiedonhallintajärjestelmät, järjestelmäarkkitehtuuri, palvelujärjestelmät, prosessit, systeemityö



## Sisällys

<b>1</b>	<b>Dokumentin tavoitteet.....</b>	<b>7</b>
<b>2</b>	<b>Johdanto: kokonaisarkkitehtuuri.....</b>	<b>8</b>
2.1	Mitä on kokonaisarkkitehtuuri ja mihin sitä tarvitaan? .....	8
2.2	60-luvun näkemys kokonaisarkkitehtuurista .....	9
2.3	Arkkitehtuurityö.....	12
2.4	Palvelupohjaisuuden perusteet.....	13
2.5	SOLEAn näkökulma kokonaisarkkitehtuuriin.....	15
2.6	Dokumentin jäsenitys.....	17
<b>3</b>	<b>Arkkitehtuurikehikot.....</b>	<b>17</b>
3.1	JHS 179.....	17
3.2	Kartturi 2.0.....	18
3.3	TOGAF™ Version 9 .....	19
3.4	Zachman Framework.....	20
<b>4</b>	<b>Kokonaisarkkitehtuurin jäsentämismallit.....</b>	<b>21</b>
4.1	SOLEA:ssa käytettävä jäsenitys.....	21
4.2	Zachman jäsenitys .....	23
4.3	Capgemini IAF (Integrated Architecture Framework).....	24
4.4	JHS 179 näkökulmat .....	26
4.5	Kartturi 2.0 näkökulmat .....	27
4.6	RM-ODP näkökulmat .....	29
4.7	Archimate jäsenitys.....	31
<b>5</b>	<b>Yhtä näkökulmaa painottavat jäsentämismallit .....</b>	<b>33</b>
5.1	Business Model Canvas .....	33
5.2	S3: A Service-Oriented Reference Architecture .....	35
5.3	RIM HL 7 V 3 reference information model.....	36
<b>6</b>	<b>Kokonaisarkkitehtuurimenetelmät.....</b>	<b>38</b>
6.1	The Architecture Development Method (TOGAF ADM) .....	38
<b>7</b>	<b>Notaatiot .....</b>	<b>40</b>
7.1	Kokonaisarkkitehtuurin osa-alueiden kuvaukset.....	40
7.2	ArchiMate.....	41
7.3	BPMN .....	44
7.4	ORM.....	45
7.5	SoaML .....	50
7.6	UML .....	53
7.7	Yhteenvedo notaatioista .....	53
<b>8</b>	<b>Mallit ja arkkitehtuurikuvaukset.....</b>	<b>54</b>
8.1	TOGAF .....	54
8.2	Keskeisimmät KA-kuvaukset .....	57

<b>9 Työvälineet</b> .....	<b>58</b>
9.1 Kuvausten tyyppi .....	58
9.2 Yksilötyö vai Ryhmätyö .....	58
9.3 Notaatioiden ja eri kuvaustapojen tuki .....	59
9.4 Koodin generointi.....	60
9.5 Työvälineiden alustavaatimukset .....	61
9.6 Kokonaisarkkitehtuurikehikoiden tuki.....	61
<b>10 Osaaminen ja kyvykkyydet</b> .....	<b>61</b>
10.1 TOGAF Architecture Capability Framework.....	61
10.2 Schekkerman Who are involved in EA.....	63
<b>11 Menetelmien ja välineiden yhteensovittaminen arkkitehtuurityössä</b> .....	<b>63</b>
<b>12 Yhteenveto</b> .....	<b>64</b>
<b>Lähteet</b> .....	<b>68</b>

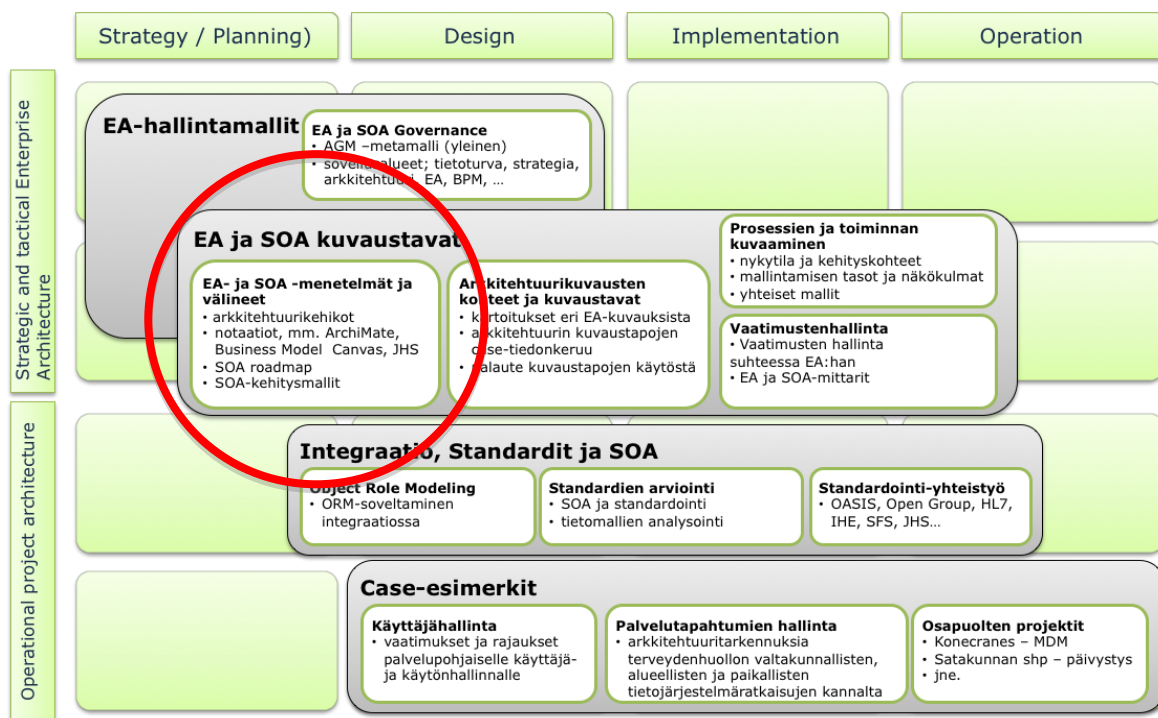


## 1 Dokumentin tavoitteet

Kokonaisarkkitehtuurin kehittämisessä ja palvelukeskeisyyden soveltamisessa keskeisiä tekijöitä ovat kehittämistyötä ohjaavat menetelmät ja työssä käytettävät välineet. Tässä dokumentissa tehdään yhteenvedoa SOLEA-projektin aikana selvitetystä EA- ja SOA-menetelmistä ja välineistä ja niiden käyttöön liittyvistä kokemuksista ja ohjeista.

Dokumentin tavoitteena on osaltaan koota SOLEA tutkimusprojektin vastaukset sen keskeisiin tutkimuskysymyksiin, joita ovat:

- Miten **palvelukeskeisyyden (SOA)** pitäisi näkyä yrityksen **kokonaisarkkitehtuurin (EA)** kehittämisessä ja miten sillä kavennetaan (liike)toiminnan ja teknisen kehittämisen kuilua?
- Miten saavutan kokonaisarkkitehtuurin kehittämisessä hyödyt palvelupohjaisesta toimintamallista ja sovitan yleiset ratkaisut **paikallisia** tarpeita tarkasti palvelevaksi kokonaisuudeksi?



Kuva 1: Dokumentin kohdealue

Sekä palveluihin että tuotteisiin kohdistuu *kehittämistavoitteiden laajennettu elinkaari*, jossa strategisten linjausten tulee ohjata sekä pitkäkestoisia arkkitehtuurivalintoja että lukuisia eri toimijoiden projekteja. Strategisten tai verkostossa tapahtuvien pitkäkestoisten kehittämisponnistusten onnistuminen ratkaistaan kuitenkin yksittäisissä projekteissa ja niissä tapahtuvassa prosessien, palvelujen, tuotteiden ja tekniikoiden yhteensovittamisessa. Tällaisen kokonaisuuden hallinnassa jäsenyykset, menetelmät ja välineet joilla kokonaisarkkitehtuuria hallitaan ovat erityisen keskeisiä. Palvelukeskeisyydellä voidaan myös jäsenyksissä ja menetelmissä pyrkiä lisäämään kokonaisuuden ketteryyttä.

Dokumentti on tarkoitettu yhteenvedoksi SOLEA-projektin aikana pidetyistä esityksistä ja tehdyistä selvityksistä, jotka liittyvät kokonaisarkkitehtuurin kehikoihin ja kuvauksiin sekä niiden tukemiseen käytettyihin jäsentämismalleihin, notaatioihin, menetelmiin ja työkaluihin. Dokumenttia voivat hyödyntää myös muut kuin projektin osapuolet. Erityisesti dokumentti on suunnattu tietohallinnon, systeemisuunnittelun, projektityön, prosessien kehittämisen ja arkkitehtuurityön ammattilaisille. Sisältö on hyödynnettävissä toimialariippumattomasti, vaikka jotkin esimerkit liittyvät terveydenhuoltoon, julkiseen hallintoon tai yritystoimintaan. Dokumentissa on mukana myös kokonaisarkkitehtuurin tiettyihin näkökulmiin tai osa-alueisiin liittyviä malleja.

Dokumentissa tuodaan esille yleisluontoisia näkemyksiä kokonaisarkkitehtuurista ja palvelukeskeisyydestä "näin se voitaisiin tehdä" -tyyppisesti. Näkemykset ja kannanotot perustuvat kirjallisiin lähteisiin, SOLEA-hankkeen työpajojen, kokousten ja työkohteiden tuloksiin ja kirjoittajien omiin mielipiteisiin ja havaintoihin.

Dokumentissa ei ole tarkoitus selvittää joidenkin menetelmien tai välineiden yksityiskohtia. Niitä varten on valmiita lähteitä runsaasti saatavilla. Myöskään SOLEA-projektin muissa tuotoksissa esille tulevaa sisältöä ei ole tarkoitus toistaa tässä dokumentissa. SOLEA-projektin muita raportteja ovat mm. vaatimustenhallinta, prosessien kuvaustavat, SOA-hallintamallit, EA- ja SOA kuvaustavat jne., joista viitataan yleisten jäsenysmallien ja notaatioiden osalta tähän dokumenttiin.

Dokumentissa pyritään erityisesti tuomaan esille pohdintaa kokonaisarkkitehtuurin ja palveluarkkitehtuurin liittymisestä toisiinsa. Usein kirjallisuudessa tarkastelu kiinnittyy johonkin teemaan kuten kokonaisarkkitehtuuriin tai palveluarkkitehtuuriin tai prosessien kehittämiseen tai tietoarkkitehtuuriin. Hyvin vähän kirjallisuudesta löytyy pohdintaa näistä yhdessä ja sitä pyritäänkin tässä dokumentissa tuomaan esille.

## 2 Johdanto: kokonaisarkkitehtuuri

### 2.1 Mitä on kokonaisarkkitehtuuri ja mihin sitä tarvitaan?

2000-luvun alkupuolella käsite kokonaisarkkitehtuuri - Enterprise Architecture - on saanut lisääntyvää näkyvyyttä konsulttien ja tietohallinnon ammattilaisten ajankohtaisissa puheenvuoroissa. Samalla aihetta käsittelevän kirjallisuuden, koulutuksen ja myös työvälineistöjen tarjonta on voimakkaassa kasvussa. Mistä kokonaisarkkitehtuurissa oikein on kysymys ja mihin sitä tarvitaan? Ohessa kahdesta lähteestä lainattua kuvausta kokonaisarkkitehtuurista.

#### **Julkinen hallinto**

Julkisen hallinnon kokonaisarkkitehtuurityön tuloksia on koottu yhteentoimivuuportaaliin <https://www.yhteentoimivuuus.fi/view/index.xhtml> . Yhteentoimivuuus.fi on kaikille avoin portaali, mutta vaatii aluksi rekisteröitymisen ja käyttäjätunnuksen luomisen. Yhteentoimivuuportaalia hallinnoi valtiovarainministeriö. Sivustolla esitellään mm. julkisen hallinnon, valtionhallinnon ja kuntasektorin kokonaisarkkitehtuurit. Sivustolta ladattavassa valtiovarainministeriön julkaisemassa yleisesitteessä todetaan seuraavasti: *Kokonaisarkkitehtuuri kuvaa, kuinka organisaation toimintaprosessit, organisaatioyksiköt, tiedot ja järjestelmät toimivat kokonaisuutena. Kokonaisarkkitehtuu-*

*rin näkökulmat ovat toiminta, tiedot, tietojärjestelmät ja teknologia. Jokaiseen näistä näkökulmista sisältyy tietoturva- ja integraatoratkaisuja.*

Kokonaisarkkitehtuurin hyödyistä todetaan seuraavasti: *Rakenteita tunnistamalla ja kuvaamalla kokonaisarkkitehtuuri toimii välineenä, joka lisää mahdollisuuksia uudelleenkäyttöön (myös organisaatorajat ylittävään), vähentää päällekkäisyyttä, johtaa kustannusten pienenemiseen, kuvaa ja selittää kehittämissuhteiden väliset suhteet ja auttaa sekä muutoksentekemisessä että hankinnoissa. Kokonaisarkkitehtuuri tuo menetelmän ja työkalun muutosten vaikutusten läpikäynnille ja arvioinnille. Syy- seuraussuhteiden järjestelmällinen läpikäynti lisää ymmärrystä kehittämiskohteista ja auttaa parempien kehitysratkaisujen ja –päätösten tekemisessä.*

### EA Wiki

EA Wiki <http://iea.wikidot.com/start> on kokonaisarkkitehtuuria käsittelevä sivusto, jolle on koottu aiheeseen liittyvää hyödyllistä tietoa ja linkkejä eri lähteisiin. Näitä ovat mm. Archimate, COBIT, TOGAF, FEAF, SOA, ITIL, XGEA, Zachman, EBA and Information First. Sivustolla kokonaisarkkitehtuuri määritellään seuraavasti: *Enterprise Architecture represents the fundamental description of an organisation, its structure, the information that it requires, the business processes that use and produce the information, its software applications that provide the services the business processes depend on, the structure of the application components, products, their interrelationships to each other and the technical infrastructure and environment in which they operate, all driven by the strategy, which states the principles, standards, etc. that govern their design and evolution over time. It also covers the performance and measures necessary to track the enterprise.*

Kokonaisarkkitehtuurin käyttötarkoitusta EA Wiki kuvaa seuraavasti:

*The Enterprise Architecture is used to support IT Management decisions for:*

- *consolidation*
- *improving performance*
- *communication*
- *governance and compliance*
- *innovation*
- *improving efficiency*
- *making cost savings*
- *realising strategies*

### TOGAF

Esimerkiksi TOGAF:in mukainen mallien käyttö liittyy yleensä siihen, että on tarpeen tehdä muutoksia. Kokonaisarkkitehtuurissa kuvataan usein nyky- ja tavoitetilaa eri näkökulmista ja kokonaisuutena. Toiminnan muutosten tavoitteiden ja toteutusten vaikutusten tarkastelu on keskeinen kokonaisarkkitehtuurin käyttökohde. Aiemmin korostuneen tietojärjestelmän näkökulman lisäksi tämän kokonaisvaltaisen muutostenhallinnan merkitys on entistä olennaisempi kokonaisarkkitehtuurin käyttökohde.

## 2.2 60-luvun näkemys kokonaisarkkitehtuurista

Mitä näissä määritelmässä on sitten uutta aikaisempiin määritelmiin verrattuna? Vuodelta 1969 peräisin olevassa Sven R. Hedénin kirjassa ATK - Automaattinen tietojenkäsittely käydään läpi "ATK-menetelmän" käyttöönotossa tarvittavia työkulkukaavioita, rekistereitä, sovelluksia ja tietokoneita (Hed, 1969). Myös yrityksen laajuinen eri "tietoalueiden" integraatio nähdään merkittävänä mah-

dollisuutena. Siis kokonaisarkkitehtuurin elementit olivat jo silloin ajankohtaisia. Kiinnostunutta lukijaa varten on ohessa kursiiivilla joitakin poimintoja kirjassa kuvatuista ATK-työn vaiheista sivuilta 337-339:

### **Esitutkimus**

*"1. Kun yrityksen johto on kiinnostunut ATK:sta ja tehnyt päätöksen tietojenkäsittelyn tutkimisesta, suoritetaan ns. esitutkimus. Tällöin tutkitaan lähinnä niitä alueita, joilla parannetun tietojenkäsittelyn oletetaan tuottavan parhaat tulokset. Näin saadaan summittainen käsitys ATK:n sopivuudesta, nähdään, "minne päin piiput ovat kallellaan". Esitutkimuksen jälkeen ei mitään tarkempaa lausuntoa asiasta voidakaan antaa, mikäli kannattavuus ei ole erittäin suuri ja selvästi todettavissa. Esitutkimuksen tarkoituksena on antaa vastaus kysymykseen: "Onko yksityiskohtaisempi tutkimus suoritettava"?"*

*2. Jos vastaus tähän kysymykseen on myönteinen ja liikkeen johto tekee sen mukaisen päätöksen, alkaa yksityiskohtainen tutkimus. Kartoittamalla ja analysoimalla nykyistä tietojenkäsittelyjärjestelmää saadaan käsitys kustannuksista, osittain nykyisen järjestelmän kustannuksista sellaiseenaan, osittain nykyisen järjestelmän kustannuksista sen jälkeen, kun kartoittamisen aikana havaitut parannusmahdollisuudet on toteutettu. Tämän jälkeen suoritetaan systeemin suunnittelu ATK-menetelmää silmällä pitäen. Uuden systeemin taloudellisia seuraamuksia verrataan nykyisestä järjestelmästä sen parannetussa asussa tehtyihin laskelmiin. Tämän perusteella voidaan vastata seuraavaan kysymykseen: "Onko yrityksemme ryhtyttävä käyttämään ATK-menetelmää?" Systeemin suunnittelu antaa myös viitteitä tietojenkäsittelyn mekaanisten välineiden valinnan osalta. Tällöin voidaan ratkaista kysymys: "Mikä konetyyppi, mikä koneenvalmistaja ja mikä tietokoneiden hyväksikäyttötapa meidän tulee valita?"*

*3. Kun yrityksen johto on ratkaissut nämä kysymykset ja tehnyt sopimuksen koneenvalmistajan kanssa, aletaan valmistella tietokoneen asennointia ja ATK-menetelmän käyttöönottoa. Tämä vaihe käsittää henkilökunnan hankkimisen ja koulutuksen, systeemin yksityiskohtaisen suunnittelun, ohjelmoinnin, varsinaisen tietokoneen asennoinnin ja siirtymisen uuteen menetelmään. Valmistelutöiden aikana tulevat uuden systeemin yksityiskohdat esille ja voivat antaa aiheita konekokoelman muutoksiin lisälaitteiden, syöttö- ja tulostuslaitteiden jne. osalta.*

*Kun kone ja ATK-menetelmä on otettu käyttöön, on varsinainen työ suoritettava taloudellisella tavalla samalla aikaa kun uusia tietojenkäsittelyalueita tutkitaan ja siirretään tietokonejärjestelmään. Useimmissa tapauksissa on nimittäin täysin mahdollonta siirtää ATK-menetelmän piiriin koko tietojenkäsittelyjärjestelmää yhdellä kertaa; sen sijaan edetään vaiheittain."*

### **Johdon sitoutuminen**

Yrityksen johdon ja tietotekniikan välinen yhteys nähdään kirjassa hyvin kiinteänä:

*"Yrityksen johdolle on jatkuvasti tiedotettava ATK-työn edistymisestä ja mahdollisesti syntyneistä pulmista. Tämän lisäksi yrityksen johdon on osallistuttava aktiivisesti ATK-työhön, jotta menetelmän käyttöönotto sujuisi niin vaivattomasti kuin mahdollista. Yrityksen johdolla on mitä suurimmassa määrin syytä ilmaista myönteinen asenteensa ja mielenkiintonsa asiaan, koska kyseessä on yritykselle elintärkeän tietojenkäsittelyjärjestelmän perusteellinen muokkaus. Kysymyksissä, jotka koskevat koko yritystä, ja silloin kun asia koskee useaa osastoa, on johdon tehtävä suoranaaisia päätöksiä. Tällöin on kysymys mm. seuraavista asioista:*

*ATK-työn tavoitteiden määrittäminen pääpiirteittäin (esim. tietojenkäsittelyssä tavoiteltu integrointiaste).*

- *Kannanotto ehdotuksiin, jotka koskevat uuden järjestelmän vaatimia yleisperiaatteiden muutoksia.*
- *Eri osastojen välisten ristiriitojen ratkaiseminen.*
- *ATK-työn organisointi.*
- *Taloustaloudet, lähinnä henkilökunnan ja ajan uhraaminen tutkimus- ja valmistelutyöhön.*
- *Tietojenkäsittelyn koneellisten apuneuvojen valinta."*

### **Kokonaisuuden tarkastelu**

Kirjan kuvaamassa esitutkimuksessa asetetaan myös tavoitteet integrointiasteelle. Sen toisena ääripäänä nähdään nykyisten konttorikonerutiinien suoraviivainen automatisointi ja toisena tietojenkäsittelyn pitkälle menevä integrointi:

*"Sen sijaan, että jokaista toimintaa, tietojenkäsittelyaluetta, osastoa jne. tarkasteltaisiin erikseen, pyritään näitä rinnastamaan toisiinsa. Tämä merkitsee, että yritystä ja sen tietojenkäsittelyä tarkastellaan kokonaisuutena. Juuri ATK-menetelmän avulla voidaan toteuttaa tällainen kokonaisote yrityksessä tapahtuvaan tietojen varastointiin, kuljetukseen ja käsittelyyn. Vaikka ATK ei osoittautuisikaan parhaaksi vastaukseksi pulmiin, on tällainen tilanteen yleistarkastelu usein hyvin kannattavaa. Työn aikana saadaan näet viitteitä siitä, miten olevaa systeemiä voidaan yksinkertaistaa, kaksinkertainen työ poistaa jne. Monien mielestä tietokoneiden suurin etu se, että ne ovat saaneet aikaan tällaisen tietojenkäsittelyjärjestelmien kokonaistutkimuksen eri yrityksissä ja laitoksissa."*

### **Uudelleenkäyttö**

Termiä palveluarkkitehtuuri ei kirjassa vielä sellaisenaan esiinny. Kuitenkin kirjassa esitetään tietoteknisten ratkaisujen uudelleenkäyttöä tukevia ajatuksia seuraavasti:

*"Yrityksen tietojenkäsittelyjärjestelmä sopii harvoin sellaisenaan ATK-menetelmään siirrettäväksi. Sitä on tavallisesti jatkuvasti kehitetty vuosien mittaan lisäämällä ja parantamalla joitakin kohtia aina sen mukaan kuin tarve tai tehtävät ovat vaatineet. Tämän lisäksi siinä ilmenee usein voimakas desentralisointi siitä syystä, ettei aikaisemmin ole ollut koneellisia apuvälineitä keskitettyä tietojenkäsittelyä varten. Nykyinen osastojako on suurelta osin näiden rajoitusten sanelema. Kun ATK-menetelmää nyt suunnitellaan, on sisäänpäin suuntautuneesta ajattelutavasta vapauduttava ja tietojenkäsittelyä tarkasteltava ennakkoluulottomasti. Tässä voi systeemimies auttaa esittämällä uusia näkökohtia ja uusia ajatuksia vanhoista pulmista, mikäli hän itse ei ole liian läheisesti sitoutunut vanhaan systeemiin. Hänen täytyy pitää silmänsä auki mm. huomataksaan täyttää yrityksen eri puolilla esiintyvät samanlaiset tietojentarpeet yhden ja saman käsittelyrutiinin avulla. Nykyisiä osastorajoja ei siis saa pitää lopullisina esim. senkään vuoksi, että nyt on mahdollista päästä tietojenkäsittelyn laajaan integrointiin ja keskitykseen."*

Kirjassa tiedostetaan jo tarve kokonaisvaltaiselle lähestymistavalle ja uudelleenkäytettävien ratkaisujen tunnistamiselle ja jopa yrityksen toimintojen uudelleen organisoinnille.

## ATK:n hyödyt

Verrattuna tietotekniikan nykyiseen kaikkialle ulottuvaan vaikutukseen silloiset näkemykset ATK:n hyödyistä olivat vielä hyvin suppeita. Niitä olivat sivulla 369 esitelyinä mm.

- *Henkilökuntasäästöt rutiinitöiden automatisoinnin ansiosta*
- *Arvo siitä, että tiedot saadaan tuoreampina, luotettavampina ja täydellisimpinä, joka puolestaan mahdollistaa mm. varaston nopeampana kiertona ja toimitusten nopeutumisena. Myös parantuneille päätöksille on mahdollista laskea arvoa.*
- *Kehittyneemmät tutkimusmenetelmät matemaattisen ja tilastollisten menetelmien myötä.*
- *Käyttämättömän koneajan myynnistä saatavat tuotot.*
- *ATK-menetelmän mukanaan tuoma myönteinen PR-arvo.*

## 2.3 Arkkitehtuurityö

Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet koontidokumentti keskittyy pääasiassa esittelemään menetelmiä ja välineitä, mutta on tärkeätä todeta, että ne eivät ole itseisarvo, vaan niitä käyttämällä on tarkoitus aikaansaada jokin tuotos. Mikä on sitten arkkitehtuurityön tuotos? Kärjistäen voisi sanoa, että arkkitehtuurityön tuloksena syntyvät erilaiset mallit ja kuvaukset eivät ole se varsinainen tuotos, vaan nimityksensä mukaisesti kyseisen tuotoksen kuvauksia. Varsinaisen tuotoksen muodostaa arkkitehdin laatima ratkaisuehdotus sille, miten hänelle esitetyt erilaiset tarpeet sovitetaan yhteen.

Kokonaisarkkitehtuurityön olemuksen selvittämiseksi on tapana hakea vastaavuuksia toisilta toimialoilta. Usein vertailuna käytetään talon rakennusta, jossa arkkitehti laatii talon pohjapiirroksen, julkisivupiirroksen, asemapiirroksen jne. Arkkitehdin tehtävänä on koota tulevan talon käyttäjien tarpeet, ottaa huomioon asemakaavan, tontin, sijainnin, ympäristön ym. asettamat vaatimukset ja reunaehdot ja laatia ratkaisuehdotus sille, miten nämä sovitetaan yhteen. Usein tapana on arvioida arkkitehdin aikaansaannosta sen mukaan, miltä rakennus näyttää. Kuitenkin vähintään yhtä tärkeä arviointikriteeri on siinä, kuinka talo toimii, ts. kuinka talo täyttää ajatellun käyttötarkoituksensa. On siis hyvä todeta, että arkkitehdin laatimat piirustukset eivät ole hänen työnsä lopputulos vaan ainoastaan lopputuloksen kuvauksia.

Toinen usein käytetty vertailukohde kokonaisarkkitehtuurityölle on kaupunkisuunnittelu. Siinä laaditaan eri tason suunnitelmia maankäytöstä, erilaisista toiminnoista ja niiden sijoittelusta, liikenneväylistä, talojen sijoittelusta jne. Suunnitelmat ovat eri tasoisia, kuten asemakaava, osayleiskaava, yleiskaava, seutukaava jne. Kaava-arkkitehdin työn tuloksena ovat siis suunnitelmat maankäytöstä jne. joita sitten kuvaukset esittävät.

Vertailukohtia haettaessa on rakennus- ja kaava-arkkitehtuurin soveltuvuutta kritisoitu siinä suhteessa, että niiden tuloksena syntyvät rakennukset ja rakennettu ympäristö ovat luonteeltaan pysyviä suhteessa yritysten toimintaprosessien ja niissä käytettävien tietojärjestelmien pysyvyyteen. Tässä mielessä vertailukohtana voisi toimia esimerkiksi jokin tehdaskompleksi kaikkine koneineen ja prosesseineen. Uuden tehtaan rakentaminen lähtee ajatellusta liiketoimintamallista, tehtaan tuottamista tuotteista ja niiden aikaansaamiseksi tarvittavista prosesseista, koneista jne. Tuotannon käynnistymisen jälkeen markkinat muuttuvat, tuotteet kehittyvät ja muuttuvat, koneita uusitaan, prosesseja muutetaan ja sovitetaan uusiin vaatimuksiin. Kuitenkin muutokset aina tehdään olemassaolevan teh-

taan rakenteisiin ja prosesseihin sopiviksi laajennuksiksi. Suunnittelun lähtökohtana ovat olemassa olevan tehtaan ja koneistojen piirustukset, joita täydennetään ja muutetaan vastaamaan uusia tarpeita. Tehdasmaailmassa As-Is ja To-Be kuvaukset ovat itsestään selvyyksiä ja niiden ero selkeä. Tehdasanalogia saattaa muutosnopeutensa suhteen osua lähelle kokonaisarkkitehtuurityötä ja siltä odotettavia tuloksia.

## 2.4 Palvelupohjaisuuden perusteet

### Muutosten vaikutusten minimointi ja uudelleenkäyttö

Kautta aikojen ohjelmointityötä tekevien ammattikunnassa on ollut kaksi keskeistä tavoitetta:

- **Muutosten vaikutusten minimointi** eli miten ohjelmat jäsennellään ja rakenteistetaan siten, että ohjelmien elinkaaren aikana tehtävien muutosten vaikutuspiiri voitaisiin minimoida? Huonosti suunnitelluissa ja toteutetuissa ohjelmistoissa muutos johonkin kohtaan saattaa laukaista ketjureaktion, joka vaatii muutosten ulottamisen moneen eri kohtaan ohjelmistoa.
- Jo kerran ohjelmoitujen ratkaisujen hyödyntäminen eli **uudelleenkäyttö** seuraavissa ohjelmointitehtävissä? Uudelleenkäyttö lisää vähentää ohjelmointityön kustannuksia sekä lyhentää seuraaviin ohjelmointitehtäviin käytettävää aikaa.

Ohjelmointityössä on kiinnitetty näihin kysymyksiin huomiota jo 60-luvulta lähtien. Erilaisia tekniikoita ovat olleet mm. strukturoitu ohjelmointi, aliohjelmien käyttö, modulaarinen ohjelmointi, komponenttien käyttö, olio-ohjelmointi, hajautetut järjestelmät, client-server arkkitehtuurit, web services ja palveluarkkitehtuurit yleisemminkin. Palvelulähtöisessä ajattelussa taustalla siis vaikuttavat nämä ohjelmointityön keskeiset tavoitteet, joiden on huomattu soveltuvan toiminnan kehittämiseen yleisemminkin.

On mielenkiintoista havaita, että samansuuntaiset tavoitteet ovat olleet keskeisiä myös tiedonhallinnan kehityksessä. Relaatiomallin ja varsinkin normalisoinnin tavoitteena on pyrkiä eristämään tietokantojen rakenteisiin tehtävien muutosten heijastusvaikutukset mahdollisimman rajatulle alueelle. Toisaalta hyvin normalisoitu tietokanta antaa mahdollisuudet eri taulujen tietojen käyttöön useissa eri ohjelmissa.

Ohjelmista ja tietokannoista eteenpäin kehitettynä palveluarkkitehtuuri voi tarkoittaa myös toimintaprosessien ja toiminnallisten palveluiden jäsentämistä pysyvien ja uudelleen käytettävien rakenteiden löytämiseksi. Esillä olleita ajatuksia ovat olleet mm. liiketoimintamallien ja liiketoiminnan infrastruktuurin erottaminen toisistaan siten, että pysyvämmät elementit muodostaisivat rakenteet, joiden tukemana liiketoiminnan nopeatempoiset muutokset olisivat mahdollisia. Palvelupohjaisuus (Service Oriented Architecture, SOA) onkin hyvin laajalle omaksuttu arkkitehtuurin laatimismalli.

### SOA manifesti ajatukset.

SOA manifesti (<http://www.soa-manifesto.org/>) on luotu SOA Manifesto Working Groupin (yhtenä vetäjänä mm. Thomas Erl, joka on laatinut aiheesta useita kirjoja) toimesta selkeyttämään näkemyksiä siitä, mitä palvelupohjaisella arkkitehtuurilla tarkoitetaan. Työ pohjaa havaintoon, että SOA:n olemuksesta löytyy sen esittäjästä riippuen useita suhteellisen samansuuntaisia, mutta silti eriäviä näkemyksiä ja mielipiteitä. Tavoitteena oli luoda perusnäkemys SOAsta, vastaavaan tyyliin kuin esimerkiksi Agile Manifesto luo raamit ketterille ohjelmistokehitysmenetelmille (<http://agilemanifesto.org/>), asettamalla käsitteeseen liittyvät arvot (*values*) ja niitä kohti ohjaavat periaatteet (*guiding principles*) käsitteelle.

Seuraavassa lyhyt tiivistys/esitys SOA-manifestistä:

**Alustus** (kirjoitettu ”Me toteutamme...” -muodossa):

Palvelupohjainen arkkitehtuuri on tulosta palvelukeskeisyyden (*service orientation*) soveltamisesta. Palvelukeskeisyyden soveltaminen edesauttaa organisaatiota tuottamaan kestäväää liiketoimintaa arvoa ketterästi ja kustannustehokkaasti sekä mukautumaan muuttuviin liiketoimintatarpeisiin.

**Arvot ja niiden priorisointi** (*Values*, kirjoitettu muotoon ”Vaikka arvostamme arvoja listan oikealla puolella, arvostamme enemmän arvoja vasemmalla puolella” ts. lihavoidut arvot)

- **liiketoiminta-arvo** ennen teknistä strategiaa => *liiketoiminta ohjaa IT:tä*
- **strategiset tavoitteet** ennen projektikohtaisia hyötyjä => *projektikohtaisuus johtaa tyypillisesti siilomaiseen rakenteeseen*
- **luontainen** (intrinsic ~natiivi) **yhteentoimivuus** ennen tapauskohtaisia integraatioita => *yhtenäisyyden tavoittelu eri tasoilla mm. teknologia, tieto ja liiketoiminta?*
- **jaetut palvelut** ennen tapauskohtaisia implementaatioita
- **joustavuus** ennen optimointia
- **asteittainen/evoluutiomainen kehittäminen** ennen suoraviivaista täydellisyyden tavoittelua ts. järjestelmät elävät liiketoiminnan mukana ja siihen mukautuen, liiketoiminnan tulevaisuutta ei voi kukaan tietää

**Ohjaavat periaatteet** (*Guiding Principles*, kirjoitettu muotoon ”Me noudatamme seuraavia periaatteita.”)

- kunnioita organisaation sosiaalisia- ja valtasuhteita
- tunnusta tosiasia, että SOA vaatii muutoksia useilla tasoilla
- SOA:n käyttöönoton laajuus voi vaihdella - pidä asetetut tavoitteet hallittavan kokoisina ja järkevästi rajattuina
- tuotteet ja standardit eivät itsessään tuota SOAa tai sovelta palvelupohjaisuutta
- SOA on mahdollista toteuttaa useiden (vaihtoehtoisten) eri teknologioiden ja standardien avulla
- koosta ja toteuta yhtenäinen joukko organisaatiotason standardeja ja käytänteitä, jotka pohjaavat teollisuus-, de facto- ja toimialastandardeihin
- tavoittele yhtenäisyyttä ulospäin samalla sallien monimuotoisuus sisäisessä toteutuksessa
- tunnista palvelut yhteistyössä liiketoiminta- ja teknologiasidosryhmien kanssa
- maksimoi palveluiden käyttö ottamalla huomioon niiden tämän hetkisen ja tulevan hyödyntämisen
- varmista, että palvelut toteuttavat liiketoiminnan vaatimukset ja tavoitteet
- kehitä palveluita ja niiden organisointia/koostamista, siten että ne vastaavat todellista käyttöä
- erota eri järjestelmien osa-alueet, jotka muuttuvat eri tahdissa
- vähennä epäsuoria (sisäisiä)riippuvuuksia ja julkista kaikki ulkoiset riippuvuudet (esim. rajapinnat jne.), joiden avulla järjestelmien vakaus lisääntyy ja muutosten vaikutuksia saadaan pienennettyä
- järjestele palvelut hallittaviin ja yhtenäisiin kokonaisuuksiin niiden toiminnallisuuden perusteella jokaisessa abstraktiokerroksessa



Edellä listatut arvot ja periaatteet eivät kaikki ole kaikille itseään selittäviä, mitä korjaamaan Thomas Erl on tuottanut myös manifestin niukkaa ilmaisua selittävän The Annotated SOA Manifeston (<http://www.soa-manifesto.com/annotated.htm>).

### Arvioita:

SOA-manifestin hyöty itsenäisenä (selkeyttää käsitystä SOAsta) on kyseenalainen eli sille joudutaan päätekijän toimesta tuottamaan selittävä dokumentaatio. Yllä kuvattuja termejä voi tulkita monella tavalla, kuten ”intrinsic interoperability” ym.

Sinällään materiaalissa ei valtavasti SOA-kohtaisia erityispiirteitä? Harjoitelmana: korvaa esim. SOA => arkkitehtuuri tms. muulla vastaavalla/palvelut => komponentti tms. Kuitenkin kaikki periaatteet ovat hyvin soveltuvia hyvän tietojärjestelmän / järjestelmäarkkitehtuurin ja yritysarkkitehtuurin (soveltuvuus esim. architecture principles -kategoriaan) rakennusperiaatteiksi ja arvoiksi – ilman tai kanssa SOAn.

Manifestin arvot ja periaatteet ovat sinänsä arvokkaita ja tavoiteltavia, mutta se jättää vielä konkreettiset ohjeet antamatta: Miten minimoidaan muutosten vaikutukset ja miten maksimoidaan uudelleenkäyttöä?

## 2.5 SOLEAn näkökulma kokonaisarkkitehtuuriin

### Tarve kokonaisvaltaiselle lähestymistavalle

Mikä on syynä kokonaisarkkitehtuurin suosion kasvuun nyt 2000-luvulla, jos kerran samoja ajatuksia on esitetty oppikirjoissa jo 1960-luvun lopussa. Mikä on muuttunut niistä ajoista? Ehkä joitakin seikkoja ovat mm. seuraavat:

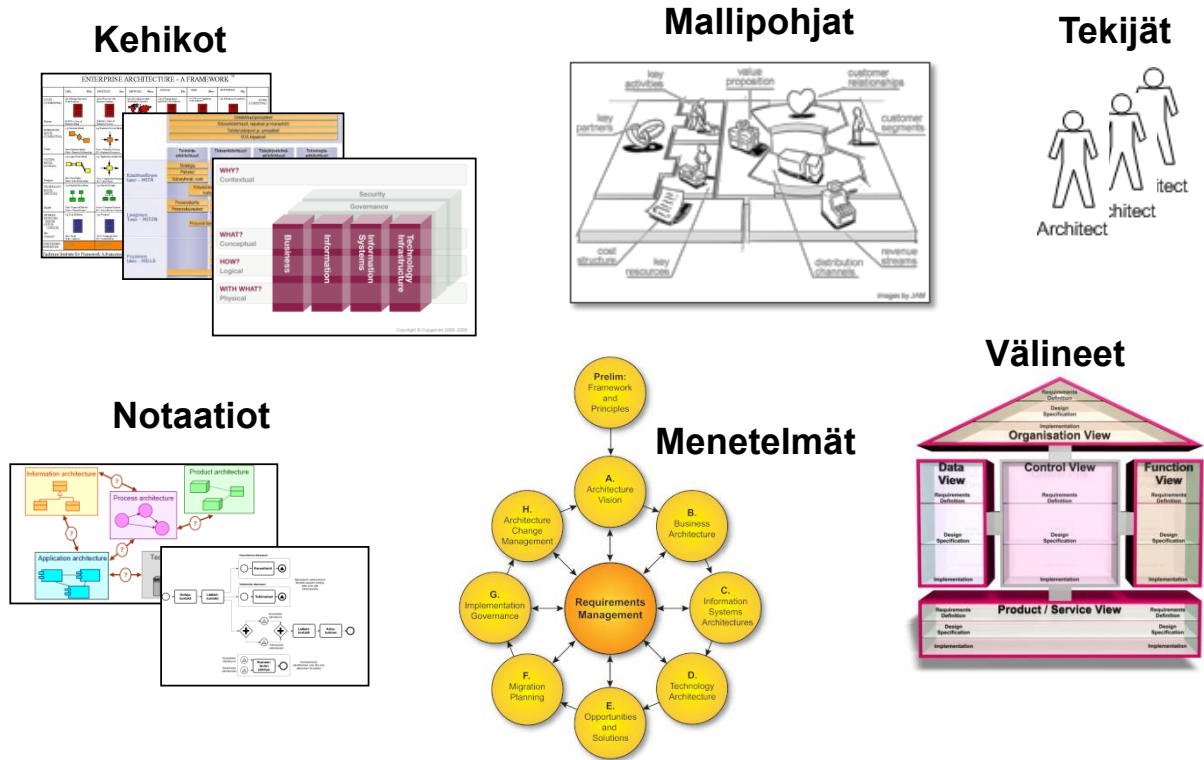
- Tieto- ja kommunikaatioteknologian (ICT) valtava kehitys erityisesti PC:n, Internetin ja mobiiliteknologian myötä. Nykyään yrityksissä lähes kaikki työntekijät käyttävät tietotekniikkaa jossain muodossa, esimerkiksi sähköpostia.
- Ohjelmistotuotannon siirtyminen yritysten omista osastoista varus- ja valmisohjelmistoja valmistaville yrityksille. Tämä on osaltaan aikaansaanut myös tietoteknisten ratkaisujen toteuttamisen eriyttämisen liiketoiminnan edustajien ulottumattomiin.
- Tietokoneiden siirtyminen konesaleista palvelukeskuksiin
- Tietotekniikan hyödyntämisen painopiste on siirtynyt henkilötyön automatisoinnista toiminnan ohjaukseen ja tuottavuuden lisäykseen, asiakaspalveluun ja kilpailuedun saamiseen, tietotekniikalla toteutettujen palveluiden tuottamiseen jne.
- Samalla myös yrityksessä käytössä olevien järjestelmien ja tietokantojen määrä on huimasti kasvanut ja toisaalta järjestelmien hankinta on usein hajautunut liiketoimintayksiköille keskitetyn tietohallinnon sijasta

Voimakas kehitys teknologiassa ja ohjelmistotuotannossa ovat toisaalta johtaneet tilanteeseen, jossa yritysten käytössä olevat tietojärjestelmät ovat yhteensopimattomia ja joiden ylläpito on aikaa vievää ja nielee suurimman osan atk-budjetista. Toisaalta yritysten toimintaympäristön muuttuminen

## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

yhä nopeammin aikaansaava paineita myös tietojärjestelmien sopeutumiselle ja uusiutumislle. Ilmeistä on, että nämä paineet aikaansaavat entistä suuremman tarpeen tietotekniikan kokonaisvaltaiselle kehittämiselle ja hallinnolle yrityksissä.

### Kokonaisarkkitehtuurin osa-alueet SOLEA-projektissa



Kuva 2: Kokonaisarkkitehtuurin osa-alueet

Kokonaisarkkitehtuurin tarkastelu jäsennellään tässä dokumentissa seuraaviin oheisen kuvan mukaisesti osa-alueisiin, jotka voidaan tunnistaa kirjallisuudessa usein esiintyviksi:

- **Kehikot:** tapa jäsentää kokonaisarkkitehtuuria tai sen osa-alueita (myös tiettyyn näkökulmaan keskittyvät viitearkkitehtuurit), esim. näkökulmat + tasot. Esimerkiksi Zachman Enterprise Architecture Framework
- **Menetelmät:** tapa vaiheistaa tekemistä, määritellä tehtäviä. Esimerkiksi TOGAF, IAF, DoDAF
- **Notaatiot:** tapa kuvata tietyillä kaavioilla tai rakenteella. Esimerkiksi UML, ArchiMate, BPMN, SoaML
- **Mallipohjat ja arkkitehtuurikuvaukset :** konkreettiset kuvaukset = kuvauksen kohde + mahdollisesti notaatio. Esimerkiksi Business Model Canvas, Enterprise Architecture for Telecom sector
- **Välineet:** millä konkreettisilla välineillä kuvauksia tehdään, jaetaan, työtä hallinnoidaan jne. Esimerkkejä ovat mm. Powerpoint, Visio, ARIS, Sparx, Bizdesign jne.

- **Tekijät ja tarvittava osaaminen ja kyvykkyydet:** millaisia kykyjä ja rooleja tekijöillä (ihmiset) on oltava arkkitehtuurityön eri osa-alueilla. Esimerkkejä ovat mm. yritysarkkitehti, sovellusarkkitehti, prosessiarkkitehti jne.

Mielenkiintoista on, että palveluarkkitehtuuri ei useinkaan esiinny osana kokonaisarkkitehtuuria, vaan sitä käsitellään sille omistautuneissa kirjoissa ja artikkeleissa. Tässä dokumentissa pyritään palveluarkkitehtuuri pitämään mukana osana kokonaisarkkitehtuuria kullakin osa-alueella.

Paikallisuus näkökulma suhteessa osa-alueisiin tulee saada näkyville (kohdealuearkkitehtuuri, segmenttiarkkitehtuuri (FEAF maaliskuu 2011 työpaja käsitelty segmenttiarkkitehtuuria)).

## 2.6 Dokumentin jäsenitys

Tämä kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet -dokumentti jäsenyy ylläolevan lähestymistavan mukaisesti. Jokaista osa-aluetta käsitellään omassa luvussaan ja lopussa tehdään yhteen vetoa.

## 3 Arkkitehtuurikehikot

Julkisen hallinnon tietohallinnon neuvottelukunnan JSH 179 suosituksessa kirjoitetaan seuraavasti arkkitehtuurikehityksen tarkoituksesta: *”Arkkitehtuurikehityksellä tarkoitetaan kokonaisarkkitehtuurin jäsenitysmallia, joka tarjoaa näkökulmia ja lähestymistapoja kokonaisuuden hahmottamiseksi ja jäsentämiseksi paremmin käsiteltävään ja ymmärrettävään muotoon. Kehityksen avulla tunnistetaan kehittämisessä huomioonotettavia näkökulmia ja asioita riittävän kokonaiskuvan saamiseksi sekä tietojen ja kokonaisuuteen vaikuttavien rakenteiden välisten suhteiden selvittämiseksi.*

*Kehityksen tehtävä on nostaa esiin niitä kysymyksiä, joita kokonaisarkkitehtuurin tai valitun ja rajatun kokonaisarkkitehtuurin osa-alueen kehittämisessä on huomioitava ja tarkasteltava. Se rajaa mm. tarkasteltavia arkkitehtuurinäkökulmia, organisatorista kattavuutta ja tarkkuustasoa (hierarkiatasot) sekä suunnittelun abstraktio- eli käsitetasoja” (JHS 179).*

Tässä luvussa esitellään lyhyesti eri kokonaisarkkitehtuurikehikoita, niiden rakennetta ja sisällön ajatusta.

### 3.1 JHS 179

JHS 179 kokonaisarkkitehtuurikehitys on osa ICT-palvelujen kehittäminen -suositussarjaa ja se täydentää muita sarjan suosituksia antamalla pohjan suosituksissa kuvatulle arkkitehtuuriohjatulle organisaation toiminnan kehittämiselle.

Muita ICT-palvelujen kehittäminen -sarjan suosituksia ovat:

- JHS 171 ICT-palvelujen kehittäminen: Kehittämiskohteiden tunnistaminen.
- JHS 172 ICT-palvelujen kehittäminen: Esiselvitys.
- JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely.

JHS 179 kokonaisarkkitehtuurikehys koostuu neljästä eri arkkitehtuurinäkökulmasta: toiminta-, tieto-, tietojärjestelmä- ja teknologia-arkkitehtuurinäkökulman. Lisäksi jokainen näkökulma jakaantuu kolmeen kuvattavaan tasoon: käsitteellinen, looginen ja fyysinen. JHS 179 kokonaisarkkitehtuurikehys pohjautuu TOGAF arkkitehtuurikehykseen.

JHS 179 kokonaisarkkitehtuurikehys on julkaistu 2011 ja sen julkaisija on JUHTA – julkisen hallinnon tietohallinnon neuvottelukunta. JHS 179 on jaettu osoitteessa: <http://www.jhs-suositukset.fi/web/guest/jhs/recommendations/179> on vapaasti kaikkien hyödynnettävissä.

Lähde: <http://www.jhs-suositukset.fi/web/guest/jhs/recommendations/179>

### 3.2 Kartturi 2.0

Kartturi 2.0 kehys on kehitetty korkeakoulusektorin organisaatioiden – korkeakoulujen ja yliopistojen sekä korkeakoulujen yhteistyöorganisaatioiden kuten tietyn maantieteellisen seudun tai tietyn kohdealueen korkeakoulusektorin toiminnan kehittäjien ja tietohallinto-organisaatioiden käyttöön. Kartturi arkkitehtuurimenetelmää suositellaan käytettäväksi sekä paikallisesti yksittäisissä korkeakouluissa että soveltaen koko korkeakoulusektorilla. Korkeakoulutoimijoiden omien asiantuntijoiden lisäksi suositellaan, että tätä korkeakoulusektorin arkkitehtuurimenetelmää soveltavat korkeakouluille tietojärjestelmiä, IT-palveluja, konsultointi- ja asiantuntijapalveluja ja/tai kehittämispalveluja tarjoavat julkisen ja yksityisen sektorin IT-palveluntuottajat.

Keskeisimmät Kartturi-kehysten taustalla olevat kokonaisarkkitehtuurikehykset ovat:

- Julkishallintoa koskeva kokonaisarkkitehtuurin JHS 179 -suositus
- Valtionhallinnon kokonaisarkkitehtuurimenetelmä
- Korkeakoulusektorin kokonaisarkkitehtuurimenetelmä
- Togaf kokonaisarkkitehtuurikehys (The Open Group Architecture Framework)
- Muut kansainvälisesti laajalti hyödynnettävät KA-kehukset (esim. IAF, Zachman Architecture Framework, FEA-framework)

Kartturi 2.0 kokonaisarkkitehtuurikehys koostuu neljästä eri arkkitehtuurinäkökulmasta: toiminta-, tieto-, tietojärjestelmä- ja teknologia-arkkitehtuuri. Lisäksi jokainen näkökulma jakaantuu kolmeen kuvattavaan tasoon: käsitteellinen, looginen ja fyysinen. Kartturi-kehysten tyypillisiä SOA-yhteyksiä ovat mm. arkkitehtuuriperiaatteet ja tietojärjestelmäpalvelujäsennys, joka liittyy läheisesti SOA-periaatteiden mukaiseen kehittämiseen.

Kartturi-arkkitehtuuriviitekehys on yhteensopiva kokonaisarkkitehtuuria koskevan JHS-suosituksen kanssa. Kyseistä viitekehystä on täsmennetty korkeakoulusektorille soveltuvaksi sektorin erityispiirteiden ja kokonaisarkkitehtuurikehysten soveltamisesta saatujen kokemusten pohjalta.

Kartturi kokonaisarkkitehtuuri on osa RAKETTI hanketta. Kartturi kokonaisarkkitehtuurimallin ja siihen liittyvä materiaali on vapaasti ladattavissa osoitteessa: <http://raketti.csc.fi/kokoa/kartturi>.

### 3.3 TOGAF™ Version 9

TOGAF (*The Open Group Architecture Framework*) on yksityiskohtainen metodi ja joukko kokonaisarkkitehtuurityötä tukevia työkaluja, jota voidaan käyttää vapaasti organisaation kokonaisarkkitehtuurin kehittämiseen. TOGAF perustuu iteratiiviselle prosessimallille (Architecture Development Method), jota tuetaan parhailla käytännöillä ja uudelleen käytettävillä arkkitehtuurityökaluilla. TOGAF on tällä hetkellä yksi yleisimmin käytetyistä arkkitehtuurikehyksistä.

TOGAF on korkean tason ja kokonaisvaltainen lähestymistapa kokonaisarkkitehtuurisuunnitteluun ja on kokonaisarkkitehtuureille tyypillisesti mallinnettu neljään näkökulmaan (*toiminta, tieto, tietojärjestelmä ja teknologia*). TOGAF nojaa vahvasti modulaarisuuteen, standardointiin ja olemassa oleviin todistettuihin teknologioihin ja tuotteisiin. Lisäksi TOGAF:n näkymät voidaan spesifioida formaaliksi kuvaukseksi järjestelmästä tai yksityiskohtaiseksi suunnitelmaksi systeemistä komponenttien tasolla. TOGAF:n ydin on ADM metodi – the TOGAF Architecture Development Method, josta kerrotaan tarkemmin luvussa 5 kokonaisarkkitehtuurimenetelmät. TOGAF käyttää pääasiassa ISO/IEC 42010:2007 terminologiaa (The Open Group 2009).

TOGAF tarjoaa valmiin kehyksen ja metodologian kokonaisarkkitehtuurin kehittämiseen. TOGAF suunniteltiin alkuperäisesti tukemaan teknologia-arkkitehtuurin hallintaa läpi jatkuvan evoluution. Uudempien versioiden myötä TOGAF:sta on kehittynyt kokonaisvaltainen arkkitehtuurikehys ja metodologia, joka tukee koko organisaation strategista suunnittelua ja ohjausta, jossa liitetoimintatapahtumat ja tietohallinto otetaan huomioon koko yrityksen kehityksen kannalta (Jin, Kung, Peng 2010).

TOGAF:a voidaan käyttää useiden erilaisten kokonaisarkkitehtuurien kehittämiseen. Lisäksi TOGAF:a voidaan käyttää yhdessä muiden yksityiskohtaisempien arkkitehtuurikehyksien kanssa (esim. Zachman Framework, SOA Reference Architecture), jotka kohdistuvat tarkemmin tietyille sektorille (esim. julkishallintoon, terveydenhuoltoon) tai tiettyyn TOGAF:n näkökulmaan.

Esimerkiksi TOGAF:a voidaan hyödyntää SOA suunnittelussa ja käyttöönotossa. Tällaisissa tapauksissa TOGAF-menetelmää on muokattava paremmin SOA:an sopivaksi, jotta kehys palvelisi asetettuja tavoitteita paremmin. Lisää tietoa TOGAF:n ja SOA:n yhteensovittamisesta, löytyy The Open Group:n sivuilta osoitteesta: <http://www.opengroup.org/soa/source-book/intro/index.htm>.

TOGAF 9 arkkitehtuurikehys koostuu seuraavista komponenteista (The Open Group 2009):

- Architecture Development Method (ADM)
- ADM Guidelines and Techniques
- Architecture Content Framework
- Enterprise Continuum and Tools
- TOGAF Reference Models
- Architecture Capability Framework

TOGAF:a ylläpitää ja kehittää The Open Group Forumin jäsenet. TOGAF:n ensimmäinen versio julkaistiin vuonna 1995, joka perustui yhdysvaltojen puolustusministeriön TAFIM (*Technical Architecture Framework for Information Management*) kehykseen. Tähän päivään (2011) mennessä

TOGAF on ehtinyt jo yhdeksänteen julkaisuunsa. TOGAF versiot julkaistaan The Open Group:n julkisilla nettisivuilla ([www.opengroup.org](http://www.opengroup.org)) ja ne ovat vapaasti käytettävissä kaikille halukkaille (The Open Group 2009).

### 3.4 Zachman Framework

Zachmanin kokonaisarkkitehtuurikehys on eksakti ja käytännöllinen malli ohjaamaan yritysten/valtion virastojen strategiaa kokonaisarkkitehtuurin rakentamisessa organisoidusti ottaen huomioon kokonaisvaltaisesti organisaation eri sidosryhmät [Zac96]. Zachmanin kehys ei tarjoa suoraan ohjeistusta siihen, kuinka organisaation kokonaisarkkitehtuuri tulee rakentaa, millaisia prosesseja kokonaisarkkitehtuurityö sisältää ja kuinka osa-alueet tulee kuvata. Zachmanin kehys on lähinnä taksonomia organisoimaan arkkitehtuurisia artefakteja (esim. suunnitteludokumentit ja -mallit) (Sessions 2007).

Zachman itse kuvaa kehystä Roger Sessionsin haastattelussa (Sessions 2007) seuraavin sanoin: *”Soveltaessa kehystä yritykseen, se on yksinkertaisesti vain looginen rakenne luokitella ja organisoida kuvauksia, joilla on merkitystä yrityksen johdolle ja järjestelmien kehittäjille”*.

Zachmanin kehysten vahvuus on, että kehys on yksinkertainen ymmärtää, koska se ei ole teknillinen, vaan täysin looginen. Kehys tarjoaa kuvauksia useasta eri näkökulmasta, kattaa koko organisaation ja kuvaa sitä tarkasti, mutta ei-teknisesti. Kehys on suunnittelu ja ongelmanratkaisu työkalu, joka auttaa saamaan selkeän kuvan koko organisaatiosta ja mahdollistaa työskentelyn abstraktiotsalla ja yksinkertaistaa asiat kuitenkin hävittämättä monialaista kokonaiskuvaa organisaatiosta.

Kehys on myös neutraali, jonka ansioista sen ohella voi hyödyntää useita eri metodeja, työkaluja ja notaatioita. Esimerkiksi hyödyntäessä Zachmanin kehystä SOA:n suunnittelussa, tarjoaa kehys arkkitehteille hyvän työkalun ymmärtää ja organisoida ihmisten ja järjestelmien suhteita, jolloin organisaatio voi vastata paremmin muutoksiin.

Oikein käytettynä kehys parantaa ammatillista kommunikaatiota sidosryhmien kesken ja lisää ymmärrystä syistä ja riskeistä, joita liittyy arkkitehtuuriseen kuvaukseen. Lisäksi kehysten käyttö mahdollistaa parempien lähestymistapojen ja arkkitehtuuristen esitysten tuottamisen ja antaa mahdollisuuden uudelleen miettiä organisaation tapoja tehdä palveluita ja sovelluksia. Zachmanin kehys ei ole metodologia sovellusten luontiin, se on ainoastaan rakenne ja lähtökohta kokonaisarkkitehtuurityölle (Zachman, 2012).

Zachman Framework on kaupallinen arkkitehtuurikehys, jonka omistaja ja ylläpitäjä on Zachman International. Lisätietoa osoitteessa: [www.zachman.com](http://www.zachman.com)

Zachmanin käyttämää jäsenystä käsitellään tarkemmin luvussa 4.2.

## 4 Kokonaisarkkitehtuurin jäsentämismallit

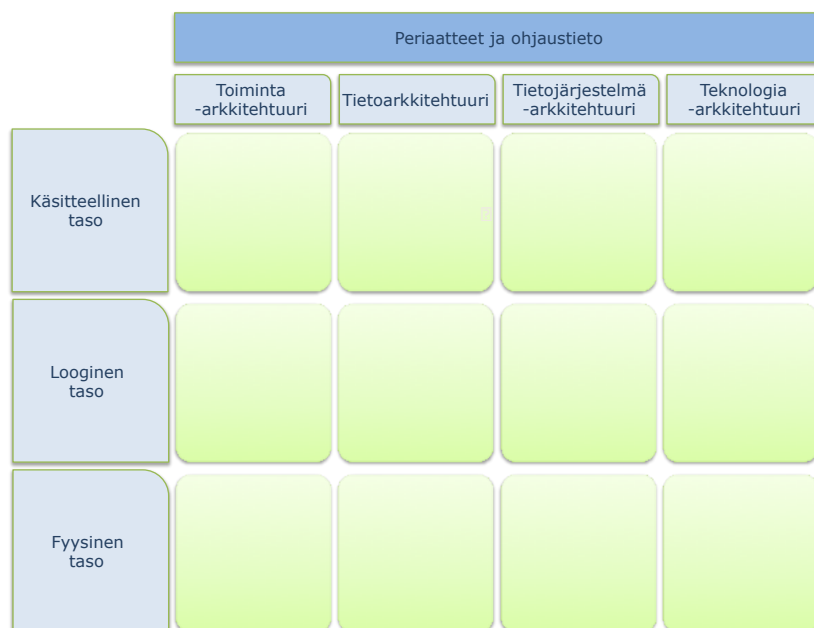
Tietojärjestelmien suunnittelijat, rakentajat, käyttäjät ja ylläpitäjät ovat kautta aikojen käyttäneet erilaisia kuvauksia työnsä tukena. Kulkukaaviot, lohkokaaviot, käsitemallit, tietomallit jne. ovat olleet tyypillisiä kuvaustapoja. Tarkoituksena on esittää ja havainnollistaa miten toimijat, tiedot, tehtävät ja niiden suoritusjärjestys liittyvät toisiinsa.

Zachmanin laatima kehikko oli ensimmäisiä eri kuvausten jäsentämismalleja. Hän havaitsi, että tietojärjestelmien rakentamiseen voidaan soveltaa muilla toimialoilla perinteisesti käytettyjä ja hyviksi osoittautuneita käytäntöjä. Hänen keskeinen ajatuksensa oli jäsentää kuvaukset siten, että ne muodostavat yhtenäisen ja toisiaan täydentävän mallin tarkasteltavasta kokonaisuudesta. Kuvausten jäsentely lähtee toisaalta eri osapuolten näkökulmista ja toisaalta kuvausten kohteista. Jäsentely esitetään kaksiulotteisena taulukkona, jossa riveinä ovat osapuolet ja sarakkeina kuvausten kohteet.

Vastaavia kehikoita on myöhemmin esitetty useita. Yhteistä niille on, että rivien esittämät kuvaukset tarkentuvat mentäessä ylhäältä alaspäin ja sarakkeet sisältävät toisiaan täydentäviä kuvauksia eri kohteista. Paljon käytetyksi jäsentämismalliksi on muodostunut kolmen rivin ja neljän sarakkeen jäsenys, jota myös SOLEA:ssa käytetään.

### 4.1 SOLEA:ssa käytettävä jäsenys

SOLEA-projektissa käytetään oheista jäsentämismallia, joka on yleisesti käytössä hiukan eri muodoissaan useissa kokonaisarkkitehtuurikehikoissa. Malli koostuu neljästä näkökulmasta (toiminta, tieto, tietojärjestelmä, teknologia) sekä kolmesta päätasosta joiden lisäksi voidaan tunnistaa (käsitteellinen, looginen, fyysinen sekä periaatteet ja ohjaus). Mallin tulkinnat ja soveltamistapa pohjautuvat kirjallisuuteen sekä projektin työpajoissa ja eri työkohteissa tehtyihin tarkennuksiin.



Kuva 3: Kokonaisarkkitehtuurin näkökulmat ja tasot.

Mallin tasot (käsitteellinen, looginen, fyysinen) ovat yleisesti kokonaisarkkitehtuurikehikoista löytyviä. Tasojen periaatteena on, että ylemmällä tasolla kuvattavat seikat ovat abstraktiotasoltaan ylempänä kuin alemmalla tasolla kuvattavat seikat. Yksityiskohtien lukumäärä lisääntyy ylemmältä tasolta alemmalle siirryttäessä. Aina jäsentämismallia käytettäessä on kuitenkin määriteltävä säännöt sille, mitä kullakin tasolla käytännössä tarkoitetaan ja millä tarkkuustasolla kunkin tason kuvauksia laaditaan. Yksinkertaiset peukalosäännöt kuten "Käsitteellinen- Mitä", "Looginen - Miten", "Fyysinen - Millä" eivät yleensä riitä yhteisen käsityksen muodostamiseen, vaan on myös tarkennettava ja rajattava, mikä kokonaisuus arkkitehtuurityön ja kuvausten kohteena on. Mallin hyödyntämisen peukalosääntö on "ylhäältä alas ja vasemmalta oikealle", eli käsitteellisen tason tulisi ohjata loogisen ja fyysisen tason muodostumista, ja toiminta-arkkitehtuurin ohjata tieto-, tietojärjestelmä- ja teknologia-arkkitehtuuria. Seuraavassa esitetään SOLEA:ssa muodostettu näkemys näistä tasoista, **kun mallia käytetään organisaation tai kohdealueen kokonaisarkkitehtuurin jäsentämisen välineenä**. Mallia on mahdollista soveltaa myös tietyn (pienemmän) kehittämiskohteen sisäisessä jäsentämisessä, jolloin tasojen merkitykset on tarkennettava suhteessa kyseiseen kohteeseen.

- *Periaatteet ja ohjaus -tasolla* esitellään aina keskeisimmät koko arkkitehtuurin kehittämistä ohjaavat periaatteet ja kuvaukset, kuten arkkitehtuuriperiaatteet. Tasolle sijoitettavat kuvaukset ja linjaukset koskevat kaikkia arkkitehtuurinäkökulmia. Lähtökohdat ja perustelut kuvattavalle kokonaisarkkitehtuurille voidaan vaihtelevasti sijoittaa käsitteelliselle tai periaatteet ja ohjaus-tasolle. Esimerkiksi mikäli yrityksen tai organisaation liiketoimintastrategiat nähdään arkkitehtuuriohjauksen kautta, ne voivat sijoittua periaatteet ja ohjaus -tasolle Jos ne taas nähdään toiminta-arkkitehtuurin kautta muita arkkitehtuurinäkökulmia ohjaavana, luonteva sijoituspaikka on käsitteellisen tason toiminta-arkkitehtuuri.
- *Käsitteellisellä tasolla* kuvataan kokonaiskuvaa, pääsisältöjä ja tärkeimpiä suhteita. Käsitteellisen tason nimi ei tarkoita, että kaikki käsitteelliset asiat tulisi sijoittaa käsitteelliselle tasolle, vaan myös loogisella ja fyysisellä tasolla on niille kuuluvia käsitteitä ja käsitteellisiä asioita. Käsitteellisen tason kuvauksia voidaan suhteuttaa esimerkiksi Zachman-mallin ylimpien roolien (planner, business owner) tarvitsemiin ja käyttämiin kuvauksiin, ja keskeistä on pyrkiä luomaan ymmärrettäviä yleiskuvauksia ja välineitä tarkempien tasojen määrittelyyn. Käsitteellisen tason kuvauksissa ei kuvata toteutustapoja.
- *Loogisella tasolla* kuvataan tunnistettuja rakenteita, prosesseja, käsitteellisen tason elementtien jäsentymistä sekä ratkaisumalleja, ja mukana on käsitteellistä tasoa enemmän palvelujen välisiä suhteita, yhteyksiä ja esimerkiksi lukumääräsuhteita.
- *Fyysisellä tasolla* kuvataan ratkaisujen sijoittelua, toteuttamista ja hallinnointia konkreettisella tasolla. Kuvauksissa käytetään usein erisnimiä ja esitetyille kohteille voidaan antaa myös yksilöivät tunnisteet myös tietojärjestelmien tai esimerkiksi verkon solmujen tasoilla.

Käytännössä eri tasojen yksiselitteinen erottaminen on vaikeaa, mikäli kuvaamisen kohdetta ja sen karkeustasoa ei ole selkeästi sovittu. Sama asia voi näkyä mallin eri tasoilla riippuen siitä, mikä on kuvauksen kohde. Esimerkiksi yksittäinen tietojärjestelmä voi kokonaisarkkitehtuuria kuvattaessa näkyä jäsentämismallin loogisella tasolla tiettyä toimintakokonaisuutta hoitavana järjestelmänä sekä fyysisellä tasolla konkreettisena tuotteena jota käytetään eri toimipisteissä. Mikäli samaa jäsenystä käytetään toimintakokonaisuuden tarkempaan kuvaamiseen, tietojärjestelmä voidaan tunnistaa ja yksilöidä jo käsitteellisellä tasolla, ja looginen ja fyysinen taso voivat tarkentaa järjestelmän tai sen eri osien ominaisuuksia tai vaatimuksia. Tasojen soveltamisen käytännön ohjeena voisi pitää sitä, että on lähdettävä kuvattavan kokonaisuuden sekä kuvausten käyttäjien ja käyttötarkoitusten (arkki-



tehtuurin kontekstin) määrittelystä ja pyrittävä tekemään kuvauksista mahdollisimman ymmärrettäviä tässä kontekstissa.

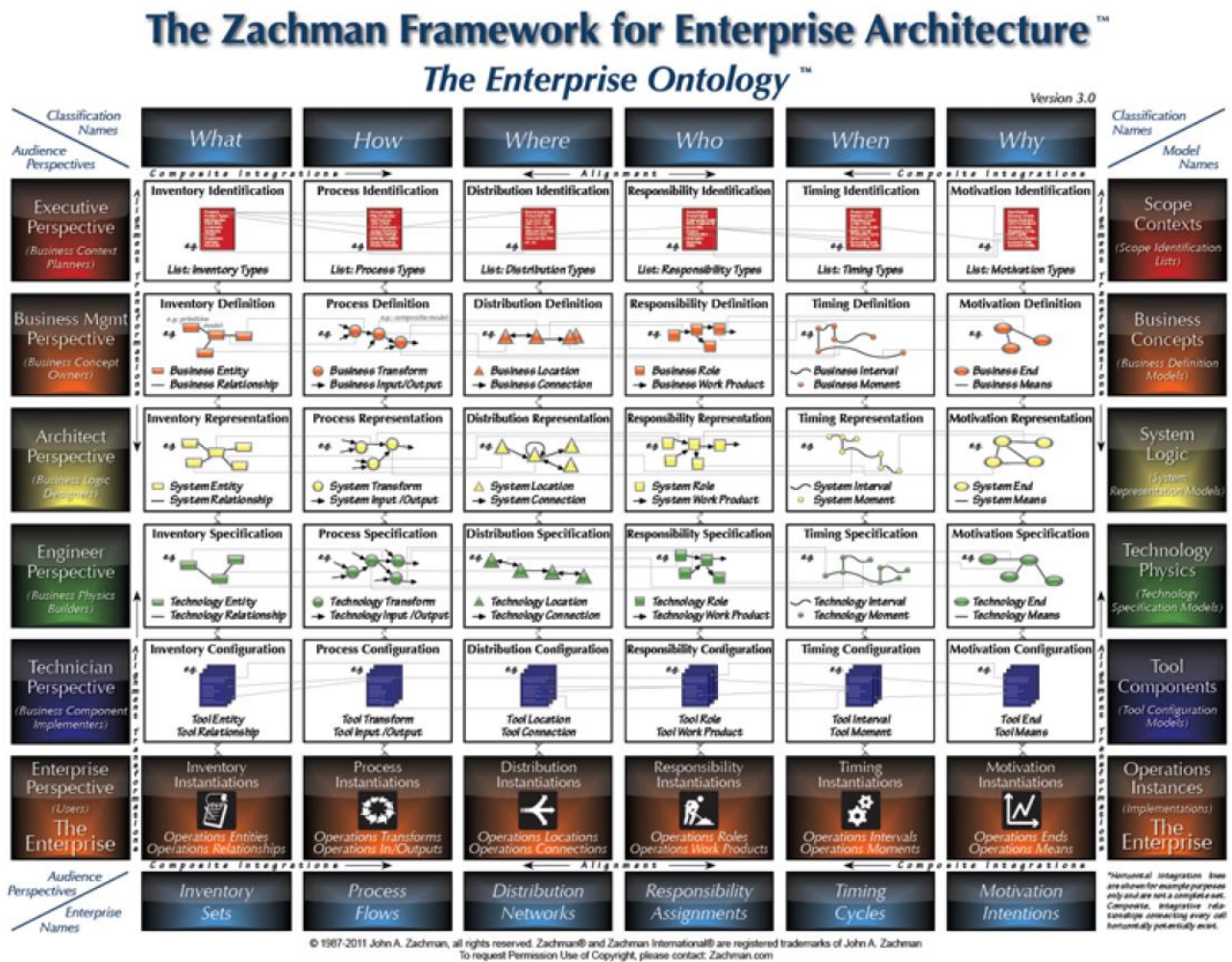
Näkökulmajäsennys (toiminta, tieto, tietojärjestelmä, teknologia) pyrkii tukemaan tarpeiden ja ratkaisujen tarkastelua näkökulma kerrallaan siten, että kunkin näkökulman erityiskysymyksiä voidaan tarkastella kokonaisuutena. Eri näkökulmia yhdistämällä voidaan tarkastelun kohteesta saada kattavampi ja kokonaisempi kuva, jossa voidaan myös porautua yksityiskohtaisemmalle tasolle. Jäsenysmallin neljän näkökulman määrittelyt ovat seuraavat mukailten mm. lähteitä (Open Group 2009, VM 2011):

- **toiminta:** kuvattavan kohteen toiminnalliset ja liiketoiminnan tavoitteet, päätoiminnot, sidosryhmät, organisaatio ja ihmiset sekä niiden tarjoamat palvelut tai tuotteet, prosessit sekä kohteessa suoritettava toiminnot ja tehtävät; voi sisältää myös toimintastrategian ja toimintaohjeet.
- **tieto:** kuvattavan kohteen tunnistetut tietokokonaisuudet, loogiset ja fyysiset tiedot ja tietovarannot, tietorakenteet, tietojen väliset suhteet sekä tiedonhallintaresurssit mukaan lukien avaintiedot.
- **tietojärjestelmä:** kuvattavan kohteen tietojenkäsittelyn ja tiedonhallinnan vaatimukset, ratkaisut, ominaisuudet, järjestelmät ja tietojärjestelmäpalvelut, joilla käsitellään tietoja ja tuetaan toimintaa.
- **teknologia:** ohjelmisto- ja laitteistoresurssit joita käytetään toiminnan, tietojen ja tietojärjestelmien toteuttamiseen, mukaan lukien ICT-infrastruktuuri, väliohjelmistot, verkot ja tietoliikenne, suoritusympäristöt ja tekniset standardit.

SOLEA-projektin aikana näkökulmajäsennyksen riskiksi on havainnoitu se, että kokonaisarkkitehtuurissa tulee huomioida eri näkökulmien tarpeet ja ratkaisut kussakin kehittämisen kohteessa. Eristäytyminen muista näkökulmista esimerkiksi tietoarkkitehtuurissa voi johtaa "näkökulmasiiloutumiseen", jolloin kehittämisen kokonaistavoitteet voivat kärsiä yksittäisen näkökulman ylikorostumisesta. Näkökulmajäsennystä tuleekin käyttää esimerkiksi työn organisoimisen välineenä harkiten. SOLEA-hankkeen eri työkohteissa on myös havaittu, että tuotettavissa kuvauksissa ja määrittelyissä erittäin käyttökelpoisina on nähty eri näkökulmat yhdistäviä kuvauksia. Eri näkökulmiin on tämän tyyppisten kuvausten tuottamisen tukemiseksi järkevää määritellä peruselementit, (esim. catalogs) joita erityyppisissä kuvauksissa hyödynnetään yhdenmukaisesti (ks. luku "Mallit ja Arkkitehtuurikuvaukset").

## 4.2 Zachman jäsenitys

Zachmanin kehikko julkaistiin alunperin 1987, nimellä Zachman Framework for Information Systems Architecture. Hän oli tarkastellut tietojärjestelmien kehittämistä ja vertasi sitä työtä vastaaviin muiden laajojen kohteiden kuten rakennusten tai lentokoneiden kehittämiseen. Hän totesi, että hankkeissa on aina eri osapuolia, rooleja, joita varten tuotetaan erilaisia kuvauksia. Zachmanin kehikko jäsenellään matriisina, jossa riveinä ovat nämä roolit ja sarakkeina näiden tarvitsemat kuvaukset oheisen kuvan mukaisesti. (<http://www.zachman.com/>) Myöhemmin Zachman laajensi tarkasteltavaa kohdetta yrityksen tietojärjestelmistä itse yritykseen ja muokkasi kehikon nimen muotoon Enterprise Architecture.



Kuva 4: Zachman Framework V3.0

Kuvan esittämä versio on julkaistu elokuussa 2011. Tärkeimpiä muutoksia V2.0 verrattuna on kuvattu oheissa Architect Corner -blogissa: <http://blogs.icmgworld.com/2012/02/02/zachman-framework-graphics-v3-0-whats-new/>

### 4.3 Capgemini IAF (Integrated Architecture Framework)

Capgemini on kehittänyt omaa kokonaisarkkitehtuurimenetelmäänsä 1990-luvun loppupuolelta lähtien (McAulay, 2004). Sen voidaan sanoa olevan yksinkertaistettu versio Zachmanin jäsentelystä. Kehikosta on hyvä kuvaus osoitteessa: <http://iea.wikidot.com/iaf>.

IAF kehikon sarakkeet käsittävät neljä osa-alueetta Zachmanin kuuden sarakkeen sijaan. Osa-alueet ovat kuvan mukaisesti seuraavat:

- Toiminta-arkkitehtuuri (joka sisältää prosessit, osapuolet ja organisaation) - yhdistää Zachmanin How ja Who sarakkeet
- Tietoarkkitehtuuri - käsittää What sarakkeen

## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

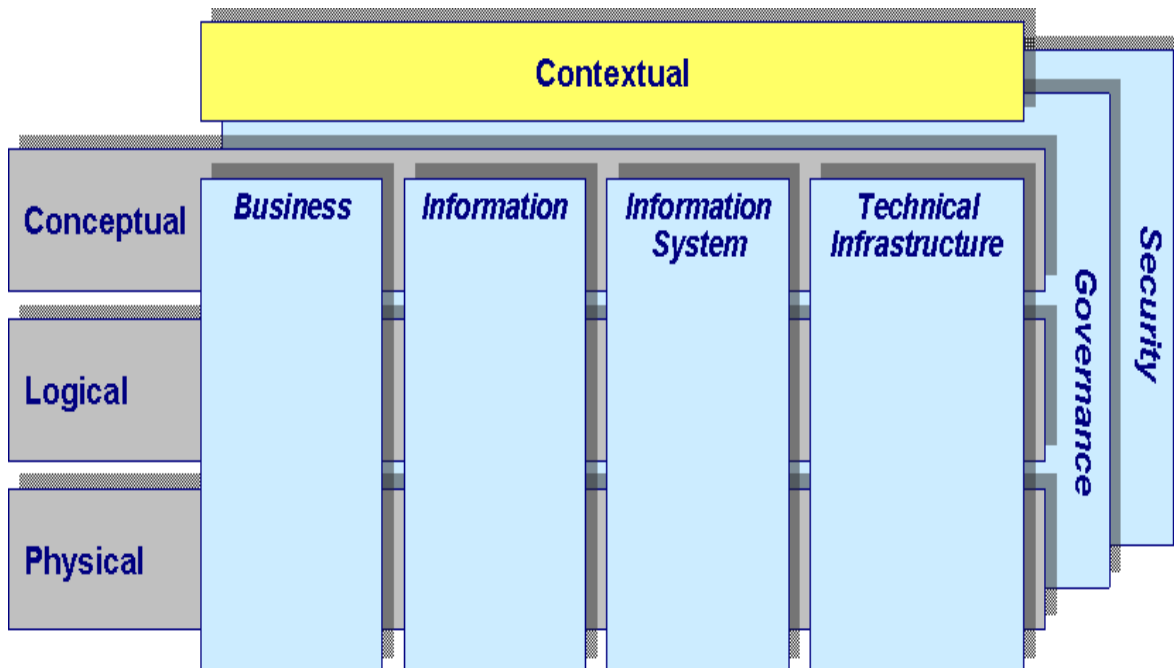
- Tietojärjestelmäarkkitehtuuri - käsittää When ja How sarakkeet
- Teknologisen infrastruktuurin arkkitehtuuri - käsittää Where sarakkeen

IAF kehikon riveille on annettu nimitykset Conceptual-Logical-Physical. Myöhemmin myös Zachman on ottanut käyttöönsä samat nimitykset vastaamaan Owner - Designer - Implementer -rivejä. Yhteistä on, että abstraktiotaso täsmentyy rivejä alaspäin mentäessä.

Mielenkiintoista on, että IAF kiinnittää myös riveille kysymykset, joihin rivillä olevien kuvausten ajatellaan vastaavan. IAF kehikon rivit ovat seuraavat:

- Contextual - Why - alla oleville arkkitehtuureille yhteiset tekijät kuten strategiat, periaatteet, rajaukset, reunaehdot jne. Kerroksen roolit ovat johto ja strategisen tason arkkitehdit.
- Conceptual - What - Näkemys palveluista (Liiketoiminnalliset palvelut, Informaatiopalvelut, Sovelluspalvelut ja Infrastruktuuripalvelut)
- Logical - How - Palveluiden tuottamiseksi tarvittavat komponentit ja näiden väliset yhteydet (Organisaatorakenteet, Informaatiokomponentit, Sovelluskomponentit ja Infrastruktuurikomponentit)
- Physical - With What - Edellä olevien komponenttien implementointi, sijoitus ja hallinta, konkretisointi

Näiden osa-alueiden rinnalla on Governance- ja Security-osa-alueet. Nekin voidaan jäsentää vastaavien abstraktiotasojen mukaisille riveille.



Kuva 5: IAF, Integrated Architecture Framework

#### 4.4 JHS 179 näkökulmat

JHS 179 arkkitehtuurikehys ottaa huomioon eri näkökulmat toiminnallisia tai teknisiä ratkaisuja kuvattaessa ja kehitettäessä. Tarkoituksena on, että kehittämisesä ymmärretään minkälaiseen toiminnalliseen tarpeeseen ja ympäristöön uutta toimintamallia tai ratkaisua kehitetään.

JHS 179 menetelmä sisältämät neljä näkökulmaa:

- Toiminta-arkkitehtuurin näkökulma.
- Tietoarkkitehtuurin näkökulma.
- Tietojärjestelmäarkkitehtuurin näkökulma.
- Teknologia-arkkitehtuurin näkökulma.

Kokonaisarkkitehtuuria voidaan kuvata kussakin päänäkökulmassa myös horisontaalisista näkökulmista, esim. palveluarkkitehtuurin-, integraatioarkkitehtuurin tai tietoturvallisuuden näkökulmista. Tärkeysnäkökulmaksi kokonaisarkkitehtuuria kehitettäessä on tietoturvallisuuden näkökulma huomioitava koko ajan. Tietoturvallisuus edellyttää jo itsessään kokonaisvaltaista organisaation rakenteiden hallintaa ja huomioon ottamista, joten erilaisten ratkaisujen tietoturvallisuus tulee ottaa huomioon kautta linjan kaikissa eri kuvauksissa (JHS 179).

Kehyksessä on kolme käsitetasoa: käsitteellinen, looginen ja fyysinen taso. Lisäksi kehyksessä on kuvattu periaatteiden taso, jolla määritetyt periaatteet ja tehdyt linjaukset ohjaavat kaikkien käsitetasojen kuvauksia (JHS 179).

JHS 179 arkkitehtuurikehyksessä kokonaisarkkitehtuuri on jaettu erilaisiin abstraktio- eli käsitetasoihin. Näiden tarkoituksena on mahdollistaa joko ns. ylhäältä-alas (top-down)- tai alhaalta-ylös (bottom-up) -suunnittelu. Tasot noudattavat IAF:n (ks. luku 4.3) tasojaottelua.

JHS 179 kokonaisarkkitehtuurin kuvaustasot:

- Käsitteellinen taso (mitä).
- Looginen taso (miten).
- Fyysinen taso (millä).

Ylimmällä käsitetasolla arkkitehtuurin osakuvaukset ovat periaatteellisia ja melko abstrakteja, tarkempia tasojia yhdistäviä kuvauksia. Alimman käsitetasojen kuvaukset taas käsittelevät hyvin konkreettisia, käsin kosketeltavia elementtejä, kuten fyysisiä palvelimia, tietoliikenneverkkoja, järjestelmiä ja sovellustuotteita sekä laitteiden sijoittumista eri laitetiloihin (JHS 179).



Kuva 6. JHS 179 Arkkitehtuurikehys (JHS 179).

Lähde: <http://www.jhs-suositukset.fi/web/guest/jhs/recommendations/179>

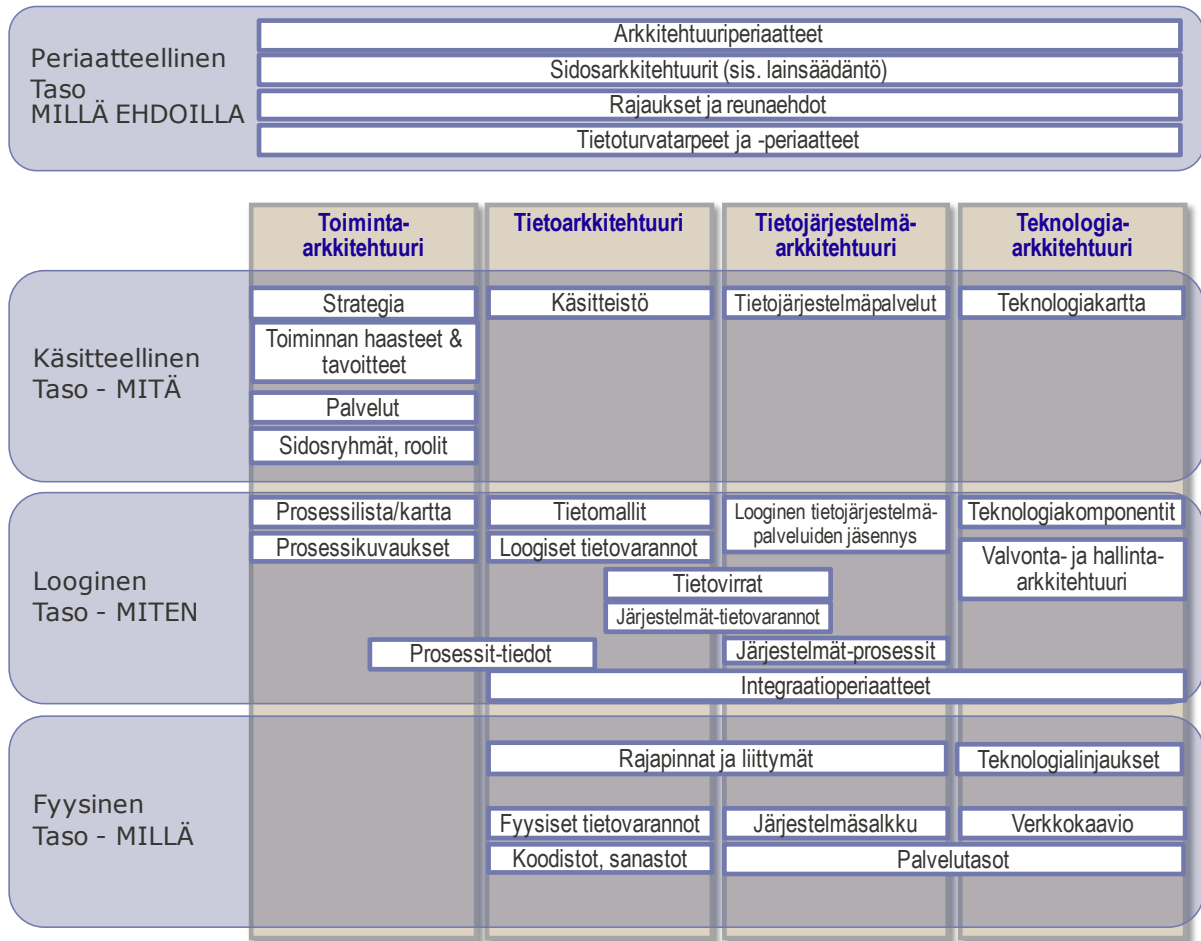
#### 4.5 Kartturi 2.0 näkökulmat

Kartturi 2.0 kokonaisarkkitehtuurikehys jäsentyy eri näkökulmiin ja käsitteellisiin tasoihin (abstraktiotasoihin) kuvassa 2 esitetyllä tavalla. Lisäksi kuvassa on mukana valkoisilla laatikoilla kuvatut yksittäiset kuvauksen kohteet, joihin ei tässä luvussa enempää syvennyttä. (Kartturi, 2012)

Kartturi 2.0 kokonaisarkkitehtuurikehysten näkökulmat ja käsitteelliset tasot muodostavat matriisin, johon yksittäiset osakuvaukset voidaan sijoittaa. On hyvä muistaa, että näkökulmat ja käsitetasot ovat suuntaa-antavia ja niiden tehtävänä on jäsentää erilaisten ratkaisujen ja ympäristöjen kuvaamista kokonaisarkkitehtuurin mukaisesti. Nämä rajat eivät ole aivan tarkkoja ja osa kuvauksista osuukin näkökulmien tai käsitetasojen rajoille.

Edistyneemmät kokonaisarkkitehtuurikehysten hyödyntäjät voivat myös tulkita yksittäisten osakuvauksen sisältöjä omilla tavoillaan. Ei ole väärin tai haitallista sijoittaa tiettyjen näkökulmien asioita toiseen näkökulmaan silloin, kun tämä tehdään hallitusti, perustellusti ja dokumentoidusti. Olennaista on muistaa, että kyseinen yksityiskohta voi olla osa kokonaisratkaisua ja sen vaatimukset tai toteutustapa tulee mahdollisesti kuvata.

## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet



Kuva 7. Kartturi 2.0 kokonaisarkkitehtuurikehyksen jäsentymisen.

### Kartturi-kehysten näkökulmat

Kokonaisarkkitehtuurikehyks ottaa laajasti huomioon eri näkökulmat palvelu- tai teknisiä ratkaisuja kuvattaessa ja kehitettäessä. Tarkoituksena on huolehtia siitä, että erityisesti kehittämisessä on ymmärretty minkälaiseen toiminnalliseen tarpeeseen ja ympäristöön uutta ratkaisua kehitetään. Arkkitehtuurin kehittämisen lähtökohtana ovat aina toiminnan tarpeet.

Kartturi 2.0 kokonaisarkkitehtuurikehyks sisältää JHS 179 suosituksen tavoin neljä näkökulmaa:

- Toiminnan näkökulma
- Tiedon näkökulma
- Tietojärjestelmän näkökulma
- Teknologian näkökulma

### Kartturi-kehysten käsitteelliset tasot

Kartturi kokonaisarkkitehtuuri on jaettu erilaisiin käsitteellisiin tasoihin. Näiden tarkoituksena on mahdollistaa joko ns. ylhäältä-alas (top-down) tai alhaalta-ylös (bottom-up) kuvaukset.

Kokonaisarkkitehtuuriin on kuvattu seuraavat kuvaustasot:

- Käsitteellinen taso (mitä)
- Looginen taso (miten)
- Fyysinen taso (millä)

Näitä ohjaavat vielä ns. periaatteellisen tason osakuvaukset, joita ei jaeta eri näkökulmiin vaan ne vaikuttavat kaikkien näkökulmien kuvauksiin alemmilla käsitetasoilla.

Ylimmällä tasolla arkkitehtuurin osakuvaukset ovat periaatteellisia ja melko abstrakteja tarkempia tasoja yhdistäviä kuvauksia. Alimman käsitetason kuvaukset taas käsittelevät hyvin konkreettisia, käsin kosketeltavia elementtejä kuten fyysisiä palvelimia, tietoliikenneverkkoja, järjestelmiä ja sovellustuotteita sekä laitteiden sijoittumista eri laitetiloihin.

Käsitetasot muodostavat hierarkian siten, että tavoitetilassa alempien käsitetasojen osakuvaukset noudattavat ylempillä tasoilla kuvattuja arkkitehtuurilinjauksia.

Lähde: <http://raketti.csc.fi/kokoa/kartturi>

### 4.6 RM-ODP näkökulmat

ISO (International Standards Organization) on standardoinut Reference Model for Open Distributed Processing (RM-ODP)-mallin alun perin hajautettujen järjestelmien tarkastelemiseksi eri näkökulmista. Näkökulmien avulla voidaan erottaa toisistaan eri näkökulmia järjestelmien suunnittelua varten, koska kaikkien aspektien sisällyttäminen yhteen kuvaukseen on yleensä mahdotonta (Mykkänen et al. 2004). Malli on hyväksytty standardiksi ISO/IEC 10746 part 1-4.

Näkökulmat ovat *enterprise, information, computational, engineering* sekä *technology*. RM-ODP-malli on toiminut taustana useille uudemmille kokonaisarkkitehtuurin jäsentämismalleille.

Näkökulmat on määritelty seuraavasti (ISO/IEC 1995):

- the *enterprise viewpoint*, which is concerned with the business activities of the specified system; Enterprise specification of an ODP system is a model of the system and the environment with which the system interacts. It covers the *role* of the system in the business, and the human user roles and business policies related to the system.

*Mallin voidaan katsoa kuuluvan enterprise-näkökulmaan, jos se kattaa järjestelmän tai järjestelmien ”yli” ulottuvan alueen, kokonaisia toimintaprosesseja, järjestelmän kontekstin tai runsaasti muiden eri näkökulmien asioita siten, ettei mikään niistä ole ensisijainen. Myös jos mallissa kuvataan pelkästään ihmisen toimintaa tukevia ohjeita tai, joihin ei liity sinällään automaattista tietojenkäsittelyä, tai kokonaisuuteen kohdistuvia vaatimuksia, on määrittelyksen alue sovellustarkastelun ulkopuolella ja kuuluu enterprise-näkökulmaan.*

- the *information viewpoint*, which is concerned with the information that needs to be stored and processed in the system; An information specification of an ODP system is a model of the information that it holds and of the information processing that it carries out. The information model is extracted from the individual components and provides a consistent com-



mon view which can be referenced by the specifications of information sources and sinks, and the information flows between them.

*Malli kuuluu information-näkökulmaan, jos sen keskeisin anti on toimialueen tai ratkaisujen tietosisällön (tietokokonaisuudet, tietoelementit) tai tietosisällön viitemallin kuvaus, sanaston, nimikkeistön, luokituksen tai koodiston määrittely tai käsitemalli.*

- the **computational viewpoint**, which is concerned with the description of the system as a set of *objects* that interact at *interfaces* - enabling system distribution; A computational specification of an ODP system is a model of the system in terms of the individual, logical components which are sources and sinks of information. Using the computational language, computational specifications can express the requirements of the full range of distributed systems, providing the maximum potential for *portability* and interworking and enabling the definition of constraints on distribution, while not specifying the detailed mechanisms involved.

*Computational-näkökulmaa voidaan käyttää siten, että malli kuuluu siihen, jos sen keskeisin anti on vuorovaikutuksen, kutsuttavien operaatioiden tai järjestelmän toiminnallisten osien tai palvelujen ja niiden toimintojen määrittely.*

- the **engineering viewpoint**, which is concerned with the mechanisms supporting system distribution; An engineering specification of an ODP system defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it identifies. It describes mechanisms corresponding to the elements of the programming model, effectively defining an abstract machine which can carry out the computational actions. and the provision of the various transparencies needed to support distribution.

*Mallin voidaan katsoa kuuluvan engineering-näkökulmaan, jos se määrittelee sovellusarkkitehtuuria, järjestelmän hajautusta tai järjestelmän suoritusta tai hajautusta tukevaa infrastruktuuria.*

- the **technology viewpoint**, which is concerned with the detail of the components from which the distributed system is constructed. A technology specification defines how a system is structured in terms of hardware and software components.

*Malli kuuluu technology-näkökulmaan, jos se keskittyy teknologiseen toteutukseen, määrittelee tekniikan, jolla tiedon tai toiminnallisuuden yksityiskohdat toteutetaan sovelluksessa tai sen rajapinnassa.*

Näkökulmat eivät ole järjestelmän kerroksia tai suunnittelumenetelmiä, vaan joukko tiettyihin asioihin kohdistuvia kysymyksiä ja ratkaisuja. Näkökulmat eivät ole toisistaan riippumattomia: mallit kustakin näkökulmasta ovat osittaisia näkymiä koko järjestelmään, ja esim. samat käsitteet voivat esiintyä useissa näkökulmissa.

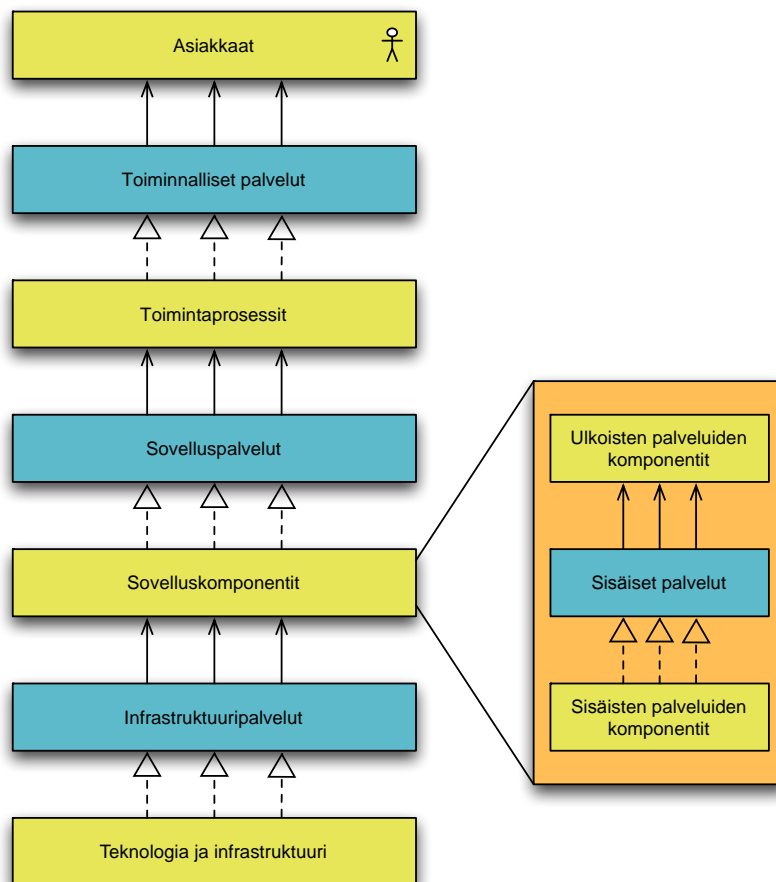
RM-ODP-malliin ei kuulu varsinaisesti mitään tiettyä notaatiota tietyn näkökulman kuvausten tuottamiseen, mutta siinä rohkaistaan formaalien notaatioiden kuten ESTELLE, LOTOS, SDL tai Z käyttöön (Vallecillo, 2001). RM-ODP ei myöskään sisällä varsinaista menetelmää mallien tuottamiseen tai hallintaan, mutta siinä on määritelty ”yhteisiä palveluja” (functions) avointen hajautettujen järjestelmien toteuttamiseen. Yhteiset palvelut on jaettu neljään ryhmään: management, coordination, repository, and security. Nämä eivät kuitenkaan kohdistu arkkitehtuurityöhön vaan hajautettujen järjestelmien suunnitteluun.



## 4.7 Archimate jäsenitys

Archimaten jäsenitys (ArchiMate, 2012) sisältää kolme kerrosta: Alimpana teknologiakerros (Technology), sen päällä tietojärjestelmäkerros (Applications) ja sitten toimintakerros (Business). Huomataan, että kääntämällä SOLEA:ssa käytettävää jäsenitysmallia 90 astetta oikealle saadaan vastaavat ArchiMate kerrokset. ArchiMaten ajatus on liittää nämä kerrokset toisiinsa johdonmukaisella tavalla niin, että ne voitaisiin esittää samassa kaaviossa. ArchiMate käyttää tähän palvelun käsitettä.

Palvelun käsite näkyy ArchiMatessa kahdella eri tavalla. Ensinnäkin palvelut voivat sijaita kunkin kerroksen sisällä, kuten toiminnalliset palvelut prosessien tukena tai tietojärjestelmäpalvelut eri ohjelmistojen käyttäminä. Toiseksi palvelut voivat sijaita eri kerrosten välillä. Alhaalta ylös tarkasteltuna teknologiapalvelut tukevat tietojärjestelmiä, tietojärjestelmäpalvelut puolestaan tukevat toimintaprosesseja, ja toimintaprosessit sitten tuottavat palveluita organisaation asiakkaille. Lankhorstin kirjassa "Enterprise Architecture at Work" on tapana esittää nämä osa-alueet päällekkäisinä kerroksina oheisen kuvan xx mukaisesti.



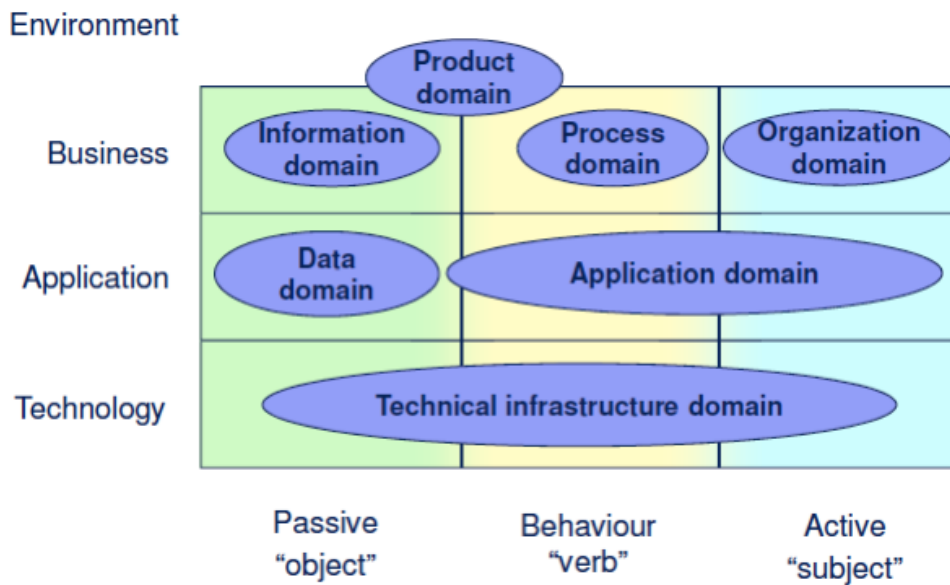
Kuva 8: ArchiMate-palvelukäsite

Oheisessa kerrostetussa mallissa vuorottelevat palvelukerrokset (turkoosit suorakaiteet) ja näiden palveluiden implementointikerrokset (oliivit suorakaiteet). Implementoivan kerroksen sisällä voi myös olla palveluita. Esimerkiksi yhteiskäyttöiset sovelluskomponentit implementoivat sisäisiä so-

velluspalveluita, joiden käyttäjiä ovat toiminnallisia palveluita tuottavat sovelluskomponentit. Arkkitehtuurikuvauksissa voidaan esittää yhtä kerrosta kerrallaan, tai sitten useampia kerroksia ja niiden välisiä palveluita toisiinsa linkattuina.

Mielenkiintoista on havaita, että ArchiMate ei käytä erillistä jäsenystä käsitteellinen-looginen-fyysinen. ArchiMatella voidaan esittää eri tarkkuustason arkkitehtuureja mahdollisesti aina fyysiselle tasolle asti.

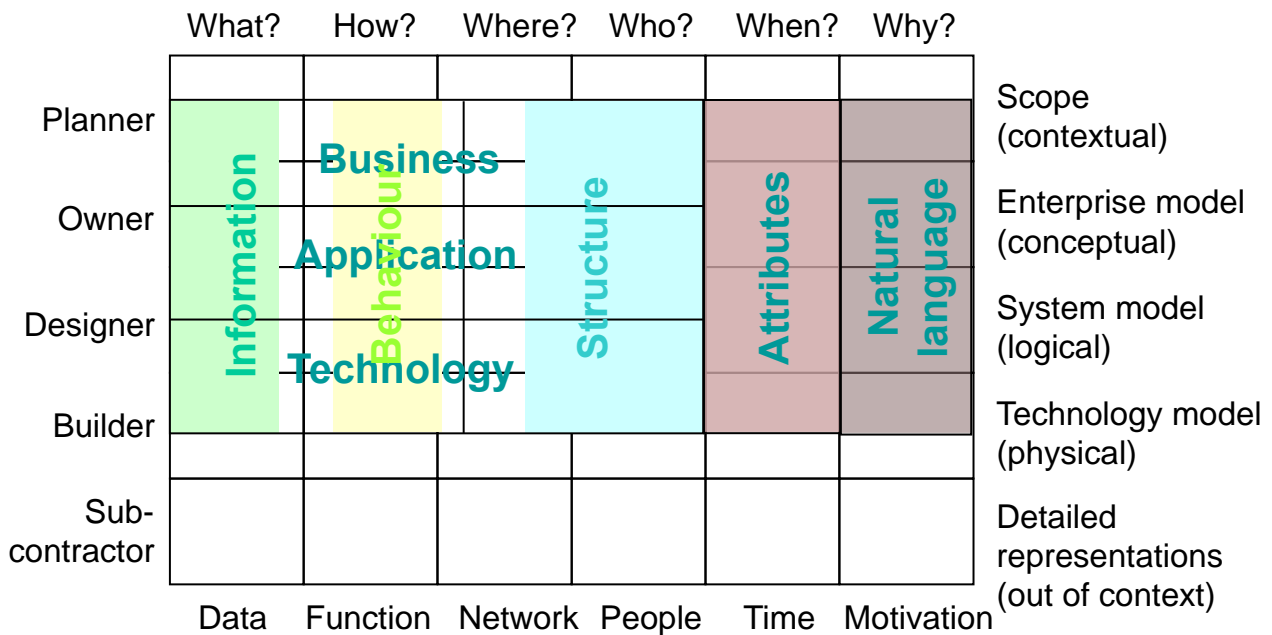
ArchiMate-notaation symbolit on jäsennelty kolmeen ryhmään, joita Lankhorst nimittää aspekteiksi oheisen kaavion mukaisesti:



Kuva 9: Archimate kerrokset, aspektit ja osa-alueet

ArchiMate notaatiossa käytettävät symbolit esitellään tarkemmin luvussa 5, notaatiot.

Lankhorst on eräässä esityksessään pohtinut ArchiMaten sijoittamista Zachmanin kehikkoon oheisen kuvan mukaisesti.



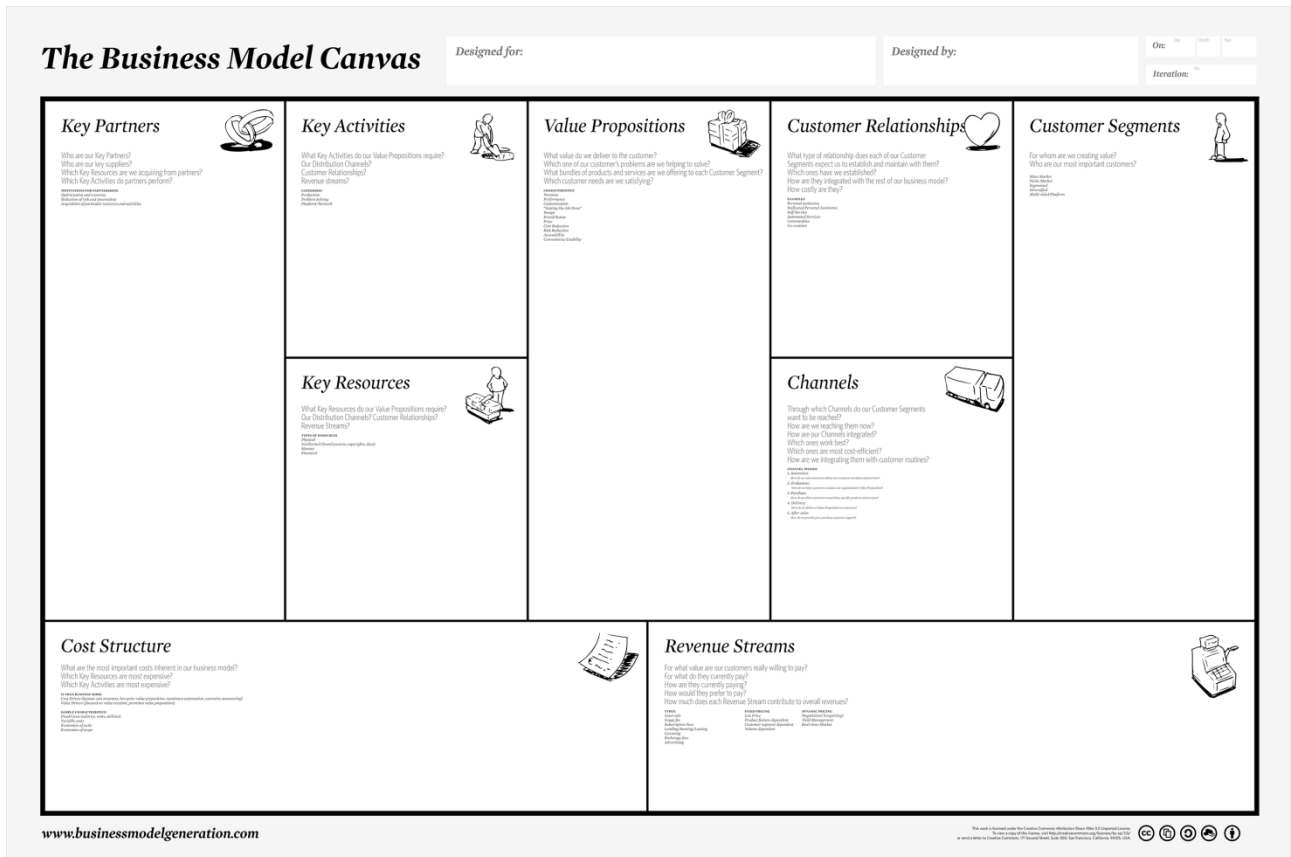
Kuva 10: ArchiMaten sijoittuminen Zachmanin kehikoon

## 5 Yhtä näkökulmaa painottavat jäsentämismallit

Kokonaisarkkitehtuurin "kattavien" jäsenysmallien lisäksi arkkitehtuurityössä on jo kauan käytetty eri näkökulmia erityisesti painottavia jäsentämismalleja. Näissä jäsentämismalleissa voidaan usein tunnistaa painopisteeksi jokin kokonaisarkkitehtuurin päänäkökulmista. Tässä luvussa käsitellään Business Model Canvas -mallia (painopisteinä toiminta-arkkitehtuuri), S3-viitearkkitehtuuria (painopisteinä palvelukeskeinen tietojärjestelmäarkkitehtuuri) sekä HL7 version 3 RIM-viitetietomallia (painopisteinä tietoarkkitehtuuri). Nämä mallit jäsentävät toiminnan ja tietojärjestelmien kehittämistä omasta näkökulmastaan tarkemmin kuin kokonaisarkkitehtuurin yleiset jäsenysmallit, mutta myös jättävät kuvaamatta useita seikkoja joita kokonaisvaltaisessa kokonaisarkkitehtuurijäsenyksessä nähdään usein tarvittavan.

### 5.1 Business Model Canvas

Business Model Canvas on strategisen johtamisen työkalu, joka mahdollistaa nykyisen liiketoimintamallin kuvaamisen ja uusien liiketoimintamallien luonnostellun ja kehittämisen. Kokonaisarkkitehtuuriin suhteutettuna Business Model Canvas on hyvin vahvasti yhteydessä toiminta-arkkitehtuurin näkökulmaan. Business Model Canvas on visuaalinen alusta, jossa on kuvan 11 tapaan valmiiksi alustettuna yhdeksän liiketoimintamallin osa-alueita (Osterwalder, Pigneur, Smith 2010).



Kuva 11. The Business Model Canvas

- Kuinka (How)?
  - Key Activities: Välttämättömät toiminnot liiketoimintamallin toteuttamiseksi.
  - Key Resources: Välttämättömät resurssit, joita tarvitaan lisäarvon tuottamiseen asiakkaalle.
  - Key Partners: Yhteistyökumppanit, jotka täydentävät liiketoimintamallin muita näkökulmia.
- Mitä (What)?
  - Value Proposition: Tuotteet ja palvelut, joita yritys tarjoaa. Arvolupaus kuvaa sitä kuinka yritys erottuu kilpailijoistaan ja on se syy, miksi asiakkaat ostavat kyseiseltä yritykseltä, eikä joltain toiselta (Alexander Osterwalder (2004). The Business Model Ontology - A Proposition In A Design Science Approach).
- Kuka (Who)?
  - Customer Segments: Kohde asiakaskunta yrityksen tuotteille ja palveluille.
  - Channels: Keinot, miten yritys toimittaa tuotteet ja palvelut asiakkailleen. Tämä sisältää myös yrityksen markkinointi ja jakelu strategia.

- Customer Relationship: Linkki, jonka yritys luo itsensä ja eri asiakassegmenttien välille.
- Talous (Finance)
  - Cost Structure: Liiketoimintamallin taloudelliset vaikutukset.
  - Revenue Streams: Yrityksen kassavirta. Yrityksen tulot.

Business Model Canvasin on alun perin kehittänyt Alexander Osterwalder. Business Model Canvas perustuu Osterwalderin huomioihin useiden käsitteellisten liiketoimintamallien samankaltaisuuksista ja hänen aiempaan julkaisuunsa Business Model Ontology, a proposition in design science approach. (Osterwalder, 2004).

### 5.2 S3: A Service-Oriented Reference Architecture

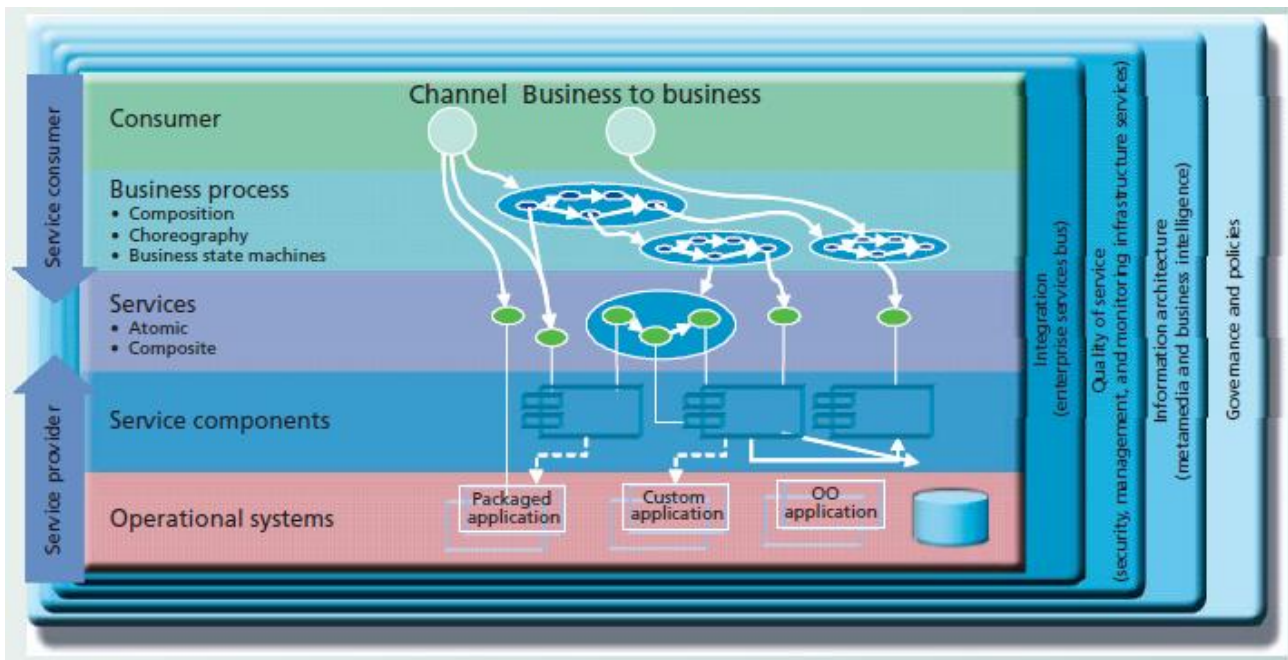
S3 referenssiarkkitehtuuri koostuu yhdeksästä arkkitehtuurisesta kerroksesta (kuva 12), jotka tarjoavat yksityiskohtaiset määrittymiset SOA arkkitehtuurista. Jokaisella yhdeksällä kerroksella on kaksi tasoa looginen ja fyysinen. Looginen taso sisältää kaikki arkkitehtuuriset rakennuspalat, suunnittelupäätökset, vaihtoehdot, tärkeimmät tunnusluvut ja muut loogiset ratkaisut. Fyysinen taso kattaa jokaisen loogisen tason ratkaisujen toteuttamisen käyttäen valittua teknologiaa ja välineitä. S3 kerrokset ovat seuraavat (Arsanjani, Zhang, Ellis, Allam, Channabasavaiah 2007):

- Consumer
- Business Process
- Services
- Service Components
- Operational Systems
- Integration
- Quality of Service (QoS)
- Information Architecture
- Governance and Policies

S3 ei ainoastaan luettele SOA ratkaisun peruselementtejä, S3 tarjoaa myös keinot löytää organisaatiolle parhaiten sopiva tapa hallita valittuja ratkaisuja. S3:n päällimmäinen tavoite on tehostaa mallintamisen prosessia ja arkkitehtuurisen suunnittelun dokumentointia, jotka ovat osa SOA:n luomista. Tarjoamalla mallin ja suuntaviivat arkkitehtuuriseen suunnitteluun, S3 vastaa kysymyksiin: mitä kerroksia täytyy ottaa huomioon, mitä rakennusosia on harkittava ja mitä päätöksiä on tehtävä, kun valitaan arkkitehtuurisia osia kerroksiin.

S3:a voivat hyödyntää useat eri toimijat, kuten kokonais- ja järjestelmäarkkitehdit. Arkkitehteille S3 toimii tarkastuslistana arkkitehtuurisista rakennuspalloista ja niiden suhteista eri kerroksiin, vaihtoehtoista ja päätöksistä, joita tulee tehdä jokaisessa arkkitehtuurisessa kerroksessa. Kerrokset tarjoavat aloituspisteen jaotteluun, kun ollaan perustamassa SOA arkkitehtuuria.

S3 on käytännöllinen vaihtoehto, koska se on toimittajavapaa malli mahdollistaen keskittymisen kohdistettuun SOA ongelma-alueeseen. S3 antaa mahdollisuuden valita eri toimittajien joukosta rakennusosat velvoittamatta käyttämään ainoastaan yhden toimittajan ratkaisuja.



Kuva 12. S3 arkkitehtuurin loogiset kerrokset  
(Arsanjani, Zhang, Ellis, Allam, Channabasavaiah 2007)

### 5.3 RIM HL 7 V 3 reference information model

Erityisesti terveydenhuollossa hyödynnetyn HL7 versio 3 -perheen standardien perustana on RIM (Reference Information Model), joka on HL7-sanomien ja asiakirjojen viitetietomalli (ISO/HL7 2006). RIM-mallin periaatteellisena lähtökohdiana on Austinin ja Searlen kehittämä puheaktiteoria, ja RIM release 1 on hyväksytty ISO-standardiksi 21731:2006 (Health informatics - HL7 version 3 - Reference information model-Release 1).

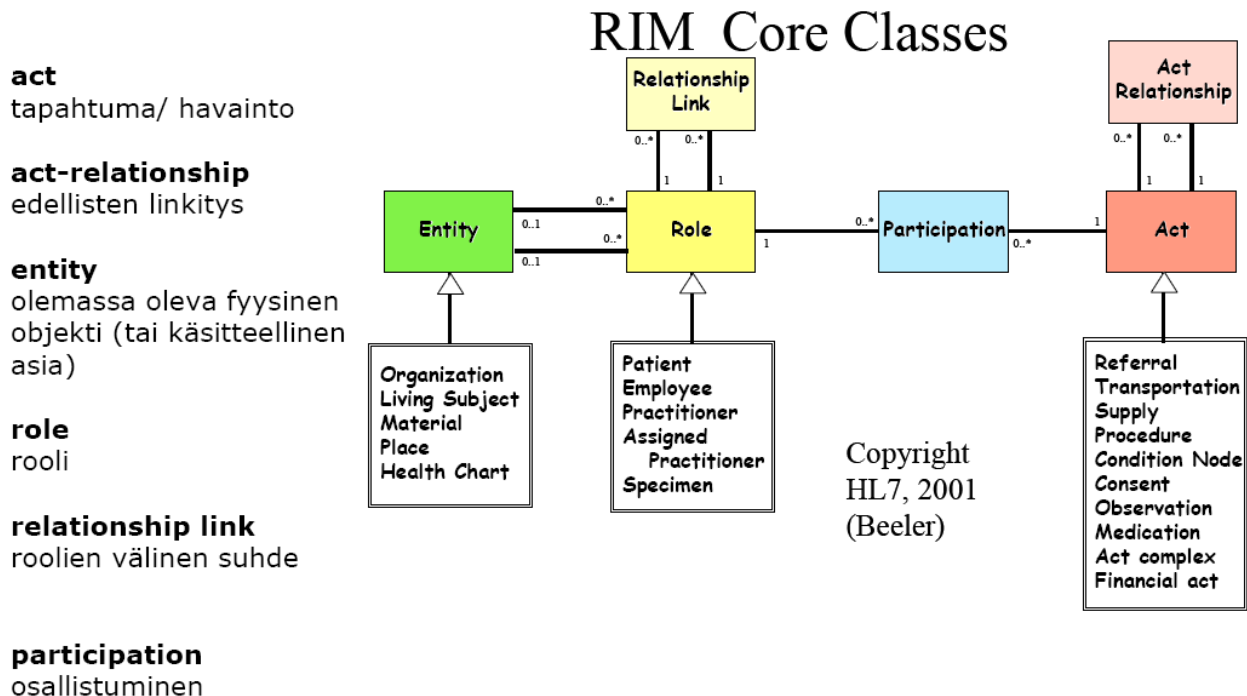
RIM on abstrakti malli, jonka avulla voidaan esittää kaikki terveydenhuollossa tarvittavat tietosisällöt. RIM-mallin tehtävänä on mahdollistaa yhtenäinen tiedon käyttö ja jakaminen useiden eri käyttöalueiden välillä. Malli sisältää UML-kielellä laaditun luokkakaavion sekä UML:n metamalliin ja luokkien värikoodaukseen kohdistuvia laajennuksia.

RIM sisältää kuusi perusluokkaa: Act, Entity, Role, Participation, ActRelationship ja RoleLink. Kolme ensin mainittua ovat ylätasen luokkia, joilla on useita tarkennettuja alaluokkia. Jälkimmäiset kolme kuvaavat ylätasen luokkien ja niiden alaluokkien ilmentymien välisiä suhteita.

Act-luokalla ilmaistaan mm. havaintoja, toimenpiteitä, lääkityksiä ja käyntejä. Act-relationship-luokalla linkitetään act-luokan ilmentymiä toisiinsa. Participation-luokan avulla voidaan kuvata osallistumisia act-luokan kuvaamaan ilmiöön. Role-luokka kuvaa tapahtumissa ilmeneviä rooleja kuten työntekijä, potilas jne. Kuusakin roolissa voi toimia jokin Entity-luokan fyysinen objekti (tai käsitteellinen asia): organisaatio, ihminen, laite, paikka jne. Lisäksi relationship link -luokan avulla voidaan kuvata roolien välisiä suhteita.

RIM-mallin kuudesta pääluokasta Act on keskeisimmässä asemassa, sillä suurin osa informaatiosta ja terveydenhuollon työprosesseista esitetään pääasiallisesti tapahtumina, joiden suorittajana on joku terveydenhuollon organisaation kontekstiin kuuluva toimija. Sen lisäksi Act-luokkaa on vielä

laajennettu kattamaan kaikki asiakirjat ja ”tietokokoelmat”, jotka nähdään siten kuvauksina tarkoituksellisesta toiminnasta – ei fyysisinä entiteetteinä (Vizenor, Smith 2004). Tapahtuma voidaan kuvata joko toteutuneena, toteutumassa olevana, aiottuna, suunniteltuna, pyydettyä tai tilattuna toimintana.



Kuva 13: RIM perusluokat

RIM käyttää abstraktia mallinnustyyliä, jossa useita eri käsitteitä voidaan esittää samalla perusluokalla. Näin on saatu vähennettyä luokkien määrää. Tarkka semanttinen käsite ilmaistaan koodilla, tai käsitteelle voidaan muodostaa myös oma luokka, jos siihen liittyy omia ominaisuuksia tai suhteita. Keskeisimpiä RIM-määrittelyyn kuuluvia rakennekoodeja ovat:

- classCode (Act, Entity, Role), joka ilmoittaa täsmällisen käsitteen tai luokan
- typeCode (Participation, Act\_Relationship, Role\_link), joka ilmaisee viittauksien luonteen
- moodCode (Act), joka ilmaisee onko tapahtuma (act) tapahtunut, suunniteltu, varattu jne.
- determinerCode (Entity), joka ilmaisee onko kuvattu kokonaisuus yleinen entiteettityyppi vai instanssi.

RIM-mallia käytetään HL7 Development Framework -menetelmäkehikon mukaisesti sanomamäärittelyissä siten, että RIM-luokista voidaan luoda mallinnettavan kohdealueen ja sanomien tarkennettuja tietomalleja. Tietomalleissa nojaututaan tyypillisesti lukuisiin sanastoihin ja koodistoihin, jotka sinällään eivät ole osa RIM-tietomallia.

RIM-mallia käytetään HL7 Development Framework -menetelmäkehikon mukaisesti sanomamäärittelyissä siten, että RIM-luokista voidaan luoda mallinnettavan kohdealueen ja sanomien tarkennettuja tietomalleja. Tietomalleissa nojaututaan tyypillisesti lukuisiin sanastoihin ja koodistoihin, jotka sinällään eivät ole osa RIM-tietomallia.

Malli on tarkoitettu terveydenhuollon tietojen mallintamiseen joustavalla tavalla standardien kehittämistä varten, mutta mallin pohjalta on myös toteutettu RIM-pohjaisia tietojärjestelmiä ja tietokantoja. Kaikki HL7 versio 3 -sanomamäärittelyt, HL7 version 3 -sovellusalueiden (domain) tietomallit sekä esimerkiksi CDA R2 -standardi pohjautuvat RIM-mallin käyttöön. Suomessa mm. valtakun-

nallisiin arkistoratkaisuihin liittyvät sosiaali- ja terveydenhuollon HL7 versio 3 -sanomaliikenteen soveltamisoppaat, hyödynnettävät CDA R2 -standardin soveltamisoppaat, HL7 v3 sekä -ajanvarausrajapintojen määrittelyt pohjautuvat RIM-malliin.

Kokonaisarkkitehtuurin kontekstissa ja SOLEA-hankkeen jäsenyksessä RIM-mallin tyyppillinen käyttö sijoittuu tietoarkkitehtuurin loogiselle tasolle: mallin avulla voidaan hyvinkin tarkalla tasolla määrittellä käytettäviä tietokokonaisuuksia, tietoja ja niiden välisiä suhteita sitomatta toteutuksia mihinkään tiettyyn tekniikkaan. Mallin pääluokkien hyödyntäminen on mahdollista myös käsitteellisen tason pääelementtien tunnistamisen apuvälineenä.

RIM-tietomalli toimii myös kokonaisarkkitehtuuri- ja palvelupohjaisen SAIF-yhteentoimivuuskehikon pohjana tietoarkkitehtuurin osalta (ks. SOLEA-hankkeen "Yhteentoimivuus, standardit ja palveluarkkitehtuuri" -tulosedokumentti).

## 6 Kokonaisarkkitehtuurimenetelmät

Menetelmillä tarkoitetaan eri arkkitehtuurikehikoissa ohjeistusta siitä, mitä vaiheita arkkitehtuuri-työhön sisältyy, missä järjestyksessä työt on tehtävä ja mitä kussakin vaiheissa tuotetaan.

### 6.1 The Architecture Development Method (TOGAF ADM)

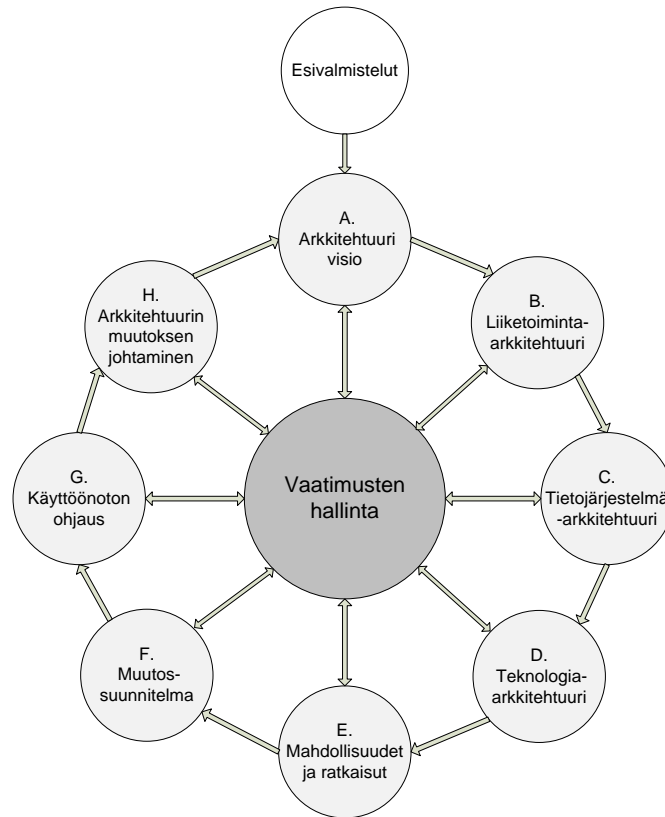
The Architecture Development Method (ADM) muodostaa TOGAF:n ytimen ja on metodi jonka avulla voidaan muodostaa organisaatiokohtainen kokonaisarkkitehtuuri. ADM on syntynyt useiden arkkitehtien parhaista käytännöistä.

ADM tarjoaa testatun ja toistettavan prosessin kokonaisarkkitehtuurin kehittämiseksi. ADM sisältää vaiheet kokonaisarkkitehtuuri-työn aloittamisesta, arkkitehtuuri sisällön kehittämiseen, tavoitetilaa siirtymiseen ja hallittuun arkkitehtuurin toteuttamiseen. Kaikki nämä vaiheet suoritetaan jatkuvassa määrittelyn ja toteuttamisen iteratiivisessa syklissä, joka sallii organisaatiomuutoksen kontrolloidusti siten, että muutos vastaa liiketoimintatavoitteita ja mahdollisuuksia.

Mallin mukaan ensin kuvataan toiminta, tiedot, järjestelmät ja teknologiat nykytilassa ja tavoitetilassa. Kun arvioidaan nykytilan ja tavoitetilaa eroja suoritetaan TOGAF:n mukainen puuteanalyysi (gap analysis). Puuteanalyysi on laajasti käytetty tekniikka ADM:n yhteydessä validoimaan tavoitetilaa arkkitehtuuria. Puuteanalyysin perustarkoitus on korostaa puutteet nykytilan ja tavoitetilaa välillä (The Open Group 2009).

Puuteanalyysin lisäksi ADM:n yhteydessä tulee hyödyntää TOGAF:n sisältökehystä (content framework). Sisältökehys on metamalli organisaation kokonaisarkkitehtuurin kuvaamiseen jokaisessa ADM:n vaiheessa. Sisältökehys tarjoaa ADM:n vaiheisiin rakennuspalikat ja ohjeistuksen, mitä kuvattavia osa-alueita vaiheeseen sisältyy. Sisältökehys ei kuitenkaan ota kantaa, kuinka osa-alueet on kuvattava.





Kuva 14. ADM-kokonaisarkkitehtuurin kehityssyklin vaiheet (The Open Group 2009).

TOGAF ADM kuvataan sarjaksi muutosprosesseja sisältäviä vaiheita (Kuva 14). ADM:n vaiheet kuvataan lyhyesti alla (The Open Group 2009):

- **Esivalmistelut** (*Preliminary*) vaihe valmistaa organisaation TOGAF arkkitehtuuri-projekteja varten. Määrittää vaadittavat valmistelut ja alustavat toimenpiteet, jotta uusi kokonaisarkkitehtuuri täyttää asetetut liiketoimintatavoitteet, sisältäen organisaationkohtaisen arkkitehtuurikehyksen, työkalut ja arkkitehtuuriperiaatteiden määritykset.
- **Vaihe A:** Arkkitehtuuri visio (*Architecture Vision*) kuvaa alustavan vaiheen kokonaisarkkitehtuurin kehityssyklissä. Tässä vaiheessa asetetaan projektin scope, rajoitukset ja odotukset. Luodaan arkkitehtuurinen visio, määritetään sidosryhmät, vahvistetaan liiketoiminta konteksti ja luodaan laskelmat arkkitehtuurityölle.
- **Vaihe B:** Toiminta-arkkitehtuuri (*Business Architecture*) kuvaa toiminta-arkkitehtuurin kehittämisen siten, että se tukee arkkitehtuurista visiota.
- **Vaihe C:** Tietojärjestelmäarkkitehtuuri (*Information Systems Architectures*) kuvaa tietojärjestelmäarkkitehtuurin kehittämisen osana arkkitehtuuriprojektia, sisältäen tietoarkkitehtuurin ja sovellusarkkitehtuurin kehittämiset.
- **Vaihe D:** Teknologia arkkitehtuuri (*Technology Architecture*) kuvaa teknologia-arkkitehtuurin kehittämisen osana arkkitehtuuriprojektia.
- **Vaihe E:** Mahdollisuudet ja ratkaisut (*Opportunities and Solutions*) vaiheessa suoritetaan alustava käyttöön-oton suunnittelu ja tunnistetaan tarpeelliset toimenpiteet

(*projektit, ohjelmat tai portfoliot*), jotka toteuttavat aiemmissa vaiheissa määritetyt arkkitehtuurit.

- **Vaihe F:** Muutossuunnitelma (*Migration Planning*) osoittaa joukon yksityiskohtaisia muutosvaiheen arkkitehtuureja tukemaan käyttöönottoa ja muutossuunnitelmaa, jotka tunnistettiin edellisessä vaiheessa. Lisäksi vaiheessa analysoidaan kustannusrakenteen hyödyt ja riskit. Kehitetään yksityiskohtainen käyttöönotto- ja muutossuunnitelma.
- **Vaihe G:** Käyttöönoton ohjaus (*Implementation Governance*) tarjoaa työkalut arkkitehtuurin käyttöönoton valvontaan. Valmistele ja myöntää arkkitehtuuriset sopimukset. Valvoo, että projektien toteutukset täyttävät arkkitehtuuriset vaatimukset.
- **Vaihe H:** Arkkitehtuurin muutoksen johtaminen (*Architecture Change Management*) tarjoaa jatkuvan seurannan ja muutoksen hallinta prosessin varmistamaan, että arkkitehtuuri vastaa organisaation tarpeita ja maksimoi liiketoimintahyödyn.
- **Vaatimusten hallinta** (*Requirements Management*): Varmistaa, että jokaisessa vaiheessa TOGAF projekti perustuu liiketoiminnan tarpeisiin.

## 7 Notaatiot

### 7.1 Kokonaisarkkitehtuurin osa-alueiden kuvaukset

Kokonaisarkkitehtuuriin liittyy olennaisesti erilaisten kohteiden kuvaaminen. Kuvaukset ovat tyyppillisesti visuaalisia kuvauksia, ts. erilaisia kaavioita ja diagrammeja, joiden tarkoituksena on selittää kuvattavaa kohdetta. Kuten tapana on sanoa, yksi hyvä kuva vastaa tuhatta sanaa. Tämä pitää paikkansa erityisen hyvin myös kokonaisarkkitehtuurin esittämisessä.

SOLEA-projektissa jäsennellään kokonaisarkkitehtuuria vallitsevien käytäntöjen mukaisesti 4x3 matriisilla oheisen kuvan mukaisesti. Arkkitehtuurin eri osa-alueiden kuvaamisen on ajan myötä kehittyntä erilaisia kuvausnotaatioita, joita tarkastelemme tässä luvussa.

Luvussa 1 mainitussa teoksessa "ATK-automaattinen tietojenkäsittely" esiteltiin mm. työnkulkukaavioita ja ohjelman lohkokaaavioita.

Työnkulkukaaviot sijoittuvat luontevasti käsitteelliselle tasolla ja ohjelman lohkokaaviot loogiselle tasolle. Työnkulkukaavioissa on tapana esittää toimintaa, tietoja, tietojenkäsittelytehtäviä ja tietovälineitä sekä näiden muodostamaa kokonaisratkaisua. Työnkulkukaavioissa käytettiin näiden elementtien kuvaamiseen yksinkertaisia loogisia symboleja kuten suorakaiteita, neliöitä, vinoneliöitä sekä näköissymboleja kuten listoja, reikäkortteja, magneettinauhoja jne. Nuolet kuvasivat sitten työn kulkua tehtävästä toiseen.

Ohjelmien lohkokaavioiden symbolivalikoima oli vielä suppeampi, käytännössä tehtäviä, loogisia päättelyitä ja haarautumia sekä nuolia ohjelmalogiikan etenemistä kuvaamaan. Periaatteessa ohjelman kulkukaavio voidaan sijoittaa loogiselle tasolle ja kyseessä on toiminta-arkkitehtuuri, ts. missä järjestyksessä ohjelman toiminnot suoritetaan. Harvemmin kuitenkaan näin sanotaan, koska ohjelman kuvauksessa on kysymys yhden hyvin tiukasti rajatun toiminnan kuvauksesta kun taas kokonaisarkkitehtuurissa kuvataan paljon laajempia kokonaisuuksia.

Työnkulkukaaviot ja ohjelmien lohkokaaaviot ovat olleet useiden pidemmälle kehittyneiden kuvausnotaatioiden lähtökohtia. Lohkokaavioista on tyypillisesti kehittynyt prosessikaavioita. Muitakin lähtökohtia notaatioille on. Yksi keskeinen niistä on UML -kieli, jonka kehittyminen ajoittuu oliosuuntautuneen ohjelmoinnin ja suunnittelun kehittymiseen. Myös yksi lähtökohta on erilaiset käsitteiden ja niiden välisten suhteiden kuvaamiseen tarkoitetut notaatiot, kuten Entity-Relationship mallit jne.

Tässä luvussa tarkastelemme kokonaisarkkitehtuurityön kannalta kiinnostavia notaatioita ArchiMate, BPMN, ORM ja lyhyesti SoAML.

### 7.2 ArchiMate

ArchiMate on mallinnusnotaatio, joka on tarkoitettu kokonaisarkkitehtuurin kuvaamiseen. Se on alunperin kehitetty hollantilaisen Telematica -tutkimuslaitoksen (nykyään Novay) ArchiMate-projektissa vuonna 2004. Projektin tuotoksia on esitelty mm. Mark Lankhorstin ja tutkimusryhmän jäsenten kirjassa "Enterprise Architecture at Work". ArchiMate siirtyi vuonna 2009 Open Groupille, joka julkaisi ArchiMate 1.0 standardin ja on viimeistelemässä ArchiMate 2.0 :n julkaisua. Open Groupin toinen standardointikohde on TOGAF kokonaisarkkitehtuurin menetelmä, josta parhailaan on käytössä TOGAF 9 versio. Jatkossa Open Group tiivistää ArchiMaten ja TOGAFin yhteentoimivuutta.

ArchiMate on avoin mallinnusnotaatio, jota useat eri mallinnustyövälineet tukevat. Esimerkiksi Microsoft Visiota tai Macille OmniGrafflea varten voi verkosta ladata tarvittavat kuvakkeet kaavioiden tekemistä varten. ArchiMate-notaatiolla voidaan kuvata toimintaprosesseja, organisaatiorakenteita, tietojen virtausta tietojärjestelmiä ja teknistä infrastruktuuria. ArchiMate soveltuu hyvin palvelulähtöisen arkkitehtuurin kuvaamiseen. Sitä on tarkemmin selvitetty luvussa 4.8.

Linkki Open Groupin ArchiMate sivuille on <http://www3.opengroup.org/subjectareas/enterprise/archimate>

ArchiMate 1.0 version voi ladata osoitteesta <http://www.opengroup.org/archimate/index.htm>.

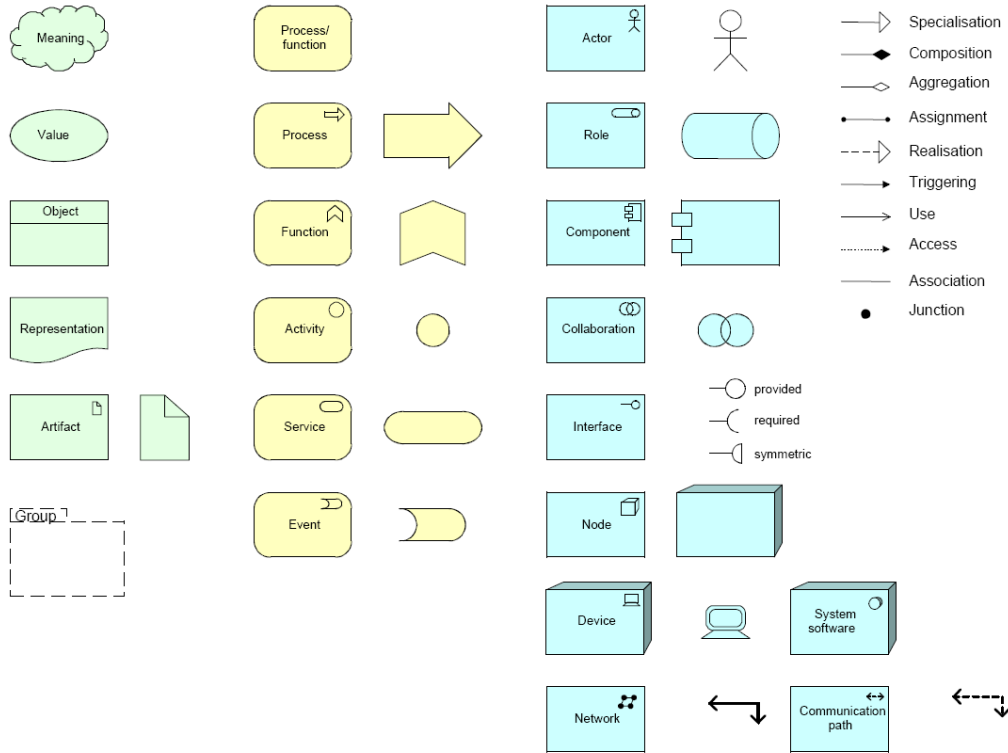
Kuvassa 15 on yhteenveto ArchiMaten symboleista.

ArchiMate on tarkoitettu karkean tason kuvauskieleksi. Sen ei ole tarkoitus korvata tarkempia prosessikuvauskieliä kuten BPMN tai ohjelmistotuotannossa käytettäviä notaatioita kuten UML. Suositus onkin laatia kuvauksia näillä eri notaatioilla toisiaan täydentämään. Erityisesti käsittekaavioiden ja tietomallien laatimiseen suositellaan käytettäväksi perinteisiä E-R kaavioita tai UML luokkakaa-vioita.

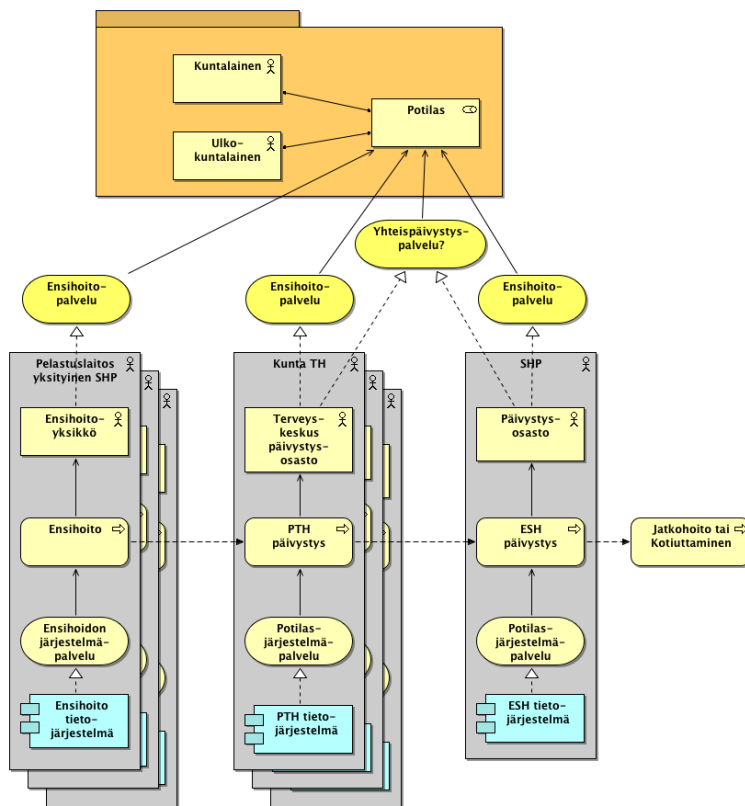
Esimerkki Archimate-notaation käytöstä on oheisessa kuvassa, joka esittää päivystystoiminnan toimijoita ja prosesseja karkealla tasolla. Kuvasta selviää, että toimijoita ovat pelastuslaitosten ja yksityisten yrittäjien ensihoitoyksiköt, kuntien terveyskeskusten sekä sairaanhoitopiirin päivystysosastot, jotka kaikki tuottavat ensihoidon palveluita kuntalaisille ja muille kansalaisille. Ensihoidon palveluiden tuottamiseen kullakin toimijalla on omat prosessinsa. Niitä tukevat kunkin toimijan tietojärjestelmäpalvelut, jotka tuotetaan kyseisillä tietojärjestelmillä.

Monisuutta eli lukumääräsuhteita havainnollistamaan on monistettu kyseisiä organisaatioita päällekkäin. ArchiMate sallii kaavioissa jonkin verran vapaamuotoisuutta havainnollistamaan jotain asiaa.

# Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet



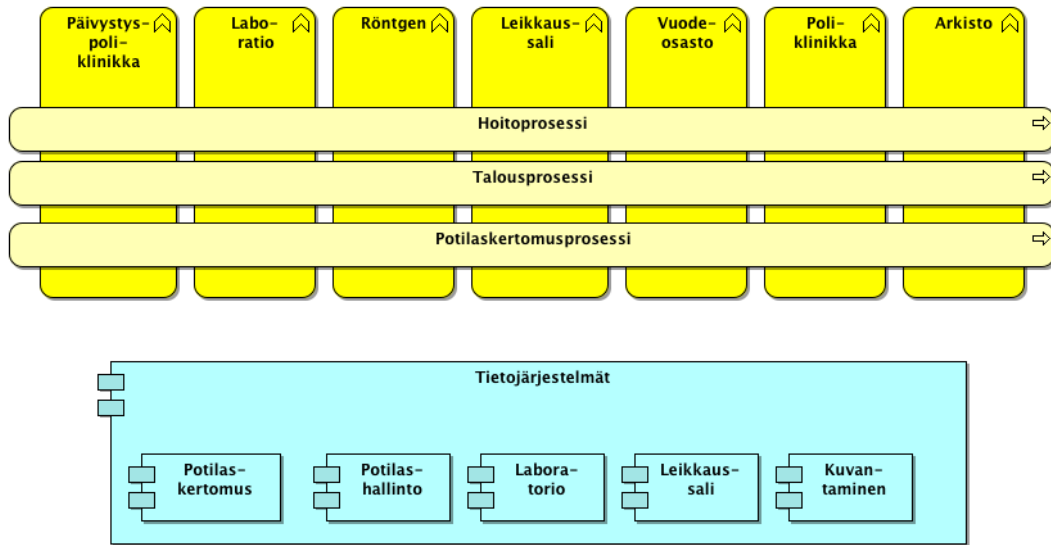
Kuva 15: ArchiMate symbolit



Kuva 16: Päivystyksen toimijat, prosessit ja tietojärjestelmät

## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

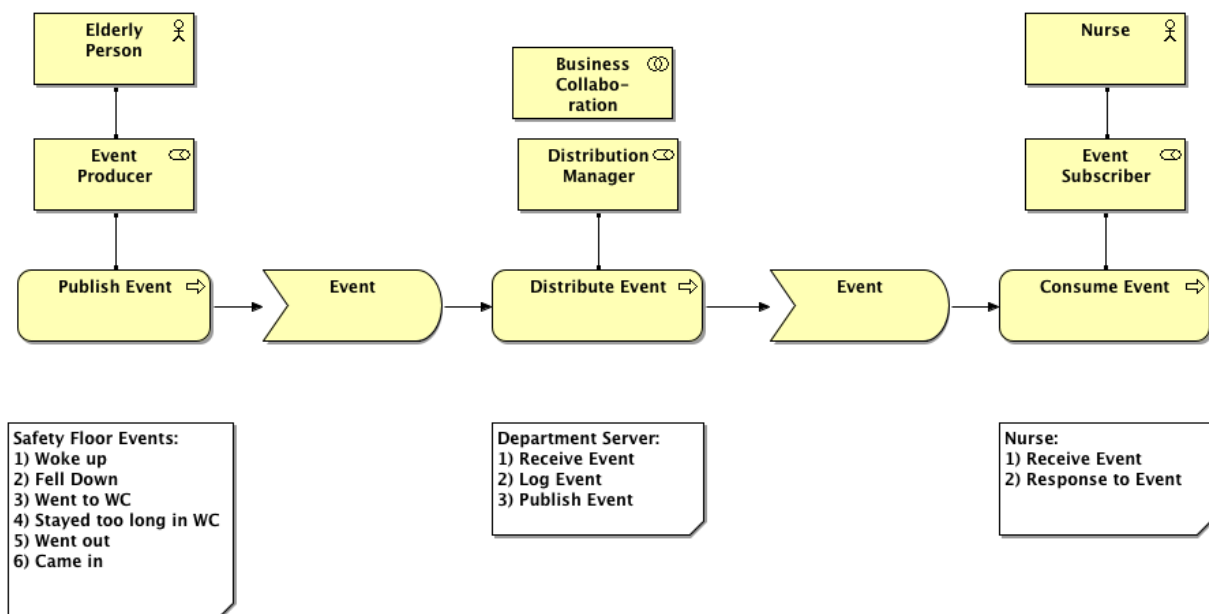
Toinen esimerkki on sitten sairaanhoitopiirin tasolta kartta toiminnoista, toimintaprosesseista ja niitä tukevista tietojärjestelmistä.



Kuva 17: Sairaanhoitopiirin toiminnot, prosessit ja tietojärjestelmät

ArchiMaten käytön aloittaminen on luontevinta aloittaa jonkin pienehkön projektin yhteydessä. Projektista riippuu, aloitetaanko mallintaminen toiminnalliselta tasolta vai tietojärjestelmistä ja teknologiasta. Mallinnettavien kaavioiden kannattaa olla aluksi yksinkertaisia ja keskittyä kuvaamaan vain oleellisia komponentteja toiminnassa tai tietojärjestelmissä. Tämä puolestaan on hyvä asia, koska se auttaakin keskittymään oleellisiin kysymyksiin.

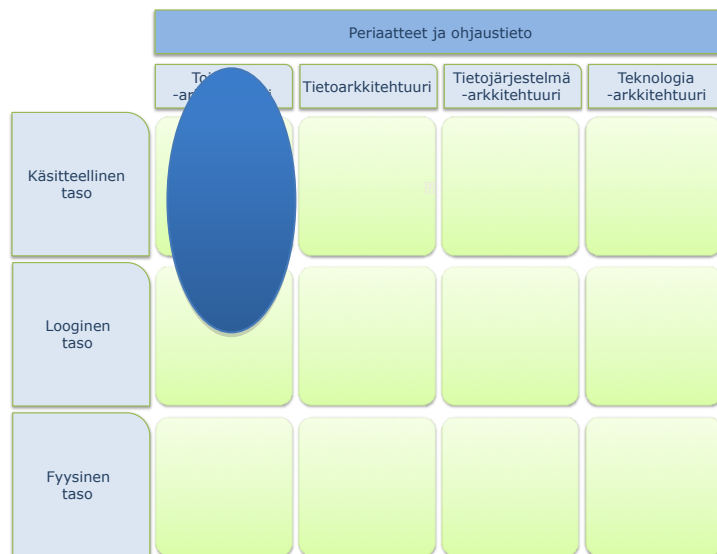
Esimerkki ArchiMaten avulla kuvatusta Event Driven Architecture -tyyppisestä tilanteesta.



Kuva 18: ArchiMaten käyttö EDA kuvauksessa

### 7.3 BPMN

BPMN (Business Process Model and Notation) on graafinen tapa kuvata toimintaprosessia. Se on alkuun kehitetty BPMI (Business Process Management Initiative) organisaatiossa, joka 2005 yhdistyi OMG:n (Object Management Group) kanssa (OMG,2012). Maaliskuussa 2011 OMG julkaisi BPMN standardista version 2.0. Standardin kotisivut ovat osoitteessa <http://www.bpmn.org>. Sieltä selviää mm. että tällä hetkellä 72 toimittajaa on implementoinut BPMN notaation omiin työvälineisiinsä.



Kuva 19: BPMN mallinnusmenetelmä kokonaisarkkitehtuurikehikossa

BPMN notaatio on hyvin samankaltainen vastaavien prosesseja kuvaavien uimaratakaavioiden kanssa. Sen peruselementit ovat aktiviteetit, tapahtumat ja yhdyskäytävät ja näitä yhdistävät, suoritustajärjestystä osoittavat nuolet. Nämä elementit voidaan sijoittaa prosessiin osallistuvia osapuolia esittävien uimaratojen sisään. Uimaratojen välille voidaan piirtää nuolia, jotka esittävät osapuolten keskenään vaihtamia sanomia.

BPMN notaation käyttö voi rajoittua puhtaasti prosessin graafiseen esittämiseen, mutta sen todellinen vahvuus tulee oikeuksiinsa, kun sitä käytetään SOA-hankkeessa mallintamaan prosesseja, palveluita ja niiden suoritusta prosessimoottorilla. BPMN-mallissa prosessi, osaprosessi tai aktiviteetti voidaan toteuttaa Web Service-palveluna, joiden keskinäistä viestien välitystä edustavat malliin kuvatut sanomat.

Monipuolisemmat BPMN mallinnusvälineet voivatkin generoida graafisista prosessikuvauksista BPEL (Business Process Execution Language)-kielellä kuvattuja prosesseja. Niihin liitetään WSDL (Web Service Description Language) kuvaukset palveluista, jonka jälkeen prosessimoottorilla voidaan suorittaa tuotettua koodia.

BPMN on saavuttanut vakiintuneen aseman toimintaprosessien kuvausnotaationa, osin laajan välinetukensa ja osin helpon luettavuutensa ja ymmärrettävyytensä ansiosta. Sitä voi myös käyttää

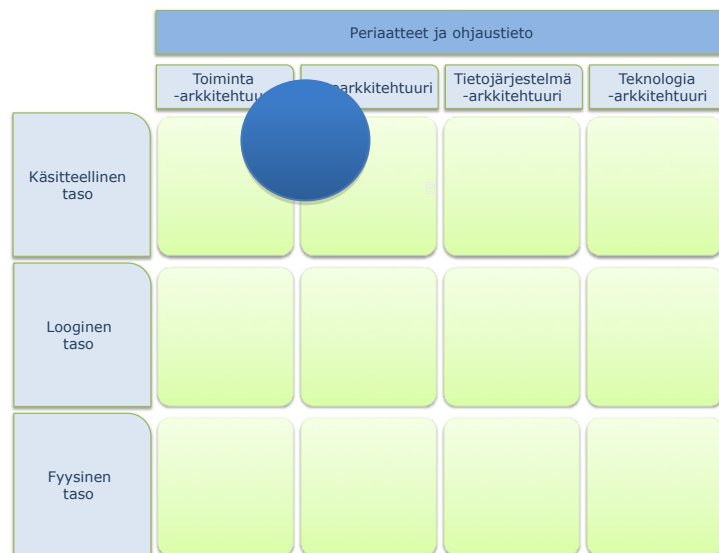
aluksi vain käsitteellisellä tasolla prosessien yleiseen kuvaamiseen. Seuraavaksi voi sitten edetä käyttämään sitä myös loogisen tason hyvin täsmälliseen kuvaukseen vaihtamatta työvälinettä tai notaatiota.

BPMN 2.0 notaation myötä on mahdollista laatia niin täsmällisiä prosessimalleja, että niitä voidaan suorittaa asianmukaisella prosessimoottorilla. Markkinoille on tullut kokonaisvaltaisia paketteja, jotka sisältävät BPMN 2.0 prosessien mallinnustyövälineet sekä prosessimoottorin, jolla voidaan suorittaa mallinnuksen tuloksena syntynyttä prosessia. Yksi esimerkki näistä välineistä on Bonitasoftin Bonita Open Solution <http://www.bonitasoft.com/products>.

### 7.4 ORM

Object-Role Modeling (ORM) on 1970 -luvulta lähtöisin oleva tiedon mallinnusmenetelmä. Sen kehittämisessä on erityisesti kunnostautunut Terry Halpin työskennellessään sekä Microsoftilla että Neumontin yliopistossa. Tunnetuin hänen teoksensa on Information Modeling and relational Databases, Morgan Kaufmann Publishers, 2001 (Halpin, 2001).

ORM kehittäjät sijoittavat mallinnusmenetelmän käsitteelliselle tasolle. SOLEA-projektin jäsentämissmallin mukaiseen kehikkoon se sijoittuu oheisen kuvan mukaisesti.



Kuva 20: ORM mallinnusmenetelmä kokonaisarkkitehtuurikehikossa

ORM:n tarkoituksena on mallintaa toiminnassa käytettävää tietoa käyttäjien ymmärtämällä tavalla. Mallinnuksessa käytetään luonnollisen kielen lauseita tai vastaavia kaavioita. Mallin pohjalta toteutettuun tietokantaan on mahdollista sitten tehdä vastaavalla tavalla luonnollisen kielen avulla muotoiltuja kyselyitä.

Seuraavassa on lyhyt johdanto kaavioissa käytettävään notaatioon. ORM mallin rakennuselementit ovat tieto-objektit (data objects) ja niiden keskinäiset suhteet. Tieto-objektit ovat yksittäisiä tietoelementtejä tai hyvin rajallisia tietojoukkoja. Kaavioissa tieto-objekteja esittävät ellipsit. Tietoobjektien keskinäisiä suhteita esitetään roolien avulla. Rooleja kuvataan suorakaiteiden avulla. Tieto-objekti yhdistetään rooliinsa suoralla viivalla. Tietoobjektien välisiin suhteisiin osallistuu tavallisesti kaksi roolia, mutta yhteen suhteeseen voi liittyä useampiakin rooleja. Yksi tieto-objekti puolestaan voi liittyä useisiin suhteisiin, joissa se on eri rooleissa.

ORM mallinnuksen käytäntöjä ja notaatioita esitellään seuraavassa:

Tieto-objektin tyyppi	Kuvaus	Notaatio
Entiteetti (Entity object, simple object)	Reaalimaailman objekti, joka yksilöidään yksinkertaisella avaimella. Esimerkkinä on henkilö, joka yksilöidään henkilötunnuksella. Notaationa käytetään pyöreäkulmaista suorakaidetta, jonka sisään kirjoitetaan entiteetin nimitys ja sulkuihin yksilöivä avain.	
Arvo (Value object)	Tämä tieto-objekti esittää jotain lukua, nimeä, päivämäärää tms. Se on luonteeltaan jokin skaalarimuuttuja, attribuutti, kuten luku, nimi, päivämäärä jne. Esimerkkinä sukunimi. Notaationa käytetään pyöreäkulmaista, katkoviivoitettua suorakaidetta.	
Tieto, Tosiseikka, Fakta (Elementary Fact) Suhde (Relationship)	Mallinnettavaa kohdealuetta kuvaavia lauseita, jotka sisältävät kohdealueelle olennaisia tietoelementtejä ja niiden välisiä suhteita. Näitä lauseita nimitetään faktoiksi. Periaatteessa yksinkertainen fakta voi esittää, että jollakin objektilla on jokin ominaisuus tai että kahden objektin välillä on jokin suhde. Esimerkki objektin ominaisuudesta on lause <b>"Henkilö, jonka henkilötunnus on 311230-123N, on eläkkeellä"</b> . Esimerkki kahden objektin välisestä suhteesta on lause <b>"Henkilö, jonka henkilötunnus on 3112-123N, omaa sukunimen Virtanen"</b> .	
Rooli (Role), Suhde (Relationship), Predikaatti (Predicate)	Faktan voidaan sanoa sisältävän väittämän roolista (role), jota objekti (object) esittää (predikaatti) faktan ilmaisemassa suhteessa (relationship). Esimerkkejä esitettävistä rooleista ovat edellisten lauseiden "on" ja "omaa". Roolin notaationa käytetään suorakaidetta. Roolin nimitys kirjoitetaan suorakaiteen alle. Suorakaide yhdistetään roolia esittävään objektiin suoralla viivalla.	
Ariteetti, monisuus (Arity)	Faktan monisuus kertoo, kuinka montaa roolia ja niitä esittävää objektia liittyy faktan predikaattiin. Edellä olevat esimerkit edustavat unääristä ja binääristä faktaa. Tyypillisesti faktat ovat binäärisiä, ts. niihin liittyy kaksi roolia. Teoriassa faktojen monisuus voi olla 1,2,3 jne. , vaikka kuinka suuri, mutta käytännössä harvoin esiintyy yli kolmea.	



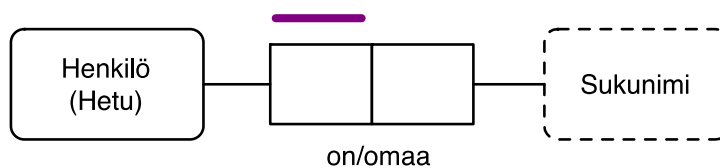
<p>Yhdistelmäobjekti (Composite Object)</p>	<p>Reaalimaailman objekti, jonka yksilöintiin tarvitaan yhdistelmäavain, ts. kahden tai useamman entiteetin avaimen muodostama yhdistelmä. Esimerkiksi tilaukseen voi sisältyä useita tuotteita, jolloin yksittäinen tilausrivi yksilöidään tilausnumeron ja tuotenumeron yhdistelmällä.</p>	
<p>Sisäkkäinen objekti (Nested Object)</p>	<p>Objekti, jonka olemassaolo lähtee kahden muun objektin olemassaolosta ja niiden välisen suhteen muodostamisesta objektiksi. Notationa käytetään entiteetin notaatiota, joka piirretään näiden kahden objektin välisen suhteen ympärille. Esimerkiksi asiakkuus, joka on yrityksen ja henkilön välinen suhde. Asiakkuus voidaan yksilöidä yhdistämällä yrityksen ja henkilön yksilöivät avaimet. Kuitenkin tavallista on muodostaa asiakkuudesta oma objekti, jolle annetaan oma yksilöivä tunnus, esimerkiksi asiakasnumero. Näin menetellään esimerkiksi kauppojen kanta-asiakaskorteissa.</p>	

Malleissa voidaan suhteisiin liittää rajoitteita (constraints). Yleisimmin käytettäviä rajoitteita on yksikäsitteisyys (uniqueness). Esimerkiksi henkilön ja sukunimen väliselle suhteelle voidaan asettaa seuraavat yksikäsitteisyyteen kantaa ottavat ehdot:

Henkilöllä voi olla vain yksi Sukunimi.

Sama Sukunimi voi olla usealla eri Henkilöllä.

Kaavioesityksessä yksikäsitteisyys ilmaistaan kyseisen roolin päälle vedetyllä viivalla.



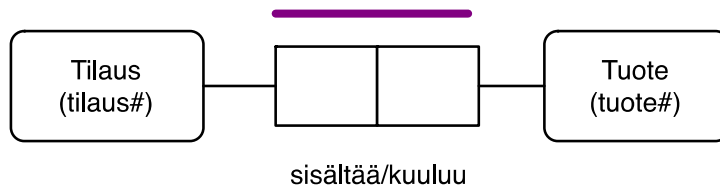
Kuva 21. Henkilö voi esiintyä vain kerran suhteessa sukunimeen.

Yksikäsitteisyys voi koskea kahden eri entiteetin avaimen muodostamaa yhdistelmää.

Tilaus voi sisältää useita tuotteita.

Tuote voi sisältyä useaan tilaukseen.

Tietystä tilauksesta voi tietty tuote esiintyä vain kerran.

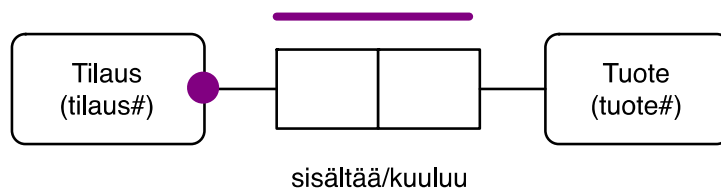


Kuva 22: Tilaukseen sisältyvät tuotteet

Tilausten ja tuotteiden suhdetta voidaan ajatella taulukkona, jossa sarakkeina ovat roolit, ts. tilausnumerot ja tuotenumerot. Tietty tilausnumero voi esiintyä taulukossa useilla riveillä. Samoin tietty tuotenumero voi esiintyä taulukossa useilla riveillä. Kuitenkin tietyn tilausnumeron ja tuotenumeron yhdistelmä voi esiintyä vain kerran. Siksi yksikäsitteisyyttä ilmaiseva viiva ylittää kummankin roolin muodostaman yhdistelmän yli.

Malleissa voidaan myös ilmaista pakollisuutta (mandatory). Pakollisuudella tarkoitetaan sitä, onko suhteeseen liittyvän objektin jokaisen ilmentymän oltava osallisena jossakin suhteessa. Esimerkki:

Jokaiseen tilaukseen täytyy sisältyä vähintään yksi tuote.



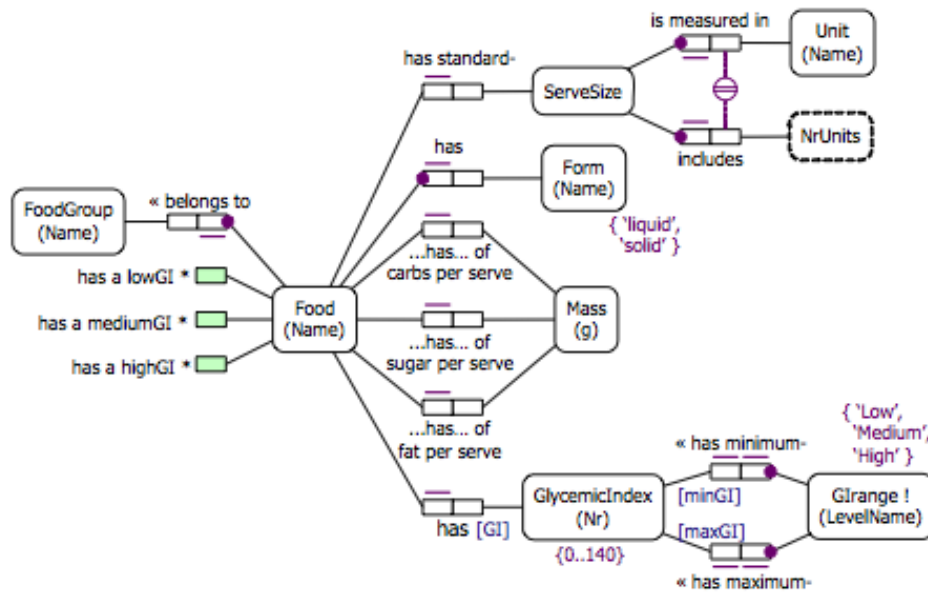
Kuva 23: Pakollisuuden ilmaiseminen

Suhteiden yli ulottuvat rajoitteet tuovat lisää mahdollisuuksia ja piirteitä mallinnukseen. Esimerkiksi on mahdollista ilmaista, että kahden eri suhteen muodostaman yhdistelmän on oltava yksikäsitteinen (uniqueness constraint). Myös on mahdollista ilmaista erilaisia vaatimuksia, jotka koskevat objektin osallistumista eri suhteissa, kuten poissulkeva (Exclusion), yhdessä esiintyvä (Equality) tai osajoukko (Subset). Näiden lisäksi on malleissa mahdollista ilmaista vielä monipuolisia erilaisia rajoitteita, joiden esittely jää kuitenkin tämän raportin ulkopuolelle.

Kuvassa 24 on esimerkki laajemmasta mallista, joka sisältää edellä mainittuja mallinnuksen elementtejä ja käytäntöjä.

ORM malleista on mahdollista tuottaa Entity Relationship malleja noudattamalla mekaanista muunnosohjetta. Näin tietokannan suunnittelija voi ottaa käsitteellisellä tasolla laaditun ORM mallin ja laatia siitä loogisen tason E-R mallin, jonka pohjalta tietokannan toteuttaja voi laatia fyysisen mallin toteutettavaksi käytössä olevalla tiedonhallintajärjestelmällä.

ORM mallinnusta tukevia työvälineitä ovat Microsoft Visio PC-käyttäjille ja Omnigraffle Mac-käyttäjille. Ne ovat piirrosohjelmia, joille on saatavilla ORM mallinnuksessa käytettävät kuvakkeet. Pidemmälle vietyä mallinnusta voi tehdä Microsoft Visual Studio ympäristöön toteutetulla NORMA -työvälineellä. Sen avulla mallintaja voi koota mallinnettavaa kohdetta kuvaavat, luonnollisella kielellä ilmaistut faktat. Näiden perusteella työväline laatii graafisen ORM notaation, joka esittää faktojen avulla ilmaistua kohdetta. Graafista mallia on sitten mahdollista täydentää vielä monipuolisilla rajoitteilla ja muilla ilmaisuilla.



Kuva 24: Esimerkki laajemmasta ORM mallista

ORM mallinnusmenetelmänä antaa mahdollisuudet rikkaaseen ilmaisuun ja monipuoliseen sekä kompaktiin tiedon mallinnukseen. Siinä ehkä piilee myös sen "heikkous", ORM menetelmän laajan hallinnan oppimiskynnys on ensi tutustumalta aika korkea. Verrattuna E-R malliin vaatii ORM menetelmän käyttö sekä mallintajalta että mallien käyttäjiltä kohtalaisen pitkää opiskelua ja asiaan paneutumista ennenkuin mallintaminen alkaa olla sujuvaa.

ORM mallinnusmenetelmä ei liene kovinkaan laajassa käytössä Suomessa. Yksi erityinen syy siihen saattaa olla se, että NORMA-työvälinettä käyttäen suomen kielellä ei ole luontevaa ilmaista faktoja. Englannin kielessä voi helposti laittaa peräkkäin lauseen subjektin ja predikaatin perusmuodoissaan. Suomen kielellä ilmaistussa faktassa myös subjekti usein taipuu johonkin sijamuotoon: Englanninkielen "A person has a name", sanotaan suomen kielellä "Henkilöllä on nimi". Tätä on vaikeata faktoja kerättäessä sellaisenaan syöttää NORMA-välineeseen.

Toinen mahdollinen syy vähäiseen käyttöön voisi olla vähäisessä välinetuessa. Pelkkä piirrosohjelma kuten Visio tai Omnigraffle eivät riitä tukemaan vähänkään laajemman kehitysprojektin tarpeita mallien jakelun, ylläpidon ja yhteiskäyttöisyyden suhteen. Toisaalta läheskään kaikki mallintajat eivät käytä Microsoftin Visual Studiota työvälineenään eivätkä näe tarpeelliseksi sen hankkimista vain ORM -mallinnukseen käytettävän -tosin ilmaisen - NORMA työvälineen alustaksi.

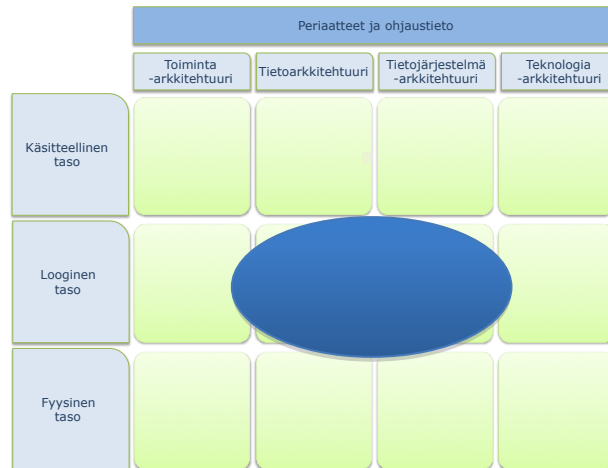
ORM:n vahvuus on siinä, että se vaatii jo käsitteellisellä tasolla täsmällisyyden ja edistää yksilöivien tunnusten (pääavainten) ilmaisemista mallinnustyön varhaisessa vaiheessa. Samoin sillä voi kompaktissa muodossa ilmaista monipuolisia rakenteita ja loogisesti oikeita väittämiä.

Tiedonmallinnuksen ammattilaisille ja vakavamielisille harrastajille ORM on tutustumisen arvoinen menetelmä. Paljon materiaalia löytyy Googlasta menetelmän kehittäjä Terry Halpin tai Object Role model hakusanoina.

## 7.5 SoaML

SoaML (Service oriented architecture Modeling Language) on Object Management Groupin (OMG) toimesta laadittu UML kielen laajennus palveluiden kuvaamiseen palveluorientoitunutta arkkitehtuuria sovellettaessa.

Tällä hetkellä OMG:n sivuilta (<http://www.omg.org/spec/SoaML/>) löytyy versio 1.0 - Beta 2, joka on julkaistu joulukuussa 2009. SoaML -kielelle on tuki useissa UML-pohjaisissa mallinnusvälineissä, esimerkiksi Sparx Systems:n Enterprise Architect ja NoMagic:n MagicDraw.



Kuva 25: SoaML mallinnusmenetelmä kokonaisarkkitehtuurikehikossa

SoaML on UML 2.1 -standardin laajennus, UML- profiili, jossa standardiin on lisätty palveluiden mallintamisessa tarvittavia elementtejä. Laajennusten tarkoituksena on auttaa esimerkiksi palveluiden tunnistamisessa, palveluiden määrittelyssä sekä palveluiden tuottajien ja käyttäjien määrittämisessä. Lisäksi SoaML- kielen on tarkoitus tukea MDA-lähestymistapaa (Model Driven Architecture), jossa mallista tuotetaan ajettava sovellus.

SoaML:n keskeistä sanastoa ovat osapuoli (participant), portti (port), palvelukuvaus (service description) ja ominaisuudet (capabilities).

Osapuoli on joko palvelun käyttäjä tai palvelun tarjoaja. Osapuoli voi olla joko ihminen, organisaatio tai IT-järjestelmän osa. Yksittäinen osapuoli voi tarjota ja käyttää useampaa palvelua yhtäaikaaisesti.

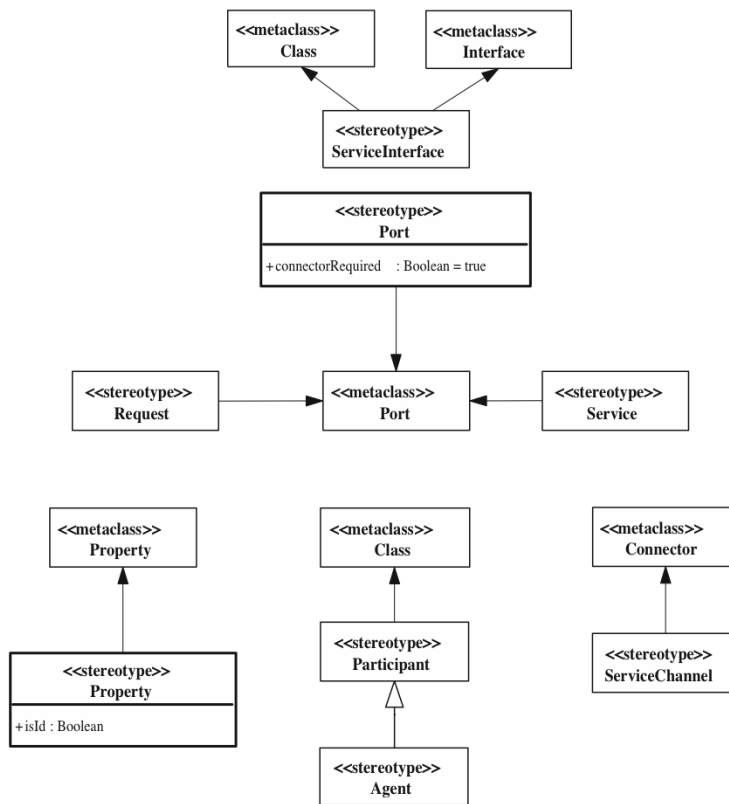
Portti on palvelun vuorovaikutuskohta, jonka kautta osapuolet tarjoavat tai käyttävät palveluita. Portti voi olla joko palveluportti, jota merkitään stereotyypillä '<<Service>>' tai käytävä portti '<<Request>>'.

Palvelukuvaus kuvaa eri osapuolien yhteistoiminnan palvelua käytettäessä tai tarjottaessa. Eri osapuolet voivat tarjota samoja palveluita, mutta kyvyt palveluiden tarjoamiseen voivat olla erilaisia. Esimerkiksi jokin palveluntarjoaja voi delegoida palvelun toteutuksen toiselle osapuolelle.

SoaML kieli tuo palveluiden kuvaamiseen tarvittavat laajennukset UML:n elementteihin. Näitä esitellään kuvassa 26. Uusia termejä ovat palvelurajapinta (ServiceInterface), palvelu (Service), palvelupyyntö (Request), osapuoli (Participant), palvelukanava (ServiceChannel) ja agentti (Agent). Kaavion mukaan luokalle voidaan asettaa stereotyyppi '<<Participant>>' ilmaisemaan että luokka

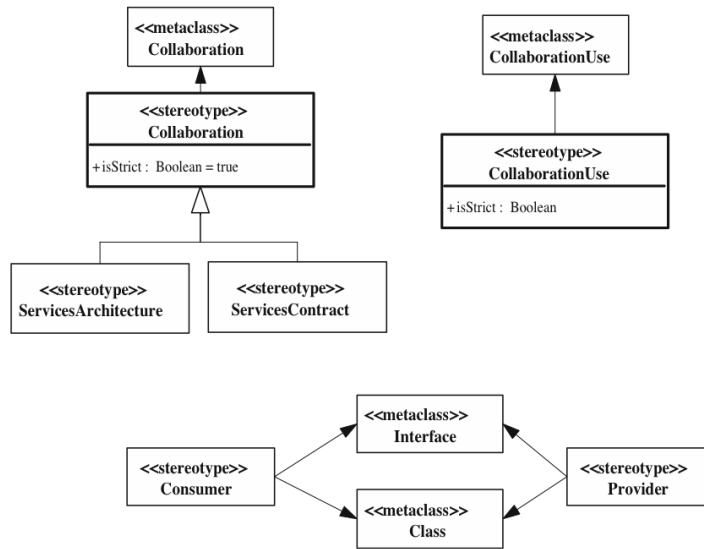
## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

esittää osapuolta. Agenti taas on osapuolen erikoistapaus, joka kuvaa itsenäistä toimijaa, ja joka voidaan merkitä luokalle stereotyypillä '`<<Agent>>`'. Vastaavasti kaaviosta ilmenevät laajennukset UML:n portille pyyntö- ja palveluporttien merkitsemiseksi.



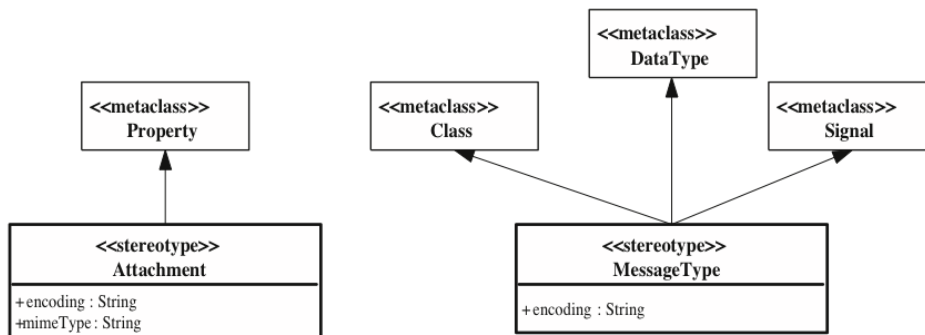
Kuva 26: SoaML Palvelut

Kuvassa 27 näkyvät SoaML:n palvelusopimuksien kuvaamiseen tuomat laajennukset: luokat ja rajapinnat voidaan merkitä stereotyypeillä '`<<Consumer>>`' ja '`<<Provider>>`' kuvaamaan palveluiden käyttäjiä ja tuottajia. UML:n kollaboraatio-elementin laajennuksina ovat stereotyypit '`<<ServiceArchitecture>>`' ja '`<<ServiceContract>>`', joilla voidaan kuvata palveluarkkitehtuuria ja palvelusopimusta.



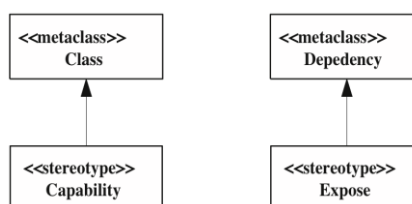
Kuva 27: SoaML palvelusopimukset

Kuvassa 28 esitetään datan kuvaus SoaML-kielellä. UML:n luokat, datatyypit ja signaalit voidaan merkitä viestityypiksi. Liitteellä taas on oma MIME-tyyppinsä sekä koodaustapansa, kuten viestityypilläkin. Käytännössä elementtejä käytetään niin, että viestityypille voidaan merkitä ominaisuuksiksi liitteitä. Näin voidaan mallintaa rakenteista dataa ja sanomapohjaista viestien välitystä.



Kuva 28: SoaML palvelujen data

Kuvassa 29 ovat stereotyypit ominaisuuksien mallintamiseen. Luokka voidaan merkitä ominaisuudeksi ja riippuvuus stereotyypillä '<<Expose>>'. '<<Expose>>' stereotyyppeä käytetään merkitsemään että palvelurajapinta toimii tietyn ominaisuuden mukaisesti. Eli esimerkiksi rajapinnan tarjoava osapuoli käyttää ominaisuusluokan toimintoja rajapinnan toteutuksessa.

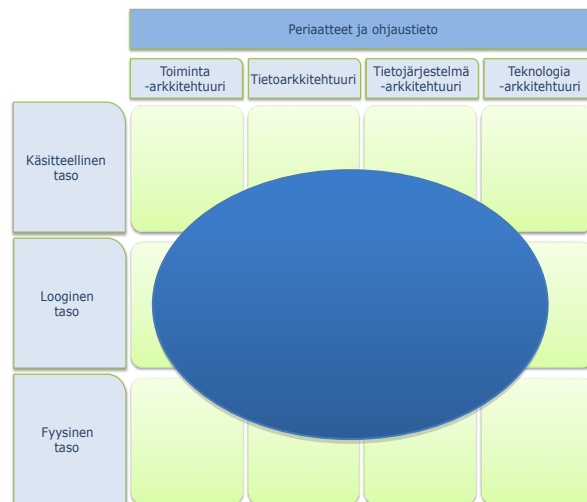


Kuva 29: SoaML ominaisuudet

## 7.6 UML

**UML-mallinnus** (*Unified Modeling Language*) on Object Management Groupin (OMG, UML, 2012) vuonna 1997 standardoima graafinen mallinnuskieli, joka sisältää 13 erilaista kaaviota. Kaavioista kuudella kuvataan rakennetta, kolmella käyttäytymistä ja neljällä vuorovaikutusta. UML on alun perin kehitetty järjestelmä- ja ohjelmistokehitystä varten. Se syntyi yhdistämällä kolme johtavaa oliomallinnustekniikkaa (OMT, Booch, OOSE). Uusin UML-standardi on 2.0 vuodelta 2004. (Wikipedia).

UML on saavuttanut yleismaailmallisen aseman ohjelmistojen kehittäjien keskuudessa. Sitä on laajennettu eri suuntiin kuten prosessien ja palveluiden ja jopa liiketoimintamallien kuvaamiseen. Näillä alueilla se ei kuitenkaan ole saavuttanut samanlaista suosiota.



Kuva 30: UML mallinnuskieli kokonaisarkkitehtuurikehikossa

## 7.7 Yhteenveto notaatioista

Yhteenvetona voi todeta, että notaatioita on tarjolla moneen eri lokeroon kokonaisarkkitehtuurikehikossa. Kaikkien visuaalisten mallien ensisijaisena käyttötarkoituksena on toimia kommunikaation apuvälineinä ja auttaa ymmärtämään mallin esittämää kohdetta. Varsinkin kokonaisarkkitehtuurin käsitteellisellä tasolla on tämä olennaisen tärkeää.

Loogisen tason malleilla on jo toisena käyttötarkoituksena pyrkiä esittämään mallin esittämä kohde niin tarkasti, että mallista voidaan tuottaa toteutusta varten fyysisen tason kohteita, kuten ohjelmaa, sanomakuvauksia tai tietokannan luontilauseita. Käytännössä tämän tarkkuustason notaatioita ei enää käytetä kokonaisarkkitehtuurityössä vaan ohjelmistotuotannon eri vaiheissa.





## **Business Architecture**

### Hakemisto/Listaukset/Luettelot (Catalogs)

- Organization/Actor catalog
- Driver/Goal/Objective catalog
- Role catalog
- Business Service/Function catalog
- Location catalog
- Process/Event/Control/Product catalog
- Contract/Measure catalog

### Taulukot/Matriisit (Matrices)

- Business interaction matrix
- Actor/role matrix

### Kaaviot/Kuvaukset (Diagrams)

#### **Core Diagrams**

- Business Footprint diagram
- Business Service/Information diagram
- Functional Decomposition diagram
- Product Lifecycle diagram

#### **Extension Diagrams**

- Goal/Objective/Service diagram
- Business Use-case diagram
- Organization Decomposition diagram
- Process Flow diagram
- Event diagram

## **Data Architecture**

### Hakemistot/Listaukset/Luettelot (Catalogs)

- Data Entity/Data Component catalog

### Taulukot/Matriisit (Matrices)

- Data Entity/Business Function matrix
- System/Data matrix

### Kaaviot/Kuvaukset (Diagrams)

#### **Core diagrams:**

- Class diagram
- Data Dissemination diagram

#### **Extension diagrams:**

- Data Security diagram
- Class Hierarchy diagram

- Data Migration diagram
- Data Lifecycle diagram

## Application Architecture

### Hakemistot/Listaukset/Luettelot (Catalogs)

- Application Portfolio catalog
- Interface catalog

### Taulukot/Matriisit (Matrices)

- System/Organization matrix
- Role/System matrix
- System/Function matrix
- Application Interaction matrix

### Kaaviot/Kuvaukset (Diagrams)

#### Core diagrams:

- Application Communication diagram
- Application and User Location diagram
- System Use-Case diagram

#### Extension diagrams:

- Enterprise Manageability diagram
- Process/System Realization diagram
- Software Engineering diagram
- Application Migration diagram
- Software Distribution diagram

## Technology Architecture

### Hakemistot/Listaukset/Luettelot (Catalogs)

- Technology Standards catalog
- Technology Portfolio catalog

### Taulukot/Matriisit (Matrices)

- System/Technology matrix

### Kaaviot/Kuvaukset (Diagrams)

#### Core diagrams:

- Environments and Locations diagram
- Platform Decomposition diagram

#### Extension diagrams:

- Processing diagram
- Networked Computing/Hardware diagram
- Communications Engineering diagram

## 8.2 Keskeisimmät KA-kuvaukset

Eri lähteissä on muodostettu ehdotuksia siitä, mitkä kuvaukset ovat erityisen usein tarpeellisia tai erityisen käyttökelpoisia kokonaisarkkitehtuurityössä. Tässä luvussa kuvataan näitä ehdotuksia.

- Kartturi on korkeakoulujen kokonaisarkkitehtuuripilotissa tuotettu kokonaisarkkitehtuurimalli, joka on mukautettu korkeakoulujen tarpeisiin. Kartturi kokonaisarkkitehtuurimalli sisältää kehyksen ja kuvauspohjat, hallintamallin ja kypsyytasmallin arviointipohjineen. Kartturin aineisto löytyy linkistä <http://raketti.csc.fi/kokoa/kartturi> .
- JHS 179 Suosituksessa määritellään menetelmä, jolla organisaation kokonaisarkkitehtuuri suunnitellaan sekä annetaan suositukset kokonaisarkkitehtuurin eri osa-alueiden kuvausten laatimisesta. Suosituksen tarkoituksena on antaa yhtenäinen suunnittelumenetelmä, suunnittelun viitekehys sekä yhtenäiset kuvaustavat ja -mallit julkisen hallinnon organisaatioiden kokonaisarkkitehtuurin kehittämisen eri vaiheissa. Suosituksen mukaan toimitaan sekä organisaation kokonaisarkkitehtuurin kuvaamisen ensimmäisellä kierroksella että kehittämisen seuraavilla iteraatiokierroksilla. Suositus on osa ICT-palvelujen kehittäminen -suositussarjaa ja se täydentää muita sarjan suosituksia antamalla pohjan suosituksissa kuvatulle arkkitehtuuriohjatuille organisaation toiminnan kehittämiseksi. Suositus löytyy linkistä <http://www.jhs-suositukset.fi/suomi/jhs179> .
- Greefhorst (Greefhorst, 2009) on suositellut olennaisia arkkitehtuurikuvauksia, jotka perustuvat kokemuksiin rahoituslaitosten ja julkisen hallinnon organisaatioiden parissa tehtyyn kokonaisarkkitehtuurityöhön

Seuraavassa taulukossa on yhteenvertausta näiden suositusten arkkitehtuurikuvauksista:

Domain of central EA element	JHS [13]	Greefhorst [14]	HY [15]
Principles	X	X	
Value chain		X	
Drivers, goals, objectives	X	X	
Organizations / actors	Xc		
Concept relationships	Xc		
Processes	Xl		
Data entities	Xl		X
Data flows	Xl		
Data repositories	Xl		
Information system svcs	Xc		
Processes + data	Xl		
Systems + data	Xl		X
Functional decomposition / systems map	Xl	X	X
Processes + systems	Xl		
Application portfolio	Xp	X	X
Systems + functions		X	X
Application communication		X	X
Interfaces	Xp		
Technology services	Xc		
Technology standards	X	X	
Technology portfolio			X

Eri projektien analysoinnissa voidaan hyödyntää myös esimerkiksi Archimate-standardista löytyviä peruselementtejä ja tarkastella, onko kokonaisarkkitehtuurikuvauksissa mukana (tai olisiko syytä olla mukana) näiden elementtien kuvauksia ja mitä elementtejä erityisesti olisi syytä yhdistellä toisiinsa (Virkanen 2011).

## 9 Työvälineet

Kuten aikaisemmissa luvuissa on todettu, kokonaisarkkitehtuurityön tuotokset ovat erilaisia kuvia. Näitä kuvauksia voidaan piirtää käsinkin, mutta nykyään valtaosa kuvauksista tehdään erilaisilla työvälineillä. Työvälineitä voidaan ryhmitellä ja luokitella monin eri tavoin.

Hyvää opastusta kokonaisarkkitehtuurityössä tarvittavien välineiden valintaan annetaan valintaoppaassa Enterprise Architecture Tool Selection Guide (Schekkerman, 2011). Oppaassa on myös lueteltu runsaasti välineitä toimittavia yrityksiä ja heidän tuotteitaan.

Oppaassa tarkasteltuja näkökulmia ovat mm:

- Työvälineiltä vaadittava toiminnallisuus
- Työvälineitä käyttävät asiantuntijat
- Työvälineiden soveltuvuus ja tuki käytettäville arkkitehtuurikehikoille
- Yksityiskohtainen vaatimusluettelo

Seuraavassa tarkastellaan lisäksi joitakin työvälineisiin liittyviä näkökulmia.

### 9.1 Kuvausten tyyppi

Edellisessä luvussa kuvauksia tyypiteltiin seuraavasti:

- Hakemisto/Listaukset
- Taulukot/Matriisit
- Kaaviot/graafiset kuvaukset

Kaksi ensimmäistä pohjautuvat tekstin käyttöön kun taas kolmas lähtee visuaalisesta esittämisestä, piirroksista, kaavioista jne. Yksi tapa jaotella työvälineitä perustukin toimisto-ohjelmistoista tuttuun jaotteluun: Tekstinkäsittely, taulukkolaskenta ja visuaalisten esitysten piirtäminen. Toimisto-ohjelmistot lienevätkin edelleen eniten käytettyjä työvälineitä kokonaisarkkitehtuurin kuvausten laadinnassa.

### 9.2 Yksilötyö vai Ryhmätyö

Yksi tapa jaotella työvälineitä lähtee tavasta jäsentää arkkitehtuurityön tekemistä. Tehdäänkö työtä yksilölähtöisesti vai ryhmässä.

- Yksinkertainen tapa on, että jokainen yksin tekee omia kuvauksiaan, tallettaa ne omalle koneelleen ja mahdollisesti jakaa niitä sähköpostilla tai yhteisten levyhakemistojen kautta. Hallinta ja työnkulut ovat yksinkertaisia ja vastuut selkeitä, kukin vastaa omista töistään.
- Toinen tapa lähtee edelleen yksittäisistä kuvausdokumenteista, mutta ne ovat arkkitehtuuria laativan ja hyödyntävän työryhmän yhteisessä käytössä. Tähän voi käyttää jotain ryhmätyöohjelmistoa, jolla voidaan huolehtia dokumenttien metatiedoista, versionhallinnasta, jakelusta, arkistoinnista jne. Ryhmätyöohjelmistot tyypillisesti tukevat myös erilaisia työnkulkuja. Arkkitehtuurityötä laajemmin tehtäessä tarvitaan arkkitehtuurin hallintamallia ja tähän voidaan ajatella käyttää ryhmätyöohjelmien ominaisuuksia. Esimerkkinä työnkulku, jossa sovitaan, kuka laatii arkkitehtuuriluonnoksia, ketkä osallistuvat niiden arviointiin ja kommentointiin, kuka hyväksyy ja keille sitten tuotokset jaetaan tai saatetaan tiedoksi ja käytettäväksi.
- Vielä kehittyneempi tapa on käyttää jotain arkkitehtuurin laadintaan ja ylläpitoon tarkoitettua ohjelmistoa, jonka kuvauskantaan talletetaan kaikki arkkitehtuurin erilaiset elementit ja joista sitten voidaan koota useita erilaisia näkymiä. Merkittävänä etuna tällaisten kuvauskanta- tai repositorypohjaisten työvälineiden käytössä on muutosten hallinnan automatisointi ja synkronointi muiden kuvausten kanssa. Esimerkiksi jonkin kuvauselementin, esim. työtehtävän nimi muutetaan, näkyy tämä muutos heti kaikissa tätä elementtiä käyttävissä kuvauksissa, kun niitä katsotaan kuvauskannasta. Toisaalta näiden laajojen työvälineiden käyttö vaatii vielä vahvempaa ohjeistamista ja työnkulkujen määrämuotoistamista asioiden pitämiseksi järjestyksessä ja hallinnassa. Usein on tapana laajemmissa kokonaisarkkitehtuuriyhteisöissä tehdä työnjakoa erilaisilla perusteilla ja rajata päivitysoikeuksia suppeammalle ryhmälle järjestyksen varmistamiseksi ja vahinkojen ehkäisemiseksi.

### 9.3 Notaatioiden ja eri kuvaustapojen tuki

Yksi tapa luokitella työvälineitä lähtee siitä, kuinka tiiviisti ne tukevat erilaisia mallinnusnotaatioita ja käytettävän mallinnusnotaation sääntöjä.

- Yksinkertaisimmat työvälineet ovat puhtaasti piirrostyövälineitä, kuten MS Powerpoint tai MS Visio. Niitä varten voi olla saatavilla elementtikirjastoja, jotka helpottavat kuvausten laatimista. Esimerkkinä UML-, BPMN-, ArchiMate-, E-R - jne. notaatiot, joiden symbolikirjastoja on saatavilla eri lähteistä internetissä. Näiden avulla voidaan piirtää kyseisiin notaatioihin perustuvia kaavioita vapaamuotoisesti. Piirrosvälineet toisaalta eivät millään tavalla varmista sitä, että syntyvät kaaviot noudattavan kyseisten notaatioiden syntaktisia ja semanttisia sääntöjä. Piirrostyövälineillä aikaansaadut kaaviot usein liitetään sanallisiin dokumentteihin copy-paste periaatteella. Laadukkaiden raporttien aikaansaaminen ja versionhallinta vaatii suhteellisen paljon manuaalista työtä ja työnkulkujen systematisointia.
- Kehittyneemmät työvälineet perustuvat käytettävän mallinnusnotaation sääntöjen tuntemiseen. Jotkin välineet antavat mahdollisuuden jälkikäteen tarkistaa laaditun kuvauksen sääntöjenmukaisuuden ja tarkistuksen tuloksena antavat mahdollisen luettelon virheistä.

Toiset välineet ohjaavat mallintajaa jo kuvauksen laadinnassa ja antavat mahdollisuuden piirtää vain sääntöjen mukaan sallittuja kuvauksia. Tekijän kannalta jälkimmäinen vaikuttaa käyttökelpoisemmalla, parempi saada heti palaute, kuin tehdä turhaa työtä välillä. Usein nämä kehittyneemmät työvälineet antavat myös mahdollisuuden kirjoittaa vapaata tekstiä kuvausten yhteyteen. Kuvauksista voidaan sitten tuottaa valmiita raporttiaihoita, joissa ovat mukana visuaaliset kuvaukset ja niihin liittyvät tekstit. Työvälineiden käyttöönottoon kannattaakin sisällyttää työnkulun pohtiminen valmiisiin loppuraportteihin saakka. Millä kannattaa tehdä kaaviot ja millä sitten kirjoittaa lopulliset raportit kaaviointeen ja sanallisine kuvauksineen.

- Työvälineiden tukemien notaatioiden laajuudessa on myös eroja. Toiset työvälineet on tarkoitettu tukemaan vain yhtä notaatiota, esimerkiksi BPMN-, UML-, ArchiMate- tai E-R -notaatiota. Arkkitehdit joutuvat sitten valitsemaan yhden tai useamman työvälineen sen mukaan, mitä kuvaustyyppäjä tulevat tekemään. Toiset, laajemmat työvälineet tukevat useita erilaisia notaatioita, jolloin arkkitehdin tai arkkitehtiryhmän on tarpeen hankkia ja opetella vain yksi väline.

### 9.4 Koodin generointi

Työvälineitä voidaan jaotella myös sen mukaan, tukevatko ne laaditun kuvauksen toteuttamista vastaavalla yksityiskohtaisemmalla kielellä.

- Esimerkiksi ArchiMate notaatio on puhtaasti kuvaamiseen tarkoitettu notaatio. Sen tarkoituksaan ei ole tuottaa mitään suoritettavaa koodia eikä sen siirtämistä työvälineestä toiseen ole standardoitu.
- BPMN notaatio puolestaan on tarkoitettu tuottamaan kuvatun prosessin suoritusta ohjaavaa koodia. Useimmiten käytettävä koodi on BPEL ( Business Process Execution Language) -koodia, jota suoritetaan ns. prosessimootorilla (BPEL-Engine). BPEL on XML-koodia. Sen ajatus on ohjata Web Services palveluiden kutsuja prosessiin kuvatun suoritustalogiikan mukaisesti.
- E-R notaation perusteella jotkin työvälineet tuottavat tietokannan yksityiskohtaiset luontilauseet ja tukevat siten tietokannan hoitajan työtä. Tätä tukee myös mahdollisuus käyttää tietokuvauksia ohjelmien tietokuvausten määrittelyssä, jolloin varmistetaan ohjelmien ja tietokantamääritysten yhteensopivuus. Tämä myös helpottaa muutosten hallintaa ja versiointia.
- UML-notaatio on myös tarkoitettu tuottamaan koodia. Esimerkkinä luokkakaavioita vastaavat luokkakirjastot.
- Huomioon otettava näkökohta on myös kuvausten siirrettävyys eri työvälineiden välillä. Esim. BPMN-notaatiosta on standardoitu siirtoformaatti, jonka tulisi mahdollistaa kuvausten siirtäminen eri työvälineiden välillä.

## 9.5 Työvälineiden alustavaatimukset

Yksi näkökohta työvälineiden jaottelussa on myös niiden asettamat vaatimukset käytettävälle alustalle. Työvälineet voivat olla sidottuja johonkin alustaan, tyypillisesti Windows-alustaan, tai niistä on saatavissa keskenään yhteensopivat alustakohtaiset versiot, kuten Windows, OS X ja Linux-versiot.

## 9.6 Kokonaisarkkitehtuurikehikoiden tuki

Kokonaisarkkitehtuurin laadinnassa ja hallinnassa saattaa tärkeäksi nousta myös tuki erilaisille KA-kehikoille. Esimerkkejä ovat mm. TOGAF-kehikkoa tukevat työvälineet tai Suomessa JHS 179 kehikkoa tukevat välineet.

# 10 Osaaminen ja kyvykkyydet

Kokonaisarkkitehtuurityö on liiketoiminnan ja tietotekniikan jatkuvaa kehittämistä vastaamaan asiakkaiden tarpeisiin, markkinoiden ja kilpailijoiden haasteisiin sekä organisaation oman toiminnan tehostamiseksi. Kokonaisarkkitehtuurityötä ei ole järkevää ulkoistaa satunnaisille konsulteille tai alihankkijoille. Kokonaisarkkitehtuurityön tulee olla yrityksen oman henkilöstön tekemää pitkäjänteistä ja suunnitelmallista työtä. Tästä syystä on välttämätöntä, että organisaatio itse hankkii ja ylläpitää kokonaisarkkitehtuurityössä tarvittavan osaamisen ja kyvykkyyden. Seuraavassa on tarkasteltu joitain osaamiseen ja kyvykkyyksiin liittyviä näkemyksiä. On hyvä havaita, että kokonaisarkkitehtuurityötä tekevän ryhmän koko on tarpeen suhteuttaa oman organisaation kokoon. Pienemmässä organisaatiossa jäljempänä lueteltuja rooleja voi olla yhdelle henkilölle annettu useampiakin.

## 10.1 TOGAF Architecture Capability Framework

Tyypillisesti organisaatioissa erillään olevia kyvykkyyksiä ovat

- Toiminta-arkkitehdit, prosessien ja liiketoiminnan mallintajat
- Tietoarkkitehdit, tietojen mallintajat, sisällön hallinta, tuotetiedon hallinta
- Tietojärjestelmäarkkitehdit
- Teknologia-arkkitehdit
- Tietosuoja- ja tietoturva-arkkitehdit

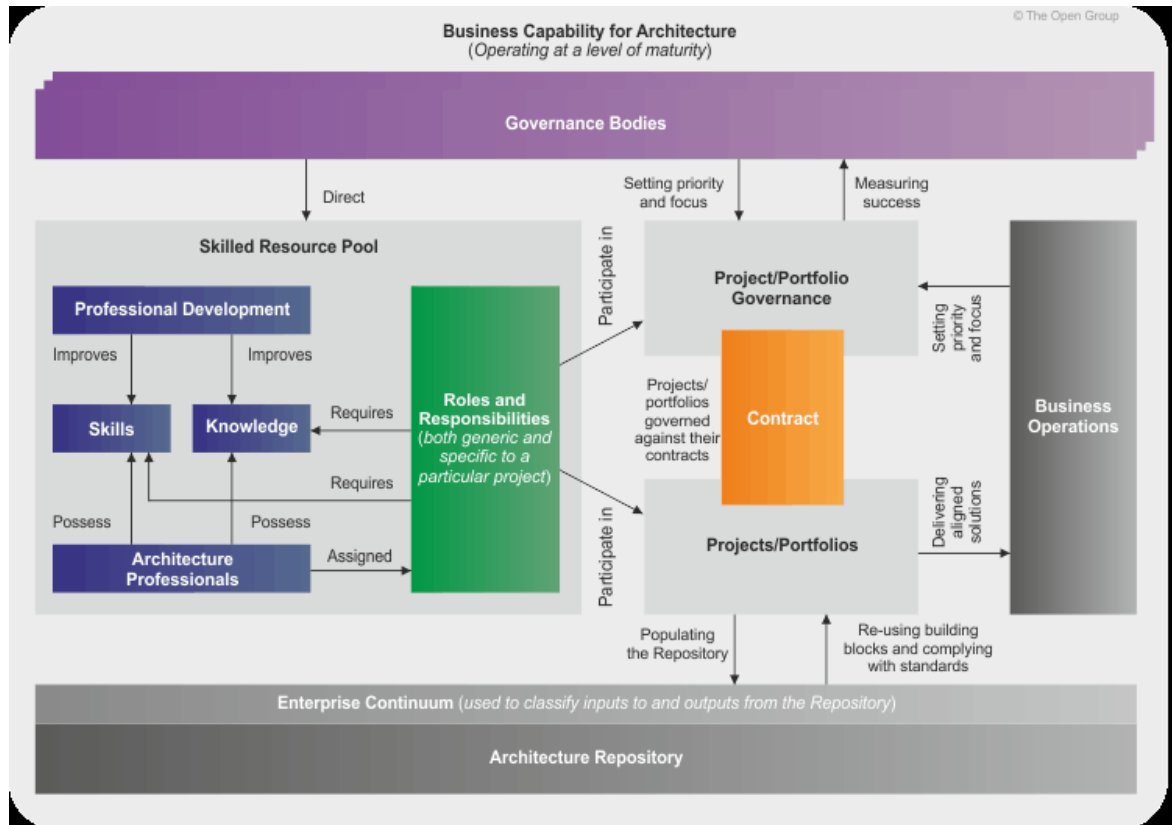
Kokonaisarkkitehtuurityössä tärkeätä on, että nämä eri osaamista edustavat arkkitehdit tekevät riittävissä määrin yhteistyötä, jotta varmistutaan syntyvien osa-arkkitehtuurien yhteensopivuudesta.

TOGAF 9 (TOGAF) luettelee erilaisia arkkitehtuurin osa-alueita ja kyvykkyyksiä seuraavasti:

- Financial Management
- Performance Management
- Service Management
- Risk Management
- Resource Management
- Communications and Stakeholder Management
- Quality Management

## Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

- Supplier Management
- Configuration Management
- Environment Management

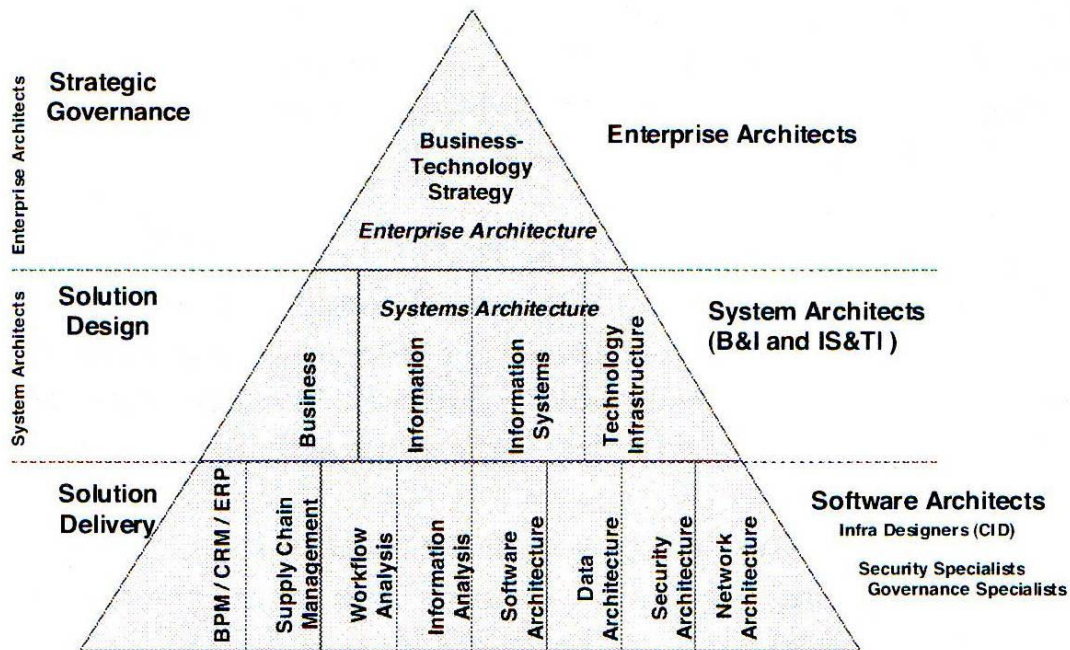


Kuva 32. TOGAF näkemys tarvittavista rooleista ja vastuista

TOGAF 9 toteaa, että kokonaisarkkitehtuuria tulee johtaa samoin periaattein, kuin muitakin liiketoiminnallisia yksiköitä ja siksi siihen tarvitaan yllämainittuja kyvykkyyksiä.



## 10.2 Schekkerman Who are involved in EA



Kuva 33. Yrityksen eri arkkitehtien roolit

Kuten edellä todettiin, kokonaisarkkitehtuurityössä tarvittavien kyvykkyyksien on tarpeen olla organisaation omassa hallinnassa. Näiden kyvykkyyksien kehittämisen ja ylläpitämisen tulee olla osa aktiivista henkilöstöhallintoa tietohallinnon ohella ja lisäksi. Kokonaisarkkitehdin uralle on tarpeen määrätietoisesti valita soveltuvia henkilöitä ja kierrättää heitä ympäri organisaatiota erilaisissa tehtävissä kokonaisnäkemyksen muodostumiseksi. Oheista Schekkermanin kuvaa voisi lukea niinkin, että ylempien tason arkkitehtien on hyvä olla mukana alemman tason arkkitehtien työssä ja varmistumassa eri arkkitehtuurien yhteensopivuudesta. (Schekkerman, 2004)

## 11 Menetelmien ja välineiden yhteensovittaminen arkkitehtuurityössä

Edellisissä luvuissa on kuvattu varsin kattavasti eri kokonaisarkkitehtuurimenetelmiä ja niiden keskeisiä piirteitä. Kokonaisarkkitehtuurimenetelmien valinnassa ja käyttöönotossa tulee kuitenkin muistaa, että mikään menetelmä ei ratkaise toiminnan ongelmia tai sovellu sellaisenaan suoraan organisaation käyttöön. Menetelmät ovat välineitä, joiden käyttötapa, -kohteet ja -periaatteet on sovittava kunkin organisaation omiin lähtökohtiin, tarpeisiin ja kyvykkyyksiin.

Kokonaisarkkitehtuurin tavoite on tarjota kokonaisvaltainen välineistö sekä organisaation eri osien että niiden muodostaman kokonaisuuden toiminnan kehittämiseen, ohjaamiseen ja seurantaan. Tässä roolissa kokonaisarkkitehtuuri ”leikkaa” koko organisaation läpi ja on siten kosketuksissa myös organisaation käyttämien muiden kehitys-, suunnittelu- ja ohjausmenettelyiden kanssa. Tästä syystä valittava kokonaisarkkitehtuurimenetelmistö ja -välineistö on sovittava yhteen muiden organisaationssa käytettävien viitekehysten ja kehitys- ja ohjausvälineiden kanssa. Toimiva kokonaisarkkitehtuuri ja sen johtaminen eivät voi toimia erillään muista organisaation ohjausvälineistä.

Eri kokonaisarkkitehtuurimenetelmillä on omat vahvuutensa ja rajoitteensa eikä ole olemassa kaikille organisaatioille sopivaa ”one-size-fits-all” –mallia. Kuten aiemmin on todettu, kokonaisarkkitehtuurityön tuloksena syntyvät erilaiset mallit ja kuvaukset ovat vain käsitteellisiä kuvauksia nykytilasta tai tulevaisuuden tahtotilasta. Toimiva kokonaisarkkitehtuuri syntyy niistä johtamis-, ohjaus- ja toimeenpanoprosesseista joilla kuvatut tahtotilat muutetaan organisaation toiminnaksi ja todellisuudeksi. Kokonaisarkkitehtuurin jalkauttamisessa ja toimeenpanossa avainasemassa onkin valitun menetelmän tai menetelmien sovittaminen organisaation tarpeisiin ja kyvykkyyksiin. Menetelmän, välineen tai mallin käyttöönotto ei ole kertaluontoinen projekti vaan jatkuvan kehitystyön alku.

Kokonais- ja palveluarkkitehtuurin johtamiseen, kehittämiseen ja hallintaan käytettävien viitekehysten ja mallien käyttöönotto ja sovittaminen päivittäiseen toimintaan kannattaa suunnitella aina organisaatiokohtaisesti. Sovitustyön lähtökohtana tulee olla liiketoiminnan asettamat tavoitteet ja vaatimukset. Kutakin viitekehystä ja mallia kannattaa käyttää sen omalla vahvuusalueella ja käytettävien mallien yhteensovittaminen tulee tehdä tarkoituksenmukaisesti siten, että ne yhdessä muodostavat mahdollisimman tehokkaan kokonaisuuden.

Käytännössä tietotekniikan johtamisen organisaatiossa tapahtuu eri viitekehysten kautta. Kokonaisarkkitehtuurin kannalta näistä viitekehyksistä tärkeimpiä ovat tietohallinnon korkean tason johtamiseen ja päätöksentekoon keskittyvä Cobit (Control Objectives for IT) sekä ensisijaisesti IT-palvelutuotantoa ohjaava ITIL. Tietotekniikan ja tietohallinnon ulkopuolisia rajapintoja ovat tyypillisesti johtamisjärjestelmän ”strategiakello” ja ”tulokortit”.

## 12 Yhteenveto

Kokonaisarkkitehtuurin menetelmien ja välineiden avulla pyritään luomaan edellytykset monimutkaisen sosioteknisen kokonaisuuden jäsentämiseen, hallintaan ja päätöksentekoon. Hallinnan välineet muodostuvat jäsennesskehikoista, arkkitehtuurityötä ohjaavista menetelmistä, käytettävistä kuvaustavoista ja notaatioista, tuotettavista arkkitehtuurikuvauksista ja malleista, työvälineistä sekä arkkitehtuurin osaamisesta ja kyvykkyyksistä. Näiden osa-alueiden tiedostaminen kokonaisarkkitehtuurin kehittämisessä edesauttaa omassa toimintaympäristössä ja organisaatiossa soveltuvien ratkaisujen löytämistä.

Erityisen olennaista muodostaa selvä kuva ja yhteisymmärrys siitä, mikä on arkkitehtuurityön kohteena oleva kokonaisuus. Kokonaisarkkitehtuurin menetelmät ja välineet ovat käyttökelpoisia organisaatioiden yhteistyöverkostoissa, laajoissa konserneissa, yksittäisissä organisaatioissa, kuin myös rajattujen kohdealueiden tai laajojen projektien hallinnassa ja jäsentämisessä. Työn kohteesta riippuen työssä painottuvat kuitenkin erilaiset tavoitteet ja kuvaukset. Saman jäsennessmallin osan (esimerkiksi loogisen tason tietoarkkitehtuurikuvaukset) tulkinta ja sisältö voi olla täysin erilainen esimerkiksi organisaation kokonaisarkkitehtuurin ohjauksessa ja yksittäisessä integraatioprojektissa (Virkanen ym. 2012). Nykytilan ja tavoitetilan erottaminen on erityisen olennaista, kun tehtyjen arkkitehtuurikuvausten avulla viestitään ja ohjataan työskentelyä. Onkin tarpeen muodostaa kehittämisen kohdealueelle sopiva systemaattinen malli siihen, kuinka erilaisten tarpeiden, vaatimusten ja ratkaisujen hallintaa pystytään tukemaan, jäsentämään ja vastuuttamaan kokonaisarkkitehtuurin avulla (Tiihonen ym. 2012).

Palvelukeskeisyys on noussut sekä yritysten toiminnan että tietojärjestelmien kehittämisessä keskeiseksi ajattelutavaksi. Palvelukeskeisyyden perusajatuksena on jäsentää kokonaisuus palvelujen avulla. Keskeisiä elementtejä ovat kokonaisuuden jakaminen osiin, palvelun tuottajan ja käyttäjän erottaminen, sekä palvelun tuottajan kuvaus palvelusta sopimisen pohjana. Palvelukeskeisyys kehit-

tämisperiaatteena on syytä erottaa palvelupohjaisesta tietojärjestelmäarkkitehtuurista (SOA), joka keskittyy tietojärjestelmä- ja ohjelmistoratkaisujen suunnitteluun ja palvelupohjaisesta tietojenkäsittelystä, joka luo tekniset puitteet ohjelmistopalvelujen kehittämiseksi, integroinnille ja käytölle.

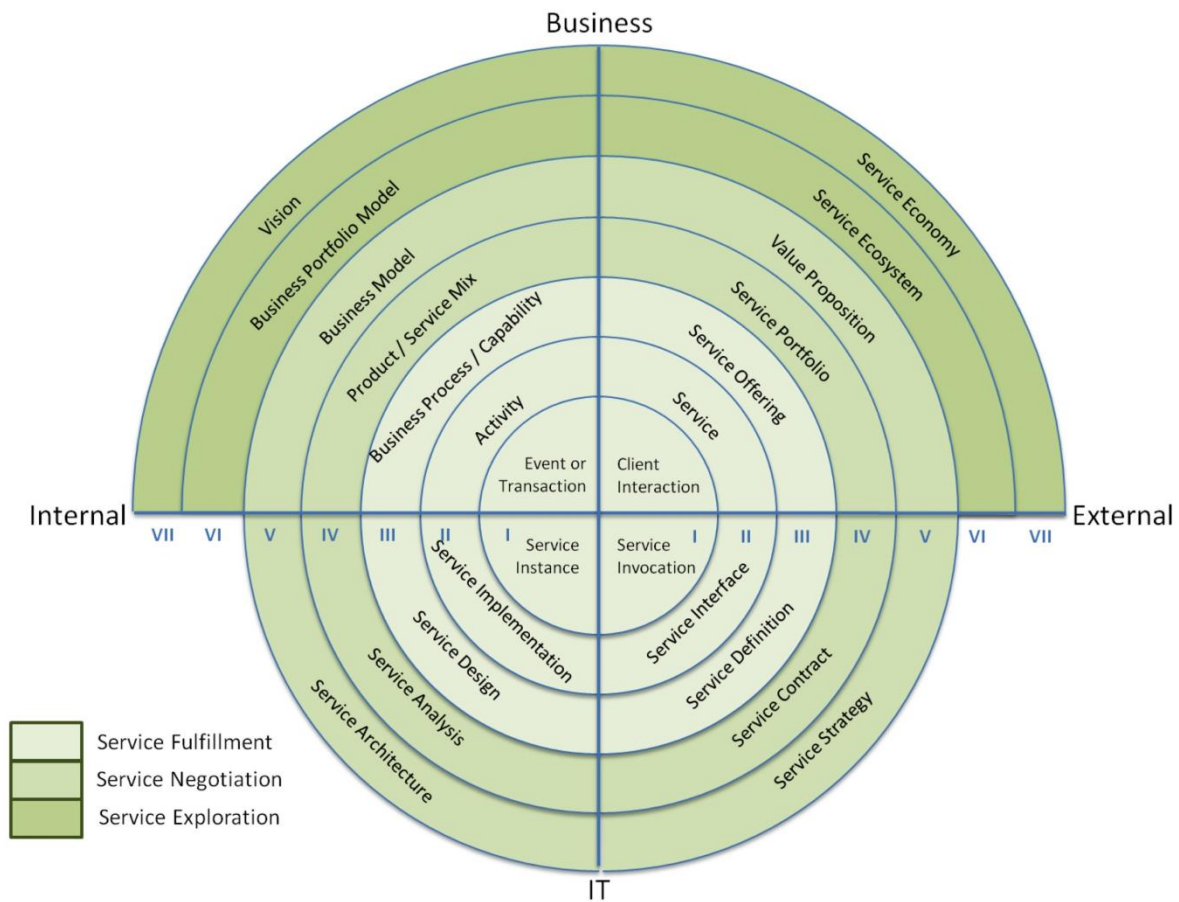
Palvelukeskeisyyttä on mahdollista korostaa organisaation kokonaisarkkitehtuurissa hyödyntämällä jo nykyisten kokonaisarkkitehtuurikehikoiden, mallinnusmenetelmien ja notaatioiden tarjoamia mahdollisuuksia. Esimerkiksi mallinnusmenetelmissä kuten Archimate ja eri kehikoissa suositelluissa arkkitehtuurikuvauksissa on jo runsaasti palvelukeskeisiä elementtejä. Ongelmana on usein kuitenkin, että "palvelu" käsitteenä voi esiintyä hyvin monilla eri abstraktiotasolla. Väärinymmärrysten välttämiseksi on syytä muodostaa yhdenmukainen sanasto, jonka avulla erityyppisiä ja -tasoisia palveluja kuvataan ja nimetään.

Palvelukeskeisen sanaston ja lähestymistavan käyttöönotto sekä erityyppisten palvelujen ominaisuuksien selkeä määrittely auttavat kaventamaan (liike)toiminnan ja teknisen kehittämisen välistä kuilua luomalla yhteistä kieltä johdon, operatiivisen toiminnan ja kehittämisen ja sen ohjauksen välillä. Eräs keskeinen havainto on kuitenkin, että mikäli kokonaisarkkitehtuuri, palvelukeskeisyys ja tuotetut kuvaukset mielletään organisaation johdon näkökulmasta "IT-osaston" toiminnaksi, kuilun kaventaminen ei onnistu. Johdon sitoutuminen arkkitehtuurityön tavoitteisiin on perusteltavissa vain saavutettujen hyötyjen kautta. Sitoutumattomuuden riskiä voidaan pyrkiä vähentämään korostamalla strategialähtöisyyden, jäljitettävyyden, toiminta-arkkitehtuurin ja kehittämispäätösten tukemisen merkitystä. Toimintamallien ja tavoitteiden kuvauksiin keskittyvät kuvaustavat kuten Business Model Canvas -elementit ovat hyödyllisiä osana kokonaisarkkitehtuuria, jos ne voidaan sovitaa tarkempaan kehittämiseen ohjaaviin kuvauksiin. Arkkitehtuurikuvausten tuottamisessa ja hyödyntämisessä on aina huomioitava kohderyhmä ja pyrittävä piilottamaan käytettävistä kuvauksista epäolennaiset elementit ja yksityiskohdat. Erityisesti käsitteellisellä tasolla on keskeistä piilottaa yksityiskohdat ja muodostaa selkeitä ja ymmärrettäviä kokonaiskuvauksia. Myös monimutkaisten kokonaisuuksien visualisointi ja rakenteistaminen eri kuvaustasolla helpottavat viestintää ja ymmärrettävyyttä.

Kokonaisarkkitehtuurin kehittämisessä voidaan saavuttaa hyötyjä palvelupohjaisesta toimintamallista eri tasoilla (Kuva 34) (Korhonen ym. 2011). Liiketoiminnan ulkoisesta näkökulmasta palvelukeskeisyys tiivistyy organisaation arvolupaukseen, joka voidaan koostaa sen tarjoamien palvelujen pohjalta (palvelusalkku ja palvelutarjoama). Organisaatioiden välisissä yhteistyöverkostoissa voidaan pyrkiä luomaan palveluekosysteemejä. Myös kokonaisten palvelujärjestelmien tasolla on nähtävissä siirtymä tuote- ja tuotantokeskeisyydestä palvelutalouteen. Organisaation toiminnan sisäisessä suunnittelussa ja johtamisessa liiketoiminnan visio, liiketoimintamahdollisuudet, toimintamalli, sekä tuote- ja palveluvalikoima voidaan jäsentää palvelukeskeisesti. Vastaavasti organisaation toiminnassa prosessit, toiminnot ja teot on mahdollista jäsentää ja organisoida palvelujen kautta. Sekä kehittämistyön että päivittäisen toiminnan johtamisessa on tarpeen tunnistaa eri tasot ja aikahorisontit sekä kokonaisarkkitehtuurin, tietotekniikan että palveluarkkitehtuurin hallinnointimallissa (Hiekkanen ym. 2012).

Tietotekniikan näkökulmasta palvelukeskeisyys näkyy ulospäin erityisesti palvelusopimusten ja organisaatioiden tarjoamien palvelumäärittelyjen ja -rajapintojen kautta. Sähköisessä toimintaympäristössä palvelukeskeiseen organisaatioiden yhteistoimintaan on sovittava tarkalla tasolla käytettävät rajapinnat ja integraatoratkaisut (ks. Mykkänen ym. 2012). Sisäisesti on huolehdittava siitä, että tietojärjestelmäpalvelut voidaan hallita yhtenäisen arkkitehtuurin avulla ja että niiden analyysissä, suunnittelussa ja toteutuksessa tai hankinnassa huomioidaan liiketoiminnan tarpeet ja tavoitteet. Palvelukeskeisten järjestelmien, sovellusten ja rajapintojen osalta on huolehdittava myös teknisen infrastruktuurin toimivuudesta ja riittävydestä. Palvelukeskeisyys korostuu monissa nykyaikaisten tietojärjestelmien hankinta- ja kehitystavoissa keskeisten järjestelmien modularisoinnista pilvipalve-

luihin. Palvelukeskeiseen tietojärjestelmien kehittämistapaan siirtymisen on kuitenkin tapahtuttava hallitusti ja vaiheittain (Rommel 2009).



Kuva 34. Palvelukeskeisyys liiketoiminta/IT- ja ulkoinen/sisäinen -akseleilla (Korhonen ym. 2011).

Kussakin toimintaympäristössä on siis olennaista päättää, mitä kautta palvelukeskeisyyden hyötyjä ensisijaisesti tavoitellaan, ja kuinka syvälle palvelukeskeisyys viedään osaksi organisaation toimintatapoja. Palvelukeskeisen tietojärjestelmäarkkitehtuurin avulla on mahdollista tukea myös organisaatioita ja verkostoja, joissa koko liiketoimintaa ei ole jäsenetty palvelukeskeisesti. Kehitystyössä on usein yhdistettävä sekä ylhäältä-alas strategia- ja prosessilähtöistä että alhaalta-ylös integraatio- ja vaatimislähtöistä lähestymistapaa. Tasoittainen toiminnan ja prosessien kuvaaminen (Luukkonen ym. 2012) sekä eri sidosryhmien tunnistaminen (Tiihonen ym. 2012, Eloranta 2008) ovat tarpeellisia elementtejä tässä työssä.

Keskeisten kokonaisarkkitehtuurin elementtien, kuvaustapojen ja välineiden valinta riippuu siis vahvasti kohdeympäristöstä sekä kuvaamisen tavoitteista. Käytettävien menetelmien ja välineiden vaatimukset riippuvat myös osallistujien määrästä, taustasta ja organisaation kehittämiskulttuurista. Kokonaisarkkitehtuurin hyödyt muodostuvat vain tuotettujen kuvausten hyödyntämisen kautta, joten keskeistä on myös valita, kuinka suurta kuvausmassaa on tarkoituksenmukaista hallita ja ylläpitää. Kuvausten ohjausvaikutus voi olla parempi, jos hyödyntämisperiaatteena "vähän mutta sitovia ja tarpeellisia" sen sijaan että pyrittäisiin luomaan tarkat ja kattavat määrittelyt joiden yhdenmukaista noudattamista ei kyetäkään takaamaan.

Kokonaisarkkitehtuurikehikoiden ja kuvaustapapohjien soveltamisessa on kuitenkin myös riskinsä. "Suositeltujen kuvausten" tai kuvaspohjien soveltaminen ilman selvää kuvaa niiden käyttötarkoituksesta ja käyttäjistä johtaa resurssien tuhlaukseen arkkitehtuurityössä ja hyödyttömien dokumentaatiopinojen tuottamiseen, ja on voinut pahimmillaan johtaa "paperipinoarkkitehtuureihin", joilla ei ole haluttua ohjausvaikutusta. Lisäksi on esimerkkejä eri arkkitehtuurinäkökulmien liiallisesta korostamisesta tai eriyttämisestä (esimerkiksi "tietoarkkitehtuuria" kehittäminen irrallaan toiminnasta ja tietojärjestelmästä), joka voi johtaa yhteen sopimattomiin ratkaisuihin tai malleihin. Paradoksi on, että tällaisia tilanteita pyritään nimenomaisesti välttämään kokonaisarkkitehtuurin avulla. Työn organisoinnissa onkin huolehdittava siitä, että eri näkökulmat ja kuvaukset täydentävät ja hyödyntävät toisiaan, ja että arkkitehtuurityön keskeisistä elementeistä on yhteinen näkemys (Virkanen ym. 2012).

Kattavassa ja monitasoisessa kokonaisarkkitehtuurityössä on riskinä myös, että ylhäältä alas muodostuvat suuntaviivat ja määrittelyt eivät vastaa riittävän nopeasti ajankohtaisiin kehittämistarpeisiin. Käytännön tarpeet ja projektit eivät yleensä voi "jäädä odottelemaan" kattavien kokonaisarkkitehtuuri-ohjeistusten kokoamista ja hyväksymistä. Käytännössä kehittämishankkeita ja projekteja on voitava myös hyödyntää kokonaisarkkitehtuurin kehittämiseen niiden omien välittömien kehittämistarpeiden lisäksi.

Kokonaisarkkitehtuuria on mahdollista hyödyntää merkittävästi yhteentoimivuuden kehittämisessä. Kokonaisarkkitehtuurin tyypillisin pääasiallinen käyttökohde on kuitenkin parantaa monimutkaisten kokonaisuusien hallittavuutta ja päätöksentekoa yleisesti. Yhteentoimivuuden edistämässä ja integraatioissa korostuvatkin usein eri arkkitehtuurikuvaukset kuin yleisesti kokonaisarkkitehtuurin hallintaan keskeisimmiksi ehdotetut (ks. luku 8.2) kuvaukset. Myös yhteentoimivuuskuvausten ja standardien kehittämisessä käytetään yhä enemmän kokonaisarkkitehtuurimalleja (Mykkänen ym. 2012).

Kokonaisarkkitehtuurikehikoita ja "parhaita käytäntöjä" kuten TOGAF on arvosteltu siitä, että ne eivät riittävästi perustu tieteellisiin teorioihin ja formaaleihin malleihin (Dietz 2009). Vaikka mallien formalisointi ja teoretisointi edesauttaa järjestelmällisyyden ja automaation saavuttamista, on kuitenkin arkkitehtuurityön ensisijaisena tavoitteena oltava tulosten käyttökelpoisuus päätöksenteossa ja viestinnässä eri tilanteissa. "All models are wrong, some are useful" -periaate<sup>1</sup> ja omaan käyttökontekstiin soveltuvien menetelmien, välineiden ja kuvausten valinta luovat pohjan arkkitehtuurityön onnistumiselle.

Arkkitehtuurityön kannalta keskeisiä kysymyksiä ovat (Schekkerman 2011):

- miksi työtä tehdään?
- mitä työn tuloksena syntyy?
- kenelle tuloksista on iloa?

Jos mikä tahansa yllä olevien kysymysten vastauksista on epäselvä, työ on syytä keskeyttää ja selvittää sen tavoitteita. Tässä dokumentissa ja muissa SOLEA-hankkeen tulodokumenteissa on pyritty tiivistämään osa hankkeen aikana työpajoissa, kartoituksissa, tutkimuksissa ja kehitysyhteistyössä käytetyistä ja kehitetyistä malleista, joilla näihin kysymyksiin vastaamisen jälkeen voidaan tukea järjestelmällistä kokonaisarkkitehtuurityötä sekä palvelukeskeisen ajattelutavan hyödyntämistä.

---

<sup>1</sup> attributed to W. Edwards Deming (1947) and George E.P. Box (1979)

## Lähteet

ArchiMate 2012. The Open Group, <http://www3.opengroup.org/subjectareas/enterprise/archimate>, viitattu 22.5.2012

Architect Corner: Zachman Framework v3.0 - what's NEW?: <http://blogs.icmgworld.com/2012/02/02/zachman-framework-graphics-v3-0-whats-new/>

Arsanjani A, Liang-Jie Z, Ellis M, Allam A, Channabasavaiah K. S3: A Service-Oriented Reference Architecture. *IT Professional* 2007:3, 10-17.

Bonitasoft: <http://www.bonitasoft.com/products>

Dietz J. Is it φτυψ or bullshit? Afscheidsrede 16 oktober 2009, Technische Universiteit Delft.

EA Wiki; <http://iea.wikidot.com/start>

Eloranta L. Stakeholder Identification in Service Oriented Architecture Projects. Master's thesis. Espoo: Helsinki University of Technology, 2008.

Greefhorst D. Using TOGAF as a pragmatic approach to architecture, KIVI NIRIA, afd. Informatica, 15 April 2009.

Halpin T. Information Modeling and relational Databases, Morgan Kaufmann Publishers, 2001

Hed SR. ATK-Automaattinen tietojenkäsittely, Tammi, 1969

Hiekkänen K, Korhonen JJ, Mykkänen J, Itälä T. Kokonaisarkkitehtuurin ja palveluarkkitehtuurin hallinnointimallit. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.

ISO/HL7 2006. Health informatics - HL7 version 3 - Reference information model - Release 1. ISO/HL7 21731:2006.

ISO/IEC 1995. Reference Model of Open Distributed Processing, Part 1. Overview. ODP Reference Model ITU-T X.901 | ISO/IEC 10746-1: 1995 DIS (E); 1995.

JHS 179. JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen: <http://www.jhs-suositukset.fi/suomi/jhs179> viitattu 21.5.2012.

Jin M, Kung D, Peng W. Research of information system technology architecture. *2010 2nd International Conference on Industrial and Information Systems (IIS)*, July 2010, IEEE, Dalian, China, 2010, pp. 293-296.

Kartturi kokonaisarkkitehtuurimalli: <http://raketti.csc.fi/kokoa/kartturi> viitattu 21.5.2012.

Korhonen JJ, Hiekkänen K, Heiskala M. Mapping out service-oriented organization. Submitted manuscript, 2011.

Manifesto for Agile Software Development: <http://agilemanifesto.org/>



- Mcaulay A. Enterprise Architecture design and the Integrated Architecture Framework. Microsoft Architects Journal 1/2004, January 2004, Microsoft . Available: [http://download.microsoft.com/download/4/d/a/4da0ddee-77e0-47d1-aaa7-a5dd619b8bca/journal1\\_english.pdf.zip](http://download.microsoft.com/download/4/d/a/4da0ddee-77e0-47d1-aaa7-a5dd619b8bca/journal1_english.pdf.zip)
- Mykkänen J, Häyrynen K, Savolainen S, Porrasmaa J. Standardien arviointi ja valinta terveydenhuollon sovellusintegraatiossa. PlugIT-projekti, 2004.
- Mykkänen J, Itälä T, Savolainen S, Virkanen H. Yhteentoimivuus, standardit ja palveluarkkitehtuuri. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.
- OMG BPMN 2012. Documents Associated with Business Process Model and Notation (BPMN) Version 2.0: <http://www.omg.org/spec/BPMN/2.0/> viitattu 21.5.2012.
- OMG SoaML 1.0, 2012. Service oriented architecture Modeling Language (SoaML) <http://www.omg.org/spec/SoaML/> viitattu 21.5.2012.
- OMG UML 2012. UML® Resource Page <http://www.uml.org/> viitattu 21.5.2012.
- Osterwalder A. The Business Model Ontology, a Proposition in a Design Science Approach. Université de Lausanne, 2004. [http://www.hec.unil.ch/aosterwa/PhD/Osterwalder\\_PhD\\_BM\\_Ontology.pdf](http://www.hec.unil.ch/aosterwa/PhD/Osterwalder_PhD_BM_Ontology.pdf), viitattu 22.5.2012
- Osterwalder A, Pigneur Y, Smith A. The Business Model Canvas, 2010: <http://www.businessmodelgeneration.com/canvas>, viitattu 22.5.2012
- Schekkerman J. How to Survive in the Jungle of Enterprise Architecture Frameworks. Trafford, 2004.
- Schekkerman J. Enterprise Architecture Tool Selection Guide v 6.3, Institute For Enterprise Architecture Developments, 2011.
- Sessions R. *Comparison of the Top Four Enterprise Architecture Methodologies By Roger Sessions*. ObjectWatch White Papers, May 2007. Referenced 9.8.2011. [http://www.objectwatch.com/white\\_papers.htm#4EA](http://www.objectwatch.com/white_papers.htm#4EA)
- SOA Manifesto: <http://www.soa-manifesto.org/> viitattu 22.5.2012
- Tiihonen T, Itälä T, Mykkänen J, Järvinen J, Tamminen M, Savolainen S, Luukkonen I, Hiekkanen K. Tarpeiden ja vaatimusten hallinta kokonaisarkkitehtuurissa. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.
- Rommel K. SOA Roapmap: Migration path to Service-Oriented Architecture. Master's thesis. Espoo: Helsinki University of Technology, 2009. 84 p.
- Schekkerman J. The difference between Architecture Thinking and Architecture Doing - The Value of a Managed Diversity EA Approach. Presentation in SOLEA seminar 25. Nov 2011, Espoo, Finland.
- The Open Group, development of open, vendor-neutral IT standards and certifications : <http://www3.opengroup.org/>

The Open Group: *TOGAF. The Open Group Architecture Framework*, 2009.

<http://www.opengroup.org/togaf/> viitattu 22.5.2012

The Open Group: TOGAF 9, Establishing and Maintaining an Enterprise Architecture Capability:

[/http://www.togaf.org/togaf9/chap02.html](http://www.togaf.org/togaf9/chap02.html) viitattu 22.5.2012

Vallecillo A. RM-ODP: The ISO Reference Model for Open Distributed Processing. DINTEL Edition on Software Engineering. No. 3. ISBN: 84-931933-2-1, pp. 69-99. March 2001

<http://www.lcc.uma.es/~av/Publicaciones/00/odpeng.pdf> viitattu 22.5.2012

Virkanen H, Järvinen J, Mykkänen J, Sammelvuo I. Arkkitehtuurikuvausten kohteet ja kuvaustavat - kooste case-tutkimuksen tuloksista. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.

Virkanen H, Järvinen J, Mykkänen J, Sammelvuo I. *Arkkitehtuurikuvausten kohteet ja kuvaustavat - kooste case-tutkimuksen tuloksista*. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.

Vizenor & Smith 2004. Vizenor, L. & Smith, B. Speech Acts and Medical Records: The Ontological Nexus. 2004. [http://ontology.buffalo.edu/medo/EuroMISE\\_HL7.pdf](http://ontology.buffalo.edu/medo/EuroMISE_HL7.pdf) viitattu 22.5.2012

VM 2011. Julkisen hallinnon kokonaisarkkitehtuuri - Kokonaisarkkitehtuurin yleiskuvaus. Määrittely, 0.95. 4.4.2011, Valtiovarainministeriö.

Yhteentoimivuus.fi, Yhteentoimivuuden tietopankki:

<https://www.yhteentoimivuus.fi/view/index.xhtml> viitattu 22.5.2012

Zachman JA. *John Zachman's Consice Defenition of the Enterprise Framework*.

<http://www.zachman.com/>. Viitattu 21.5.2012



## Liite 1. Sanasto SOLEA-hankkeen keskeisistä käsitteistä

Käsite	Kuvaus
<b>Ajurit</b>	Ulkoiset toimintaympäristön tekijät, jotka vaikuttavat toimintaan tai sen tavoitteisiin (erityisesti kontekstitason tarkastelu), kuten lainsäädäntömuutokset, markkinoiden tai palvelujen kysynnän kehitys, mille ei tunnistettavissa sisäistä omistajaa.
<b>Aktiviteetti-kaavio</b>	Kaaviotyyppi, joka on tarkoitettu erityisesti aktiviteettien, kuten työkulkujen, liiketoimintaprosessien sekä rinnakkaisia toimintoja sisältävien järjestelmien sekä niiden välisten suhteiden sekä UML versiossa 2 prosessien kuvaamiseen (Fowler 2004, KuntaIT).
<b>Aliprosessi</b>	Aliprosessi (esim. yksittäinen SOA-palvelu) on pääprosessin osa. Vrt. Osaprosessi.
<b>Arkkitehtuuri-periaate</b>	Yleinen kokonaisarkkitehtuuria tai jotain sen osa-aluetta yli useiden eri projektien ohjaava periaate, jonka perusteella voidaan tehdä valintoja erilaisten vaihtoehtojen välillä. Periaatteet ovat yleisempiä kuin linjaukset eli niitä ei ole välttämättä kohdistettu mihinkään yksittäiseen kehittämiskohteeseen.
<b>Asiakasprosessi</b>	Prosessi, jossa kuvataan asiakkaalle annettava palvelu. Lähellä ydinprosessia joissakin tapauksissa.
<b>Asiantuntijatyön prosessi</b>	Prosessi, jonka vaiheiden järjestys tai sisältö perustuu tyypillisesti jossain vaiheessa prosessia asiantuntijan hiljaisen tiedon tai kokemuksen hyödyntämiseen ja asiantuntijuuteen, jota on vaikea automatisoida. Usein dynaaminen.
<b>Automatisointi</b>	Manuaalisten työvaiheiden tuottaminen tietoteknologian avulla.
<b>Dynaaminen prosessi</b>	Prosessi, jonka vaiheet tai niiden järjestys eivät ole tarkalleen etukäteen määriteltäviä; prosessin osat voivat olla suunnilleen samoja, mutta suoritusjärjestys vapaampi. Vrt. Staattinen prosessi.
<b>EA</b>	Ks. Kokonaisarkkitehtuuri (engl. Enterprise Architecture)
<b>Ei-toiminnalliset vaatimukset</b>	Määrittelee rajoitukset ja reunaehdot toiminnallisille vaatimuksille. Ei-toiminnalliset vaatimukset eivät liity suoraan palveluihin vaan kertovat, mitä ehtoja järjestelmän on täytettävä, jotta toiminnalliset vaatimukset voidaan toteuttaa (JHS 173). Esim: vasteaikavaatimus ja saumattomuus.
<b>IT-järjestelmä</b>	IT-järjestelmällä tarkoitetaan organisaation koko teknistä järjestelmää sisältäen laitteistot, tietoverkot sekä ohjelmistot. Vrt. Tietojärjestelmä.
<b>JHS</b>	Julkisen hallinnon suosituksien ( <a href="http://www.jhs-suositukset.fi">www.jhs-suositukset.fi</a> ).
<b>Järjestelmävaatimus</b>	Ilmaisee mitä, millä ehdoin ja kuinka hyvin järjestelmän on tehtävä (jotain) tai millainen sen on oltava (reunaehto) sidosryhmien tarpeiden tyydyttämiseksi (JHS 173).
<b>Kehittämistavoitteet</b>	Organisaation sisäiset tietyn kokonaisuuden kehittämiseen liittyvät tavoitteet, esim. tietyn toiminnon tehostaminen, tietojärjestelmäkokonaisuuden hankinta tai kehittäminen, uuden toimipisteen tai kumppanin hankinta, uuden prosessin kehittäminen, prosessin uudelleensuunnittelu.
<b>Kohdearkkitehtuuri</b>	Kohdearkkitehtuuri on yhteenkuuluvan rajatun alueen arkkitehtuurikokonaisuus kattaen toiminnan, tiedon, järjestelmät ja teknologiat. Se luo puitteet kyseisten keskitettyjen palveluiden tarkemmalle suunnittelulle ja toteuttamiselle jäsentäen ja määrittäen arkkitehtuurin keskeiset rakenneosat.

Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

<b>Kokonaisarkkitehtuuri</b>	Synonyymi: Yritysarkkitehtuuri, joka on yksityisellä sektorilla käytetty nimitys kokonaisarkkitehtuurista (suom. KA, engl. Enterprise Architecture, EA). <i>Menetelmäpainotteinen määritelmä:</i> Kokonaisarkkitehtuurilla tarkoitetaan toiminnan, tietotarpeiden, tietojärjestelmien ja teknologiaratkaisujen mallintamista, kuvaamista ja suunnittelemista yhtenäisen mallin mukaisesti (KuntaIT). <i>Tuotospainotteinen määritelmä:</i> Kokonaisarkkitehtuuri on toiminnan, prosessien ja palvelujen, tietojen, tietojärjestelmien ja niiden tuottamien palvelujen muodostaman kokonaisuuden rakenne (JHS 171).
<b>Kuvaus</b>	Ks. Malli.
<b>Kuvaustapa</b>	Kuvaustapa on kuvaustyyppin tarkennus, esim. TOGAF:in prosessitaulukko. Samaa kuvastapaa (esim. uimaratakaavio) voidaan tarvittaessa toteuttaa erilaisten notaatioiden avulla.
<b>Kuvaustaso</b>	Kuvaustaso kertoo, miten tarkalla tasolla kuvauksen kohdetta, kuten prosesseja tai toimintaa, kuvataan/mallinnetaan, kuinka suuren (organisatorisesti) ja yleistettävän (yksityiskohtien abstrahointi) kokonaisuuden kuvaus kattaa; esim. organisaatiotason yleiskuva, yksi prosessi, henkilön tai yksikön toiminta, palvelu tai sen operaatio.
<b>Kuvaustyyppi</b>	Kuvaustyyppi on tuotettavien kuvausten perusmuoto/-rakenne, kuten matriisi, kaavio, taulukko, teksti, hakemisto tai lista.
<b>Käyttjävaatimus</b>	Määrämuotoinen ilmaisu siitä mitä, kuinka hyvin ja millä rajoituksin käyttäjä (tai muu sidosryhmä) haluaa järjestelmällä tehdä tai aikaansaada, tai mitä ominaisuuksia järjestelmän on omattava. Vaatimuksella on varottava ilmaisemasta erityistä ratkaisua tarpeeseen tai ongelmaan (JHS 173).
<b>Liiketoimintastrategia</b>	Liiketoiminta- eli kilpailustrategian voidaan sanoa olevan se toimintatapa, jolla yritys kilpailee markkinoillaan, ja kuinka se yrittää luoda kilpailijoihinsa nähden kilpailuetua luoden edellytykset yrityksen olemassaololle (Simons, 1990). Lähdetessä määrittelemään yrityksen kilpailustrategiaa on edellytyksenä se, että on ratkaistu, missä liiketoiminnassa ja millaisin päämäärin ollaan liikkeellä. Tämä määrittely lukitsee samalla toimialan, missä yritys operoi, tuotteet mitä se asiakkailleen tarjoaa ja markkinat missä se toimii.
<b>Linjaus</b>	Linjaus kohdistuu johonkin määriteltyyn kehittämisen kohteeseen. Linjaus on vastaavan tahon hyväksymä ja se on ainakin jossain määrin sitova. Se kuvaa, mitä tullaan tekemään tai mitä kuvausta tai tarkempaa määrittelyä pitää käyttää tiettyyn määriteltyyn kehittämisen kohteeseen. Linjauksia voi olla myös suosituksissa, standardeissa, asetuksissa ja laeissa.
<b>Malli</b>	Malli koostuu kuvauksen sisällöstä ja kuvaustavasta. Se, mille tasolle yksittäinen malli kuuluu, määräytyy sisällön perusteella.
<b>Notaatio</b>	Notaatiolla tarkoitetaan mallinuskien graafisia komponentteja. Notaatio tarkoittaa sääntöä, jonka mukaan menetelmän käsitteistöä mallinnetaan. Notaatio ottaa kantaa siihen, kuinka esim. kaaviossa esiintyvä luokka esitetään, esim. suorakaiteella vai ympyrällä (KuntaIT).
<b>Ohjelmisto</b>	<i>Ohjelmisto</i> tai <i>tietokoneohjelmisto</i> on useista tietokoneohjelmista, niiden käyttämistä tiedostoista ja niihin liittyvästä dokumentaatiosta muodostuva kokonaisuus.
<b>Oletus</b>	Tiettyyn kehittämisen kohteeseen tai ympäristötekijään liittyvä oletus tulevasta kehityksestä tai nykytilasta, jota ei voida pitää varmana. Voidaan käyttää perusteluina erilaisille tavoitteille ja ratkaisuvaihtoehdoille.
<b>Organisaatio</b>	Hallinnollinen yksikkö, esim. Kuopion yliopistollinen sairaala.

Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

<b>Organisaatio-yksikkö</b>	Organisaation sisällä oleva yksikkö, esim. ihotautipoliklinikka.
<b>Osaprosessi</b>	Osaprosessit ovat ydin- tai tukiprosessien osia (JHS 152). Kuvaamistavat kuten prosessilla, mutta yksityiskohtaisemmin. Voi olla myös aliprosessi.
<b>Osatoiminta</b>	Toimintaan kuuluva, pienempi kokonaisuus. Kuvataan kuten toiminta.
<b>Palvelu</b>	<i>Toiminnallisessa arkkitehtuurissa:</i> Sopimuksen avulla kuvattu joukko ominaisuuksia, joiden avulla palvelun tarjoaja tuottaa haluttuja tuloksia palvelun käyttäjälle.  <i>Sovellusarkkitehtuurissa:</i> Sopimuksen / määrittelyn avulla kuvattu joukko palvelun tarjoajan tarjoamia tietoja ja toimintoja, joiden avulla palvelun käyttäjä pystyy kokoamaan prosesseja tai sovelluksia. SOA-arkkitehtuurissa palvelut toimivat keskiössä ja ovat näin järjestelmien toiminnan edellytys. Palvelut toimivat toiminnallisuuden ja sovelluslogiikan rakennusosina (KuntaIT).
<b>Palvelukeskeisyys</b>	Palvelukeskeiseen avoimeen arkkitehtuuriin kuuluu perusinfrastruktuuri ja valmiita yleiskäyttöisiä tukipalveluja, komponentteja ja rajapintoja, joita voidaan suoraan hyödyntää palvelujen rakentamisessa (KuntaIT).
<b>Palvelupohjainen arkkitehtuuri (SOA, service-oriented architecture)</b>	Tietojärjestelmien kehittämisen lähestymistapa, jossa sovelluksia tai toimintaprosesseja muodostetaan pienemmistä, määriteltyjä osatehtäviä toteuttavista palveluista. Tietojärjestelmäkokonaisuus hahmotetaan joukkona palveluita (sovelluspalveluita), joita tarpeen mukaan yhdistelemällä voidaan entistä helpommin toteuttaa tai mukauttaa sovelluksia eri käyttötarpeisiin (Mykkänen 2004).
<b>Poikkeus</b>	Prosessissa tai palvelussa tapahtuva normaaliin työn tai suorituksen kulkuun kuulumaton tapahtuma, joka tyypillisesti estää etenemisen tai odotettujen tulosten tuottamisen.
<b>Prosessi</b>	Sarja toisiinsa liittyviä toistuvia toimintoja tai tapahtumia, joiden avulla prosessiin liittyviä resursseja käyttäen päästään lisäarvoa tuottavasti toivottuun tulokseen. Prosessilla on selkeä alku ja loppu.
<b>Prosessiaskel</b>	Prosessiaskel tarkoittaa toiminnan etenemistä eli prosessin tai sen osan siirtymistä vaiheesta toiseen (JHS 152). Vrt. Prosessin vaihe.
<b>Prosessikaavio</b>	Prosessikaavio on tapa kuvata prosessin toiminnot graafisesti. Prosessin toiminnot, tietovirrat ja tuotteet kuvataan sovitulla symboleilla. Prosessikaavio auttaa ymmärtämään toimintojen järjestystä ja niiden välisiä riippuvuuksia (JHS 152). Voi olla esimerkiksi vuokaavio.
<b>Prosessikartta</b>	Organisaation tasolla tehty yleinen, usein graafinen kuvaus organisaation tärkeimmistä prosesseista ja niiden välisistä yhteyksistä (JHS 152).
<b>Prosessin frekvenssi</b>	Miten usein prosessi toistuu.
<b>Prosessin omistaja</b>	Prosessin omistaja on prosessin toiminnasta, tuloksesta, tuloksellisuudesta ja kehittämisestä vastuussa oleva henkilö (JHS 152). Prosessiin nimetty vastuullinen toimija, jonka tehtävänä on koordinoida omistamansa prosessin kuvausta, käyttöönottoa, vakiinnuttamista ja kehittämistä sekä seuranta.
<b>Prosessin vaihe</b>	Prosessille lisäarvoa tuottava toimijan (ihminen/sovellus) aliprosessi toiminto tai tapahtuma. Prosessin vaiheesta toiseen siirrytään prosessiaskelten kautta.
<b>Sekvenssikaavio</b>	Kaaviotyyppi, joka esittää uimaradoilla kuvattujen osapuolten väliset kutsut ja niiden välisen järjestyksen tai tietoliikenteen.

Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

<b>Rajoite</b>	Selkeästi ja tarkoituksella rajoittaa suunnittelua, toteutusta, käyttöä, elinkaarta tai päätöksentekoa; kehittämistä ohjaavat projektin linjaukset, joiden avulla rajoitetaan mahdollisten suunnittelupäätösten joukkoa. Esim. tietyn teknologian käyttö, lain asettama vaatimus, yhteisen tietomallin määrittelemien tietojen käyttö projektikohtaisen sijaan.
<b>Simulointi</b>	Simuloinnissa toiminta kuvataan mallin avulla, johon liitetään toimintaa kuvaavia parametreja ja luodaan matemaattinen malli. Simulointimallin avulla (parametreja muuntamalla) voidaan arvioida toiminnan sujuvuutta eri tilanteissa. Simulointi toteutetaan usein tietokoneella.
<b>SOA</b>	Ks. Palvelupohjainen arkkitehtuuri.
<b>Sovellus</b>	Tietokoneohjelma/-ohjelmisto (engl. application).
<b>Staattinen prosessi</b>	Pysyväluonteinen prosessi; prosessin osat ja suoritusjärjestys määritellyt. Vrt. Dynaaminen prosessi.
<b>Standardi</b>	Tunnustetun osapuolen hyväksymä dokumentti, jossa on määritelty yleistä ja toistuvaa käyttöä varten sääntöjä, ohjeita tai piirteitä tuotteille, prosesseille tai palveluille (Project Management Institute 2000).
<b>Strategia-prosessi</b>	Liiketoiminnan tavoitteellinen kehittäminen tai laadunhallinnallinen jatkuva parantaminen. Kokonaisarkkitehtuurin suunnitteluprosessissa tärkeimpänä syöteenä toimii organisaation johdon asettama strategia (JHS 179).
<b>Strategiset tavoitteet</b>	Organisaation sisäiset, koko organisaation toimintaa ja erityisesti kehittämistä ohjaavat tavoitteet, esim. markkinaosuus, toiminnan volyymin tavoitearvot, uusien palvelualueiden kehittäminen ja organisaatioiden yhdistäminen.
<b>Syöte</b>	Syöte on tietoja ja materiaalia, joka syötetään prosessiin, palveluun tai sovellukseen (Laamanen & Tinnilä 2009). Syöte ei ole sama asia kuin raha, laitteet tai ihmisten osaaminen, jotka ovat resursseja ja siten osa prosessia (JHS 152).
<b>Tarve</b>	Jonkin halutun, tarvitun tai käytännöllisen ominaisuuden/asian puuttuminen. Kuvatut ja määritellyt tarpeet voidaan nähdä vaatimuksina.
<b>Tehtävä</b>	Yksittäisen toimijan määritelty joukko toimenpiteitä halutun tuloksen aikaansaamiseksi. Prosessikuvausten sisällä käytettävä käsite tehtävä (engl. task) tai osatehtävä tarkoittaa käsittelyvaihetta. Nämä tehtävät ovat yleensä yksilön tai ryhmän suorittamia käytännön toimenpiteitä (JHS 152).
<b>Teko</b>	Yksittäisen toimijan lyhytkestoinen toimenpide (operation) halutun tuloksen aikaansaamiseksi.
<b>Teknologia-arkkitehtuuri</b>	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation teknologista infrastruktuuria ja järjestelmäarkkitehtuurin teknologiaavainoita. Teknologia-arkkitehtuurin suunnittelussa linjataan käytettävät tekniset järjestelmien ja ICT-infrastruktuurin ratkaisuvaihtoehdot, standardit ja rakenteet siten, että kokonaisuus tukee parhaalla mahdollisella tavalla organisaation tavoitteita (JHS 179).
<b>Tieto-arkkitehtuuri</b>	Kokonaisarkkitehtuurin näkökulma, joka kuvaa informaation rakenteistamista, organisointia ja luokittelua sekä välitystä. Tarkastellaan organisaation tietotarpeita, tietopääomaa, tietojen välisiä suhteita, informaatioarvoketjua, tietojen rakenteita sekä informaation organisointia ja hallintaa (JHS 171).

<b>Tietojärjestelmä</b>	<p><i>Sosiotekninen määritelmä:</i> Tietojärjestelmä (engl. Information System) käsittää siihen sisältyvän tiedon (data, informaatio, tieto, tietämys), teknologian, työtoiminnan, kommunikaation, organisaation ja ihmiset. Siten tietojärjestelmän perimmäinen tarkoitus on palvella sitä työtä ja sitä työtoimintaa, mihin se on tarkoitettu käytettäväksi palvelun tuottamiseksi (Mursu &amp; Ikävalko 2007).</p> <p><i>Välinekeskeinen määritelmä:</i> Tietojärjestelmän avulla tietoa tallennetaan, säilytetään, välitetään ja hyödynnetään. Voi sisältää sekä manuaalisia, sähköisiä että toiminnallisia elementtejä. Tietoa käsittelevät toimijat (ihmiset ja ohjelmistot) sekä itse tieto (data) ovat osa järjestelmää.</p>
<b>Tieto-järjestelmä-arkkitehtuuri</b>	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation keskeiset järjestelmät sekä niiden arvioidun elinkaaren, kriittisyyden, niiden käyttämät/tuottamat tiedot ja suhteet muihin järjestelmiin. Organisaation järjestelmäpääoma (JHS 171).
<b>Tieto-kokonaisuus</b>	Toisiinsa kiinteästi liittyvä joukko tietoja, esim. asiakirja tai samassa tehtävässä tarvittavien tietoelementtien muodostama kokonaisuus.
<b>Tietovirta</b>	Tiedon liikkumisen suunta ja sisältö kahden tai useamman toimijan tai prosessin vaiheen/toiminnon välillä.
<b>Tietoväline</b>	Väline, jonka avulla tietoa välitetään. Voi olla manuaalinen, sähköinen tai toiminnallinen esim. paperilomake, ilmoitustaulu, ohjelmisto, puhelin tai palaveri.
<b>Toiminnalliset vaatimukset</b>	Määrittelee kehitettävän tai hankittavan järjestelmän käyttäytymistä tai toiminnallisuutta, kuten mitä palveluja ohjelmiston on tarjottava, miten ohjelmisto reagoi syötteisiin ja miten se käyttäytyy annetuissa tilanteissa. Voi olla käyttäjä tai järjestelmävaatimus. Vrt. Ei-toiminnalliset vaatimukset.
<b>Toiminta</b>	Yksittäisen tai kollektiivisen toimijan tavoitteellinen joukko tekoja, joilla pyritään haluttuun lopputulokseen.
<b>Toiminta-arkkitehtuuri</b>	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation strategisiin vaatimuksiin liittyvää ydintoimintaa ja sitä tukevia tukiprosesseja, resursseja sekä palvelutarjontaa (JHS 171).
<b>Toiminta-kokonaisuus</b>	Useiden toimintojen muodostama joukko, joista kullakin toisiinsa liittyvää toimintaa. Joukko tehtäviä, joilla saadaan aikaan tietty tulos.
<b>Toimintatarina</b>	Tyypillisesti yhden toimijan näkökulmasta tehty sanallinen kuvaus tai esimerkki prosessin etenemisestä tai toiminnasta.
<b>Toiminto</b>	Toiminnoksi kutsutaan joukkoa tehtäviä, joiden avulla saadaan aikaan tietty tulos (JHS 152). SOA-kehittämisessä: ohjelmiston tai sovelluspalvelun tarjoama toiminnallinen ominaisuus.
<b>Tapahtuma (event)</b>	Havainnottava lyhytkestoinen sisäinen tai ulkoinen ilmiö, joka voi esim. käynnistää prosessin tai prosessin vaiheen. Esimerkiksi tilauksen saapuminen on heräte tilauksen käsittelylle.
<b>Tukiprosessi</b>	Tukiprosessit avustavat ydinprosesseja ja luovat edellytykset niiden toiminnalle. Tukiprosesseilla voi olla sisäisiä asiakkaita. Tukiprosesseja ovat muun muassa hallinnolliset toiminnot, kuten henkilöstöhallinto sekä osaamisen ja toimintojen kehittäminen (JHS 152), sekä johtamisen, viestinnän, talous-, henkilöstö-, tieto- ja kiinteistöhallintojen sekä materiaalihuollon prosessit.
<b>Operaatio</b>	Ohjelmiston toteuttama toiminta, jonka tarkoituksena on aiheuttaa tietokoneen muistiin, tietokantaan tai tietojärjestelmään tallennettujen tietojen haku sekä mahdollisesti käsittely ja käsittelyn tuloksena olevien tietojen palauttaminen.

Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet

<b>Työnkulku</b>	Manuaaliset ja automatisoidut toimintoketjut, jotka voivat koostua yhden tai useamman toimijan toiminnoista tai teoista (manuaaliset) tai operaatioista (automatisoidut IT-palvelut).
<b>Työnkulku-kaavio</b>	Kaavio, jolla työnkulku kuvataan. Esim. uimaratakaavio.
<b>Työtoiminta</b>	Työtoiminta (work activity) tarkoittaa kokonaisuutta (toimintajärjestelmä), jossa <i>joukko ihmisiä työskentelee järjestäytyneellä tavalla jonkin yhteisen kohteen parissa – ei välttämättä samassa paikassa ja yhtä aikaa – tuottaakseen jonkin yhteisen lopputuloksen</i> (Korpela 1994).
<b>Työtoiminnan analyysi ja kehittäminen</b>	(engl. ActAD: Activity Analysis and Development) Toiminnan teoriaan ja kehittävään työn tutkimukseen perustuva malli, jonka avulla voidaan jäsentää monimutkaista, monen ihmisen yhteistoimintana tapahtuvaa toimintaa (Korpela 1994). Ks. Työtoiminta.
<b>Uimarata</b>	Uimarata on visuaalisessa prosessin kuvaamisessa käytetty tapa ilmaista eri rooleja. Kukin rooli (organisaation tai henkilön) kuvataan omana uimaratanaan (pysty- tai vaakasuoraan), ja tällöin sille sijoitetut prosessiasteleet kuuluvat tämän roolin vastuulle (JHS 152).
<b>Vaatus</b>	Tiettyyn kehittämisen kohteeseen kohdistuva dokumentoitu tarve, jonka toimeenpanosta ja johon liittyvästä ratkaisusta voidaan tehdä päätöksiä.
<b>Viite-arkkitehtuuri</b>	Viitearkkitehtuuri on rajatun arkkitehtuurikonaisuuden abstrakti toimittaja- ja toteutusneutraali rakenne. Se on esitys arkkitehtuurikonaisuuden loogisista osista ja niiden välisistä suhteista. Viitearkkitehtuurilla ohjataan arkkitehtuurisuunnittelua halutunlaiseen toteutusrakenteeseen. Viitearkkitehtuuri voi olla organisaation sisäinen, toimialaan liittyvä tai yleinen looginen rakennemalli (JHS 179).
<b>Vuokaavio</b>	Tietojenkäsittelyn toimintosarjaa kuvaava, toimintoja esittäviä sovittuja symboleja käyttäen laadittu piirros, josta käy selville suoritusjärjestys (JHS 152). Vuokaaviosymbolit on Suomessa standardoitu (SFS-3204).
<b>Ydinprosessi</b>	Organisaation ydinprosessit kehittävät, tuottavat ja toimittavat (ulkopuolisten) asiakkaiden tarvitsemia materiaaleja, tavaroita ja tietoja (tuotteita) (JHS 152).
<b>Yritys-arkkitehtuuri</b>	Ks. Kokonaisarkkitehtuuri

## Sanaston lähteet

- Fowler M. UML Distilled Third Edition. A Brief Guide to the Standard Object Modeling Language. Pearson Education Inc. USA, 2004.
- Gotel O.C.Z, Finkelstein A.C.W. An Analysis of the Requirements Traceability Problem, Proceedings of the 1st International Conference on Requirements Engineering (ICRE 1994), s. 94-101, Colorado Springs, U.S.A., April 18-22, 1994.
- Grant R.M. (1998). Contemporary strategy analysis (3. painos).
- JHS 152. Prosessien kuvaaminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 6.6.2008. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS152/JHS152.pdf>
- JHS 171. ICT-palvelujen kehittäminen: Kehittämiskohteiden tunnistaminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.1. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS171/JHS171.pdf>
- JHS 173. ICT-palvelujen kehittäminen: Vaatimusmäärittely. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.0. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.pdf>
- JHS 179. ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.0. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179/JHS179.pdf>
- Korpela M. Nigerian practice in computer systems development. A multidisciplinary theoretical framework, applied to health informatics. Doctoral dissertation. Helsinki University of Technology. Department of Computer Science. Reports TKO-A31. s. 273, 1994.
- KuntaIT. Kuntasektorin SOA-tekniikkalinjaukset. Versio 1.0, 16.12.2008.
- Laamanen K, Tinnilä M. Prosessijohtamisen käsitteet. 4. painos. Teknologiateollisuuden julkaisuja, 2009.
- Mursu A, Ikävalko P. Tietojärjestelmät palvelemaan työtä - Apuna toimintalähtöinen kehittämismalli. Systemityö 1/2007, s. 9-12, 2007.
- Mykkänen J, Pöyhölä A, Toroi T, Riikonen P, Riekkinen A. Palveluarkkitehtuurin soveltaminen terveydenhuollossa - Osa 1: hyödyt, kustannukset, arviointi ja hankinnat. Kuopio: SerAPI-projekti, Kuopion yliopisto, 2007.
- Project Management Institute 2000. Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute, 2000.
- Simons R. The role of management control systems in creating competitive advantage: New perspectives. Accounting, Organizations and Society. Vol. 15, nro 1/2, s. 127-143, 1990.
- Sprott D. Best Practice Report The Business Case for SOA. CBDI Journal, June, 2006.