

Rakennetun Ympäristön Yläontologia (RYYO)

Loppuraportti ja
hallintamallin kuvaus

Otso Helenius, Ubigu Oy

18.12.2022

KESKENERÄINEN VERSIO



Rakennetun ympäristön
tiedon yhteentoimivuus



Sisällys

1	Yhteenveto	3
1.1	Projektin tausta ja tulokset	3
1.1.1	Mikä on rakennetun ympäristön yläontologia (RYYO)?	3
1.1.2	Mikä on RYYO-määrittelyprojekti?	6
1.1.3	Miksi RYYO on tarpeellinen?	7
1.1.4	Mitkä ovat RYYO:n tavoitteet?	9
1.1.5	Miten voin hyödyntää RYYO:n tuloksia?	10
2	Määrittelyn reunaehdot	12
2.1	Ratkaistavan ongelman määrittely	12
2.1.1	Toiminnalliset vaatimukset	12
2.1.2	Ei-toiminnalliset vaatimukset	12
2.2	Kuvauskielten kartoitus	16
2.2.1	Käytettävissä olevat kielet	16
2.2.2	Kielten välisistä eroista	17
2.2.3	Yhteenveto	23
3	Määrittely	24
3.1	DOLCE	24
3.1.1	Tilanteet ja kuvaukset	26
3.1.2	Suunnitelmat	27
3.1.3	Tietokohteet	27
3.1.4	Kokoelmat	27
3.2	Mallinnusperiaatteet	27
3.3	Lähtöaineiston analyysi	28
3.4	Hallintamalli	30
3.4.1	Sisällön päivittäminen	31
3.4.2	Teknisen rakenteen päivittäminen	31
4	Lähteet	32

1 Yhteenveto

Tässä luvussa kuvataan mahdollisimman selkokielisesti Rakennetun ympäristön yläontologian sisältö, projektin tausta sekä tavoitteet. Tarkempi sisältö, perustelut sekä taustoitukset lainsäädännön, teorian ja teknologian muodostaman kehikon puitteissa löytyvät seuraavasta luvusta. Tiivistelmässä on käytetty käsitteitä niiden kansantajuisissa merkityksissä, ellei muuta määritelmää ole annettu.

1.1 Projektin tausta ja tulokset

1.1.1 Mikä on rakennetun ympäristön yläontologia (RYYO)?

Rakennetun ympäristön yläontologia (myöhemmin RYYO) on kokonaisuus, joka koostuu tietomallista sekä sitä koskevasta hallintamallista (tämä dokumentti). RYYO kuvaa keskeisimpiä rakennetun ympäristön aihepiirin käsitteitä koneluettavina rakenteina. RYYO on tehty edistämään rakennetun ympäristön tietojen yhteentoimivuutta kansallisella tasolla Tiedonhallintalain ja EU:n yhteentoimivuuskehikon puitteissa.

Rakennetun ympäristön yläontologian *tietomalli* kuvaa rakennetun ympäristön aihepiirin keskeiset *tietorakenteet*, sekä niiden väliset *yhteydet*. Nämä tietorakenteet koostuvat

- erilaisista oleellisesti itsenäisiä asioita (usein puhutaan ”kohteista”, ”entiteeteistä” tai ”olioista”) kuvaavista *luokista*,
- niitä oleellisesti yhdistävistä *assosiaatioista* (yhteyksistä toisiin luokkiin) sekä
- niiden olemusta oleellisesti kuvaavista *attribuuteista* (piirteistä).

Tässä painotus on sanalla *oleellisesti*, sillä tavoitteena on pyrkiä määrittämään ne yhteydet ja piirteet, joilla jokin asia voidaan yksiselitteisesti tunnistaa kuuluvan luokkaan X (tai toisin ilmaistuna ”olevan tyyppiltään X”). Tietorakenteiden välisillä yhteyksillä taas tarkoitetaan niitä sääntöjä, joilla tietorakenteet on yhdistetty toisiinsa.

Sanalla *ontologia* on monia vaihtelevia merkityksiä eri aloilla, mutta tämän dokumentin puitteissa käytetään tieto- ja informaatiotieteiden määritelmää: ontologia on *formaali* malli jonkin aihealueen tietojen kuvailuun käytettävästä *käsitteistöstä*. Käsitteistöllä tarkoitetaan yllä kuvattujen tietorakenteiden muodostamaa kokonaisuutta, jolla asioita voidaan kuvailla. Formaaliilla taas tarkoitetaan sitä, että tämä käsitteistö perustuu matemaattiseen logiikkaan, mikä mahdollistaa sen varassa kuvatun tiedon koneymmärryksen sekä konepäätelyn.

Ontologioilla onkin vahvat juuret juuri tekoälytutkimuksessa ja -kehityksessä. Niiden keskeisiä sovelluskohteita ovat alusta asti olleet ihmisasiantuntijoiden hyödyntämän ja tuottaman monimuotoisen tiedon koneluettava kuvailu (engl. knowledge representation), uuden koneluettavan sisällön ”uuttaminen” tiedosta (engl. knowledge acquisition), sekä koneluettavan vastineen pohjalta tapahtuva konepäätely (engl. knowledge-based reasoning) (Hoekstra, 2009).

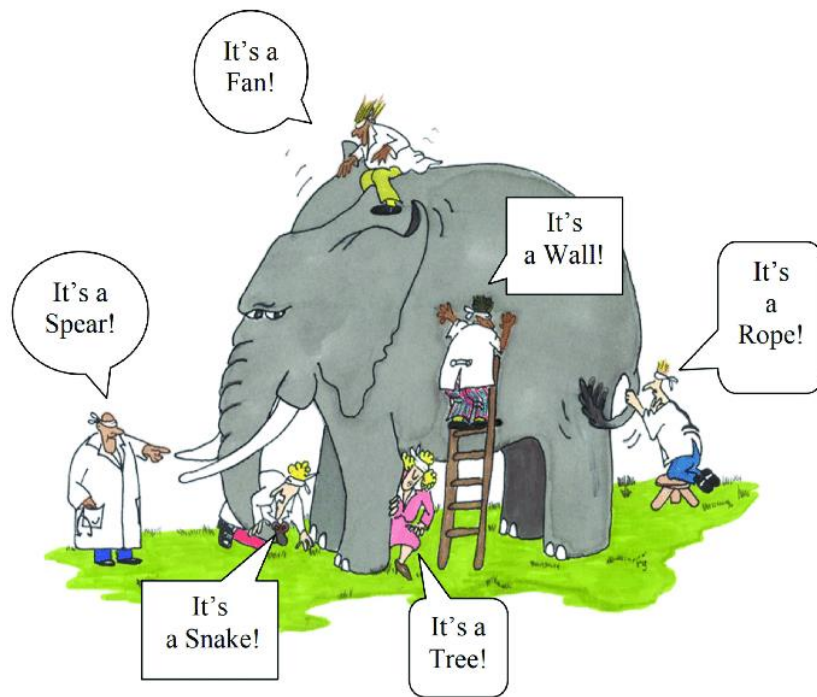
Ontologioiden toinen keskeinen käyttökohde ovat nk. linkitetyn datan sovellukset, joissa merkittävänä tavoitteena yllä kuvatun käytön lisäksi on lukuisten itsenäisten toimijoiden sekä heterogeenisten tietoaaineistojen integrointi ja yhteentoimiva hyödyntäminen.¹

Ontologioita on eri tasoisia riippuen siitä, millä tarkkuudella ja millaista tietoaletta ne kuvaavat. Tässä projektissa määrittelyn kohteena on ns. yläontologia (engl. upper ontology): siinä määritellyt tietorakenteet ovat nimensä mukaisesti *ylätasoisia* eli mahdollisimman yleispäteviä. Yläontologian varaan voidaan määrittää tarkempia ja rajatumpia aihepiiri- tai sovelluskohdeontologioita (engl. domain ontology). Tällaisia rakennetun ympäristön piirissä ovat esimerkiksi rakennusluvitukseen, asemakaavoitukseen tai infrarakentamiseen liittyvät määrittelyt.

RYYO:n rooli yläontologiana tarkoittaa vakaan perustan tarjoamista, jolle rakennetun ympäristön kokonaisarkkitehtuurin aihepiiriin kuuluvat määrittelyt (prosessit, roolit, tapahtumat, tehtävät, tilat, vastuut, oikeudet, tavoitteet, yhteydet, tietorakenteet, sisällöt, jne.) voidaan määrittää. On siis tärkeää ymmärtää, että kyseessä ei ole vain ”perinteisten” tietomallien sisältöön liittyvä määrittely. RYYO toimii eräänlaisena temaattisena sateenvarjona, joka on *peittävä* eli määritelty siten, että kaikki tarkemmat määrittelyt – kuten esimerkiksi rakentamisluvituksen tai kaavoituksen tietomallit, prosessimäärittelyt tai koodistot – on sijoitettavissa sen tarjoamien ylärakenteiden varaan.

Lappalaista (2014) lainaten yläontologian tarkoituksena on ”määrittää valittujen tiedonalojen yleisimmät käsiteläokat ja täten mahdollistaa osaltaan semanttinen yhteentoimivuus eri tietoresurssien välillä”, ja että ”informaatiotieteen ontologiat eivät yleensä pyri luokitteluamaan kaikkea olevaa tai maailmaa yleensä, vaan tietyn rajatun aihealueen (engl. domain) yhteisiä käsitteistöjä, ja näin mahdollistamaan tietojen vaihdon eri resurssien ja järjestelmien välillä. Lisäksi näissä ontologioissa ei oteta kantaa siihen, onko luokiteltuja asioita ’todellisuudessa’ olemassa; jos luokittelu on toimiva tiedonjakamisen kannalta, se täyttää tehtävänsä”.

¹ <https://www.w3.org/standards/semanticweb/ontology>



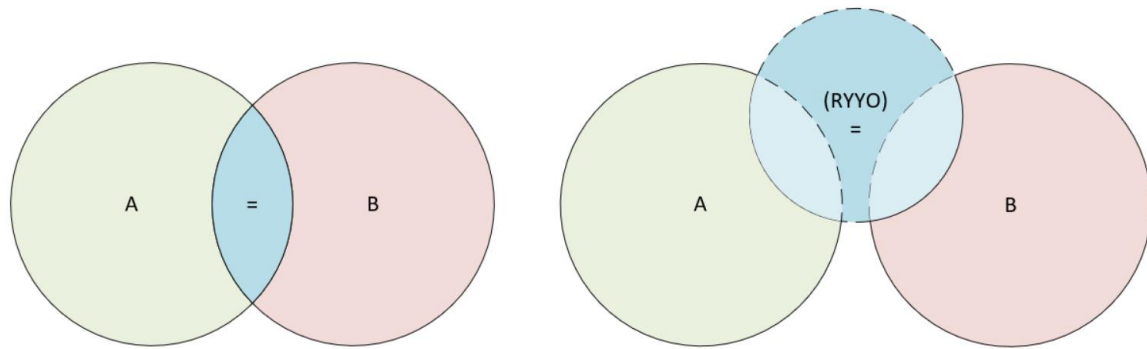
Kuva 1. Samasta asiasta voidaan antaa tarkastelun perspektiivistä riippuen hyvin ristiriitaisia kuvauksia (G. Renee Guzlas, <https://doi.org/10.1046/j.1523-1755.2002.00600.x>).

Tietorakenteet on toteutettu siten, että niiden mukaisena tuotettu tai niillä täydennetty tieto on *semanttisesti yhteentoimivaa*, eli että se on yksikäsitteisesti ymmärrettävää niin ihmisille kuin koneillekin. Semanttisen yhteentoimivuuden mahdollistaminen tietojen määrittelyssä, kuvailussa, käsittelyssä ja välittämisessä on koko yläontologian keskeisin tavoite.

RYYO on kehitetty tiedostaen, että nykyisin käytössä olevia käsitteistöjä ja rakenteita ei voida pyyhkiä pois ja aloittaa määrittelyjä täysin puhtaalta pöydältä. Tämän vuoksi RYYO ei myöskään ole *puhtaan normatiivinen*: se ei siis pakota toimijoita sitoutumaan auktoriteetinomaisesti ”vain yhteen totuuteen” tietyn käsitteen määrittelyssä, sillä tällaista neutraalia näkökulmaa ei varsinaisesti edes ole olemassa (ts. monissa tapauksissa Kuva 1:ssä esitettyä yksikäsitteistä entiteettiä ei ole olemassa, on vain kokoelma osittain toistensa kanssa yhteneviä kuvauksia).

Sen sijaan RYYO on pehmeän normatiivinen: se ohjaa ottamaan käyttöön parhaisiin käytänteisiin ja tutkimustietoon pohjautuvia menetelmiä, joiden pitkäjänteisen ja johdonmukaisen soveltamisen myötä tuotetut määritelmät ohjautuvat kohti mahdollisimman yksikäsitteistä ja yleispätevää ”yhtä totuutta”.

Keskeinen RYYO:n tavoite on rakennetun ympäristön tietojen semanttisen yhteentoimivuuden muodostamisen ja sen ylläpidon tukeminen sekä vahvistaminen. Siinä missä uutta määriteltäessä RYYO:n normatiivinen ohjaus on vahvempaa, olemassa olevan tiedon kohdalla RYYO:n soveltamisessa painopiste taas on vahvemmin sen käyttämisessä *semanttisena liimana* olemassa olevien tietojen yhdistämiseen keskenään. Tietojen yhdistämisen myötä syntyy tilanne, jossa niitä voidaan ryhtyä hallitusti harmonisoimaan yhteentoimivuusmenetelmän avulla.



Kuva 2. Vasemmalla esitettynä tilanne, jossa tiedetään kahden tietueen kuvaavan samaa kohdetta molemmista löytyvän saman yksilöivän tunnisteiden perusteella (sininen leikkaus), oikealla tilanne, jossa sama yhtäläisyys voidaan päätellä ulkoisen rakenteen kautta, vaikka tietueet eivät jaa yhteisiä tunnisteita.

Yksinkertaisimmillaan aiemmin erillisten tietojen yhdistäminen voi tarkoittaa esimerkiksi sitä, että kaksi keskenään ristiriitaisilla tietomalleilla rakennuksia kuvaavaa tietovarantoa voivat automaattisesti tunnistaa puhuvansa samasta asiasta (rakennus) – hienostuneemmalla tasolla taas esimerkiksi sitä, että tietorakenteiden eroista huolimatta on mahdollista konepäätelyn perusteella johtaa (engl. infer) aukottomasti niiden puhuvan tietyssä tilanteessa yhdestä ja samasta rakennuksesta silloinkin kun tietueista ei löydy yhteisiä yksilöiviä tunnisteita (esim. VTJ-PRT) tai muita suoraan identifioinnin mahdollistavia tietoja (Kuva 2).

Edeltävä kappale voidaan kokonaisuudessaan tiivistää yhdeksi selkokieliseksi määritelmäksi:

RYYO on tietomalli, joka mahdollistaa rakennetun ympäristön piiriin kuuluvista asioista ”puhumisen” tarkalla, yksiselitteisellä ja yhdenmukaisella tavalla niin ihmisten kuin koneiden kesken.

Tätä kuvausta ei kuitenkaan tule tulkita siten, että RYYO:n määrittely koskee vain tietojärjestelmiä, tietomalleja, tai olisi muutoin luonteeltaan suljettu. Tämän dokumentin tavoitteena on kuvata, kuinka RYYO kytkeytyy verkostomaisesti kaikkiin rakennetun ympäristön ekosysteemin osa-alueisiin. Lisäksi dokumentissa muistutetaan, että rakennetun ympäristön käsittely *verkostona* on kriittisen tärkeää kaikissa määrittelyhankkeissa, sillä nykyiset ratkaistavat haasteet ovat pitkälti seurausta tähän asti vallinneista rakenteellisista (esim. tietyn tietojärjestelmän näkökulma), toiminnallisista (esim. tietyn kunnan rakennusluvituksen prosessinäkökulma) tai rooliperusteisista (esim. tietyn viranomaisen näkökulma) siiloista.

1.1.2 Mikä on RYYO-määrittelyprojekti?

RYYO on tuotettu toukokuun ja joulukuun 2022 välillä Ympäristöministeriön tilaamassa projektissa, jossa hankinnan kohteena oli asiantuntijatyö RYTJ-määrittelytyön aikana tehdyn yläontologian alustavan luonnoksen päivittämiseksi. Projekti on toteutettu osana RYHTI-hankkeen rakennetun ympäristön yhteentoimivuustyötä, ja sen tuloksia hyödynnetään jatkossa niin RYTJ:n toteutusvaiheessa kuin yhteentoimivuustyön osana.

Määrittelyn on tuottanut konsulttikonsortio, jonka työryhmään kuuluvat:

- Otso Helenius (projektipäällikkö, erityisasiantuntija), Ubigu Oy

- Kaisu Laitinen (erityisasiantuntija), Ramboll Finland Oy
- Juha Liukas (erityisasiantuntija), Sitowise Oy
- Johanna Jalas (erityisasiantuntija), Ramboll Finland Oy
- Anu Soukki (erityisasiantuntija), Sitowise Oy

Projektin ohjausryhmään kuuluvat:

- Anssi Hänninen (erityisasiantuntija), YM
- Katri Seppälä (johtaja), Sanastokeskus Ry
- Teemu Pekkanen (RYTJ-rakennustietovarannon tuoteomistaja), SYKE
- Seija Lonka (RYTJ-kaavatietovarannon tuoteomistaja), SYKE
- Jari Vaarma (erityisasiantuntija, mukana projektin ensimmäisellä puoliskolla), YM

Projektin lopputuotteisiin kuuluvat tämä loppuraportti, koneluettava ontologia RDF/XML-tiedostona sekä siitä automaattisesti tuotettu dokumentaatio. Lopputuotteet on julkaistu

- Yhteentoimivuusalustalla (<https://tietomallit.suomi.fi/model/ryyo/>), sekä
- SYKE:n tietomallisivustolla (<https://tietomallit.ymparisto.fi/ry-ylaontologia/>).

1.1.3 Miksi RYYO on tarpeellinen?

Rakennetun ympäristön piiriin kuuluvat tiedot (lainsäädäntö, tietomallit, koodistot, sanastot, suunnitelmat, päätökset, arkkitehtuurikuvaukset, prosessit, toimijoiden roolit jne.) ovat kokonaisuus, jonka osat ovat

- syntyneet eri aikoina erilaisiin toimintaympäristöihin,
- tuotettu erilaisiin tarpeisiin pohjautuen eri lähtökohtiin,
- tarkkuudeltaan, ajantasaisuudeltaan sekä rakenteeltaan vaihtelevia,
- eri toimijoiden määrittelyvallan sekä ylläpitovastuun alaisia,
- riippuvaisia tulkinnasta mm. oikeuskäytänteiden tai soveltamisen kautta, ja
- eri asteisesti säänneltyjä.

Analysoimatta kokonaisuutta tai sen muodostumiseen johtaneita syitä tässä syvemmin, voidaan todeta rakennetun ympäristön piirissä harjoitettavan toiminnan kärsivän nykyisellään ainakin alempana kuvatuista haasteista, joista merkittävä osa on tunnistettu mm. Tulevaisuuden maankäyttöpäätökset -hankkeessa (Aarnio ym., 2020). Seuraavassa käydään läpi merkittävimmät ratkaisua kaipaavat ongelma-alueet sekä rakennetun ympäristön tietopohjaan kohdistuvat vaateet tarkemmin.

1.1.3.1 Tietosisältöihin liittyvät haasteet

Alla kuvatut nykyiset haasteet kasvattavat rakennetun ympäristön tietojen tulkintaan käytettävää resurssitarvetta ja lisäävät monitulkintaisuuden riskiä:

- Valtaosassa tietoja (asiakirjat, tietomallit, koodistot jne.) ei ole selvillä, ovatko käytettyjen termien määritelmät (merkitys) peräisin tietystä terminologisesta sanastosta, säädöksestä, alan vakiintuneesta jargonista tai toisaalla tiettyyn käyttöön tehdystä paikallisesta (*ad-hoc*) määrittelystä.
- Säädösten sisältö ei läpinäkyvällä tavalla perustu kattaviin ja validoituihin terminologisiin sanastoihin, jotta niitä voitaisiin helposti hyödyntää määrittelyn pohjana.

- Osa edelleen käytössä olevista sanastoista ja luokitteluista ei ole yhteentoimivassa ja koneluettavassa muodossa, eikä niitä hyödyntäviä aineistoja täten ole voitu liittää kiinteästi sanastoihin metadatatason tasolla.
- Määrittelyjen pohjana mahdollisesti käytettävät standardit vaativat sovellettaessa tulkintaa, niissä hyödynnettyjä notaatioita ei tyypillisesti ole nimetty, ja sovellettaessa yhteen eri aihepiirien standardeja ei soveltaajalla tyypillisesti ole tietoa siitä, ovatko standardit keskenään harmonisoituja.
- Monissa järjestelmä- ja mallitoteutukset kärsivät nk. heikosta semantiikasta (engl. weak semantics), toisin sanoen käsiteltävän sisällön merkitys joudutaan päättämään sen sijainnista osana tietorakennetta, tietuetyypin nimeämiseen käytetystä teknisestä nimestä, tai huonoimmassa tapauksessa itse tietosisällöstä ja sen syntaksista.

1.1.3.2 Tietorakenteisiin liittyvät haasteet

Alla kuvatut haasteet ylläpitävät rakennetun ympäristön tietojen siiloutumista järjestelmiin, ja hankaloittavat tietojen ongelmatonta käsittelyä niiden välillä:

- Tietorakenteiden kuvausta varten tehtävissä käsitteellisen tason tietomalleissa ei tyypillisesti ole läpinäkyvää viitettä pohjana käytettyyn käsitteistöön, eikä tietoa, perustuuko käsitteistö johdonmukaiseen kokonaisuuteen (esim. terminologinen sanasto).
- Loogisen tason tietomalleissa ei tyypillisesti ole kuvattu niitä perusteita, joilla käsittemalli on tulkittu loogiseksi rakenteiksi, hankaloittaen mallien välisen vastaavuuden validointia.
- Loogisen tason tietomalleissa ja taksonomioissa (kuten koodistoissa) on käytössä lukuisia toisistaan eroavia esitystapoja samaa tarkoittavista käsitteistä, tai käsitteiden välinen hierarkia on esitystapojen välillä ristiriitainen (engl. semantic type inconsistency).
- Loogisen ja käsitteellisen tason malleista löytyy toisinaan rakenteita, jotka ovat osa valitun mallinnuskielen tai toteutustason ohjelmiston, tietokannan, rajapinnan tai serialisointitavan erityispiirteitä, mutta jotka eivät kuulu osaksi kuvattavaa mallin sisältöä loogisella ja käsitteellisellä tasolla.
- Arkkitehtuuri- ja prosessimalleja tehdään vaihtelevin tavoin (pääosin ei koneluettavassa muodossa), jolloin niiden johdonmukainen kuvailu käsitteistöllä on virheeltistä ja rakenteiden vertailu manuaalista ja työlästä.
- Valtakunnallisesti käytettäviä standardeja ei ole määritelty riittävän täsmällisesti.

1.1.3.3 Suhde toimijoiden odotuksiin ja kansallisesti määriteltyihin tavoitteisiin

Samanaikaisesti rakennetun ympäristön kokonaisuuteen kohdistuu alla kuvattuja vaateita (Aarnio ym., 2020; Frisky & Anjoy & Netum Oy, 2022; Oinonen ym., 2017; Riekkinen, 2022; Rissanen & Lahti, 2022; Valtioneuvosto, 2022):

- Yhteentoimivan tiedon tulee olla esteettömästi käyttäjien saatavissa koko yhteiskunnassa.
- Yhteentoimiva tieto virtaa oikeansuuntaisin tietovirroin, hyvin organisoitujen tietovarantojen välillä.
- Hallittu tietorakenne, yhdenmukaiset käytännöt ja kansalliset palvelut vähentävät päällekkäistä työtä ja kustannuksia.
- Tieto on rakenteistettu ja kuvattu kansallisten laatuksien mukaisesti.
- Laadukas tieto auttaa tekemään parempia päätöksiä sekä edistää liiketoimintaa ja palveluja.

- Tiedon määrityksissä otetaan huomioon myös muu kuin alkuperäinen käyttö (niin sanotut toissijaiset käyttötarkoitukset).

1.1.3.4 Suhde kansallisen ja ylikansallisen yhteentoimivuuden viitekehykseen

Rakennetun ympäristön kokonaisuuteen kohdistuu myös alla kuvatun viitekehyksen mukaisia vaateita ja tavoitteita:

- Tietojärjestelmien ja tietovarantojen yhteentoimivuutta tulee edistää ja viranomaisten tietoaaineistojen yhdenmukainen ja laadukas hallinta varmistaa siten, että tietojen merkitys ja käytettävyys säilyvät tietoja hyödynnettäessä ja vaihdettaessa eri tietojärjestelmien välillä (Laki julkisen hallinnon tiedonhallinnasta, 2019).
- Julkisen hallinnon elimien tulisi asettaa asiakirjojaan saataville uudelleenkäyttöä varten siten, että asiakirjat sekä niitä kuvaavat metatiedot ovat saatavilla avoimessa muodossa, joka voi olla koneellisesti luettava ja jolla varmistetaan yhteentoimivuus, uudelleenkäyttö ja tietojen saatavuus (Euroopan parlamentin ja neuvoston direktiivi 2019/1024, 2019).
- Eurooppalaisen yhteentoimivuuskehikon (EIF) puitteissa on linjattu suosituksista, joiden tarkoitus on auttaa julkishallintoja hallinnoimaan niiden omistamia tietoja yhteisissä semanttisissa ja syntaktisissa muodoissa, jotta niitä olisi helpompi julkaista portaaleissa ja koota yhteen, jakaa ja käyttää uudelleen (Euroopan komissio, 2017).

On selvää, että edellä kuvattujen haasteiden puitteissa tavoitteiden toteutuminen tehokkaalla tavalla ei nykyisellään ole mahdollista.

1.1.3.5 Suhde rakennetun ympäristön kokonaisarkkitehtuuriin

RYYO:n määrittelyprojektin aikana on lokakuussa 2022 käynnistynyt myös rakennetun ympäristön kokonaisarkkitehtuuriin (KA) konsulttityö, jonka tehtävänantona on ”rakennetun ympäristön ydintietoihin liittyvän toiminnan, prosessien, tietojen, tietojärjestelmien sekä niiden tuottamien palveluiden rakenteen määrittely” (Kokonaisarkkitehtuuriin teemaryhmän kokouksessa nro. 6/2022 esitelty työsuunnitelma).

Kyseinen arkkitehtuurityö hyödyntää ja täydentää RYYO:ssa määriteltyä pohjaa. Toimiva kokonaisarkkitehtuuri on määritelmällisesti (viitaten etuliitteeseen ”kokonais-”) riippuvainen yhteisestä ja ristiriidattomasta käsitteistöstä, sillä muutoin arkkitehtuurin osien välistä johdonmukaisuutta on mahdotonta arvioida laadukkaasti. Johdonmukaisuuden arvioinnissa keskeistä on järjestelmäsuunnittelun (engl. systems engineering) kieltä lainaten tehtyjen määrittelyjen *validointi* (”Olemmeko määritelleet oikean asian?”) sekä *verifiointi* (”Olemmeko määritelleet sen oikein?”) (Cloutier & Hoboken, 2022).

1.1.4 Mitkä ovat RYYO:n tavoitteet?

RYYO:n ensisijaisena tavoitteena RYHTI-hankkeen projektin puitteissa on tarjouspyynnön perusteella

- 1) tarjota pohja, ”jota tullaan hyödyntämään tulevien tietomallien laatimisessa, rakennetun ympäristön tietokomponenttikirjaston koostamisessa sekä tietomallien harmonisoinnissa”,
- 2) toimia ”keskeisenä pohja-aineistona rakennetun ympäristön yhteentoimivuustyössä, sekä tietomallien ja tietokomponenttien keskinäisten kytkösten hahmottamisessa”, sekä

- 3) tehdä ehdotus siitä, miten RYYO:n ” sekä Yhteentoimivuusalustan Rakennetun ympäristön tietokomponenttikirjaston keskinäinen hallinta ja päivittäminen kannattaisi järjestää”.

RYYO:n toissijaisena pidemmän tähtäimen tavoitteena on

- 1) edistää Tiedonhallintalain sekä sen taustalla vaikuttavien EU-direktiivien sekä Eurooppalaisen yhteentoimivuuskehikon (EIF) mukaisen semanttisen yhteentoimivuuden kansallista toteutumista,
- 2) lisätä sidosryhmien tietoutta semanttisen yhteentoimivuuden merkityksestä erityisesti julkishallinnon tiedonhallinnan sekä rakennetun ympäristön laajemman toimijaverkoston kontekstissa,
- 3) esitellä sidosryhmille ajantasaiseen tutkimustietoon, standardeihin sekä parhaisiin käytänteisiin perustuvia periaatteita ja menetelmiä yhteentoimivuuden toteuttamiseen tietomallintamisessa, ja
- 4) täsmentää sekä selkeyttää tietomallintamisesta alalla käytävää keskustelua ja käsitteistöä.

Keskeisenä työkaluna RYYO:n tavoitteista ja menetelmistä viestimiseen toimii tämä loppuraportti. Aiheen laajuuden ja teknisen luonteen vuoksi sen sanallistaminen mahdollisimman ymmärrettävästi on haastavaa, ja dokumentti ei välttämättä ole ilmaisultaan riittävän selkeä koko potentiaaliselle lukijakunnalle. Raportin sisältöä koskevia kysymyksiä ja kommentteja otetaan mielellään vastaan esimerkiksi sähköpostitse (otso@ubigu.fi) tai Yhteentoimivuusmenetelmän Slack-työtilassa (<https://yhteentoimivuus.slack.com/>).

1.1.5 Miten voin hyödyntää RYYO:n tuloksia?

RYYO-määrittely koneluettavana ontologiana ei ole tämän projektin päättyessä suoraan tuotantokäyttöön kelpaava, sillä projektin resursseilla ja aikataululla sellaisen tuottaminen ei olisi realistinen tavoite. Tämän projektin lopputuotteet muodostavat pohjan pitkäjänteiselle ja *jatkuvalle* rakennetun ympäristön ontologiatyölle, joka mahdollistaa korkeatasoisten tietomäärittelyjen ja yhteentoimivuuden tason saavuttamisen sekä sen ylläpitämisen.

On tärkeää tiedostaa, että edes tuotantovalmiiseen versioon RYYO:n sisällöstä ei tulisi suhtautua staattisena (muuttumattomana) tuotoksena, joka voidaan ottaa käyttöön ilman muutostenhallintaa. Aihepiirin laajan rajauksen vuoksi RYYO:n soveltamisalue on niin laaja ja monimutkainen, että kaikkia käyttötapauksia ja käytössä ilmeneviä rajatapauksia ei voida kartoittaa etukäteen perinteisen vesiputousmallin tapaan. Keskeistä on osallistua yhteentoimivuustyön esimerkiksi Yhteentoimivuuden teemaryhmien kautta.

Sen sijaan, RYYO:n määrittelyssä käytettyjä yhteentoimivuutta edistäviä menetelmiä ja teknologioita voidaan ottaa suoraan käyttöön, sillä tähän loppuraporttiin on valikoitu sellaiset, jotka ovat kypsyyssasteeltaan, lisensseiltään ja teknologioiltaan mahdollisimman vakaita, saavutettavia ja avoimia.

Sanastojen, koodistojen ja muiden tietomallien tavoin RYYO tarvitsee määrittelyissä tavoitteissaan onnistuakseen yhteentoimivuusmenetelmän kaltaisen menetelmän hyödyntämistä: rakennetun ympäristön toimijoiden soveltaessa mallia käytäntöön nousee esiin muutostarpeita, jotka käsitellään ja joiden kautta syntyvät muutokset voidaan viedä tuotantoon hallitusti. Tiivistettynä: koska itse toimintaympäristö on jatkuvassa muutoksessa, malleja pitää pystyä päivittämään sekä vastaamaan muutoksiin että ohjaamaan muutosta normatiivisesti oikeaan suuntaan.

Nykyisessä muodossaan RYYO:n tulosten soveltamisesta on eniten hyötyä organisaatioiden käsitteistön sekä ydintietomäärittelyjen validoinnissa tietomallinnuksen, sanastotyön ja tiedohallinnan kehittämisen yhteydessä, sekä semanttisen yhteentoimivuuden käytänteiden jalkauttamisessa. Tarkempaa taustoitusta ja kuvausta RYYO:n käyttöönotosta teknisellä tasolla on seuraavissa kappaleissa.

2 Määrittelyn reunaehdot

RYYO:n määrittelytyö koostuu seuraavista vaiheista:

- 1) On kuvattu, mitä RYYO-määrittelyssä tulee ratkaista ja missä puitteissa, esitetty määrittelyä koskevien vaatimusten muodossa (englanniksi nk. problem space).
- 2) On kuvattu, millä mahdollisilla tavoilla määrittely voidaan ratkaista (englanniksi nk. solution space), ja valittu tunnistetuista tavoista toteutuskelpoisin.
- 3) On analysoitu projektin lähtöaineistot, ja valittua toteutustapaa noudattaen tehty määrittelyn laajuuden rajausta sekä tuotettu sisältö.

2.1 Ratkaistavan ongelman määrittely

Jotta RYYO:n lopullinen rakenne vastaa kappaleessa 1.1.4 määritelyihin tavoitteisiin eikä toista kappaleessa 1.1.2 kuvattuja ongelmia, täytyy määrittelyn perustua laajalti validoituun ja teoreettisesti kestävään vaatimusmäärittelyn kehikkoon. Kappaleen 1.1.4 tavoitteet on tulkittu RYYO:a koskeviksi vaatimuksiksi (engl. requirements) jotka on jaettu järjestelmäsuunnittelun periaatteiden (Adams, 2015; Dick ym., 2017) mukaan toiminnallisiin (engl. Functional Requirements, FRs) ja ei-toiminnallisiin (engl. Non-Functional Requirements, NFRs).

2.1.1 Toiminnalliset vaatimukset

RYYO:lla itsellään ei perinteisessä mielessä ole varsinaista ”toiminnallisuutta”, mutta sillä tulee olla ominaisuuksia, jotka antavat sitä hyödyntäville toimijoille kyvykkyudet saavuttaa tai edistää kappaleen 1.1.4 tavoitteita. Toiminnalliset vaatimukset on tässä ilmaistu käyttäjätarinoiden muodossa, minkä vuoksi ne ovat ei-toiminnallisia merkittävästi lyhyempiä. Käyttäjätarinat on kirjoitettu rakennetun ympäristön yhteentoimivuustyötä koordinoivan toimijan perspektiivistä.

- FR1** Haluan, että kaikkien rakennetun ympäristön määrittelyiden pohjana ovat validoitujen ja verifioitujen koneluettavien sanastojen käsitteet, joihin määrittelyt linkitetään koneluettavasti.
- FR2** Haluan, että kaikki rakennetun ympäristön yhteiset (tietokomponenttitasoiset) tietorakenteet on määritelty formaalisti FR1 mukaisesti.
- FR3** Haluan, että kaikkien rakennetun ympäristön koodistojen koodit on määritelty FR1 mukaisesti.
- FR4** Haluan, että kaikki (soveltamisprofiili)tietomallien rakenteet on johdettu FR2 ja FR3 mukaisista määritelmistä ja niitä tarkennettaessa määritelty FR1 mukaisesti.
- FR5** Haluan, että kaikki jo tuotetut (soveltamisprofiili)tietomallit voidaan harmonisoida FR4 mukaisiksi.

2.1.2 Ei-toiminnalliset vaatimukset

Seuraavassa on kuvattu RYYO:a koskevat ei-toiminnalliset vaatimukset, jotka kertovat puitteet, joissa toiminnallisten vaatimusten tulee täyttyä. Pääosa vaatimuskehikosta pohjautuu ontologiamäärittelyä (engl. ontology engineering) koskevaan kirjallisuuteen (Colomb, 2007; Gangemi, Catenacci, ym., 2004; Gómez-Pérez ym., 2004; Gruber, 1995; Jackson, 2009; Object

Management Group, 2014a; Uschold & Gruninger, 1996; Wang ym., 2008). Vaatimusten pakolliset ehdot on määritelty imperatiivissa (tulee), suositeltavat mutta ei pakolliset konditionaalissa (tulisi).

NFR1 (selkeys)

Selkeyttä (engl. clarity) koskevan vaatimuksen mukaan

- RYYO:n rakenteiden tulee olla merkitykseltään yksiselitteisiä.
- RYYO:n rakenteissa tulisi käyttää kattavia määritelmiä (engl. complete, loogisesti välttämätön ja riittävä) osittaisten (engl. partial, loogisesti välttämätön tai riittävä) sijaan.²
- RYYO:n rakenteiden määritelmien tulee olla saatavilla luonnollisella kielellä kuvattuina.
- RYYO tulee olla piilo-oletuksista vapaa: kaikkien määritelmien ja oletusten tulee olla läpinäkyvästi esitettynä.
- RYYO:n rakenteiden määritelmien tulisi olla kuvattu formaalien aksioomien varaan.

Vaatimuksen ensimmäiseen osaan liittyvät mm. Ryhti-hankesuunnitelmassa ja Tulevaisuuden maankäyttöpäätökset -selvityksessä kuvatut vaatteet tietojen, suunnittelun ja lain ymmärrettävyydestä sekä tiedolla johtamisesta (Aarnio ym., 2020; Rautiainen, 2021).

Esimerkki: ontologian tulee yksikäsitteisesti kuvata, millä ehdoin ja minkä tyyppisiä olioita se määrittelee rakennuksiksi.

NFR2 (johdonmukaisuus)

Johdonmukaisuutta (engl. coherence, groundedness of primitives) koskevan vaatimuksen mukaan

- RYYO:n määrittelyn tulee olla perusteiltaan (esim. aksioomat) sisäisesti ristiriidaton.
- RYYO:n rakenteiden määritelmien tulee pohjautua yhtenäisiin ja selkeästi kirjattuihin mallintamiskäytänteisiin (engl. ontology design patterns), joista käy ilmi mallintamisen motiivi ja käyttökohde.
- RYYO:n määrittelyn johdonmukaisuuden tulee ulottua myös RYYO:n luonnollisella kielellä kuvattuihin osiin.

Esimerkki: ontologia ei saa samanaikaisesti määritellä rakennusta vain ja ainoastaan ”tilalliseksi rakennelmaksi” (Mäkelä ym., 2018) eli rakennelman tarkemmin rajatuksi alaluokaksi, ja muilla määritelmillä (esim. rakennusluvituksen näkökulmaan pohjautuen) ilmaista, että rakennelma on vain ja ainoastaan täysin rakennuksesta erillinen (ts. rakennuksen ja rakennelman luokat eivät ole leikkaavia).

2 Jos lause p on lauseen q välttämätön ehto, q ei voi olla tosi, ellei p ole tosi. Esimerkki: jos p on ”x sisältää suljettavaa tilaa” ja q on ”x on rakennus”, x:n on sisällettävä suljettavia tiloja, jotta se voidaan luokitella rakennukseksi. Kuitenkaan se, että x sisältää suljettavaa tilaa ei kuitenkaan takaa sen olevan rakennus (se voi olla rahtikontti, laiva, kallioluola, jne.).

Jos taas lause p on lauseen q riittävä ehto, q on tosi, jos p on tosi (mutta q voi olla tosi myös, kun p ei ole tosi). Esimerkki: jos p on ”x sisältää huoneita” ja q on ”x on rakennus”, p:stä seuraa automaattisesti, että x:n on oltava rakennus. Kaikki rakennukset eivät kuitenkaan sisällä huoneita, joten x voi olla rakennus vaikkei se sisältäisi yhtäkään huonetta.

Kun tehdään jonkin tietyn entiteetin määrittelyä, molempien ehtojen tulisi täyttyä, jotta määritelmän rajaus kohdistuu täsmällisesti.

NFR3 (kuvaustapavinouma ja vastuiden erottelu)

Kuvaustapavinoumaa (engl. encoding bias) ja vastuiden erottelua (engl. separation of concerns) koskevan vaatimuksen mukaan

- RYYO:n tulee olla määritelty tiedollisella tasolla (engl. knowledge level), ts. määrittely ei saa olla alisteinen teknisten tiedon kuvaustapojen, rajapinta-, protokolla-, tai järjestelmäratkaisujen määräämälle muodolle.

Esimerkki: rakennuksen käyttöönottovuosi tulisi määritellä vain yleisesti kokonaislukuna ja pidättäytyä määrittämästä, että käyttöönottovuosi on tietotyyppiltään esimerkiksi tietylle arvovälille rajautuva uint tai leksikaalisesti merkkijonoksi eikä kokonaisluvuksi kuvautuva `xsd:gYear` (kyseinen tyyppi voi sisältää ISO 8601 mukaisesti aikavyöhyketietoa, esim. "2022+02:00"). Varsinainen semanttinen määritelmä (arvon tulkinta tietyssä kontekstissa) seuraa käyttöönottovuoden määritelmästä, ei itse lukuarvon tietotyyppistä tai leksikaalisesta esitysmuodosta. Käytetty kalenterijärjestelmä ilmaistaan erikoistamalla tai täydentämällä ontologian käyttöönottovuoden määritelmää itse käyttökontekstissa (esim. tietty tietovaranto).

Esimerkki: rakennuksen omistajaa ei tule määritellä konkretisoinnin (reifikaation) kautta omana rakennuksina ja henkilöitä yhdistävänä (m:n) luokkana vain siksi, että tämä on tyypillinen tapa ratkaista asia relaatiomalleissa. Loogisen mallin tasolla rakennuksen ja henkilön välinen omistussuhde on nimensä mukaisesti entiteettien välinen suhde ja myös mallinnetaan sellaisena. Varsinainen omistajuus reifioituna roolia kuvaavana luokkana määritellään erikseen osajoukkona niistä henkilöistä, joilla on kyseinen suhde.

Esimerkki: rakennukselle tai sen omistavalle henkilölle ei tule määritellä ominaisuustietoina esimerkiksi linkitetyn listan kaltaista linkaaren hallintaan käyttävää rakennetta pysyvine ja versiotunnisteineen – nämä rakenteet ovat historiatietoa (provinienssia), eli eivät ole itse rakennuksen tai henkilön käsitteelliseen määritelmään kuuluvia, vaan itse omistussuhteen todennettavissa olevaa historiaa metadatanä.

NFR4 (laajennettavuus ja peittävyys)

Laajennettavuutta (engl. extendibility) ja peittävyyttä (engl. semantic coverage) koskevan vaatimuksen mukaan

- RYYO:n tulisi kyetä vastaamaan tuleviin ennakoituihin ja ennakoimattomiin käyttötarpeisiin mahdollistamalla RYYO:n laajentamisen siten, että se ei vaadi olemassa olevien rakenteiden määrittelyn välitöntä ja usein toistuvaa korjaamista.³
- RYYO:n tulisi sisältää rakenteet, joiden varaan voidaan määritellä kaikkia niitä entiteettityyppejä, joita rakennetun ympäristön tietoaalueella tarvitaan.
- RYYO:n laajennettavuus tulisi toteuttaa mahdollisuuksien mukaan modularisoimalla eli pilkkomalla ontologioita eri näkökulmien kautta osiin ja linkittämällä niitä toisiinsa tarpeen mukaan. Laajentamisen tulee tapahtua transitiivisesti.

³ Tällä ei tietenkään tarkoiteta sitä, että ontologian rakenne on täysin muuttumaton, vaan sitä, että ontologian laajentaminen ei lähtökohtaisesti pitäisi johtaa sen rakenteen jatkuvaan uudelleenarviointiin ja korjaamiseen. Ajan myötä kaikki käsitejärjestelmät muuttuvat, pakottaen pidemmällä aikavälillä tarkastelemaan myös perustana olevien käsitteiden ajantasaisuutta.

Modularisoinnilla tarkoitetaan sitä, että ontologiasta pyritään irrottamaan osia, jotka ovat sellaisenaan uudelleenkäytettäviä mahdollisimman useassa kontekstissa. Esimerkiksi yleiset julkishallinnon rooleja, asioita, päätöksiä ja menettelyjä koskevat rakenteet ovat rakennetun ympäristön lisäksi relevantteja lukuisilla muilla hallinnonaloilla, ja ne tulisi tämän vuoksi sisällyttää omaan JH-ontologiaansa hyödynnettäväksi mm. RYYO:ssa.

Esimerkki: rakennusta ei tule olla määritelty siten, että uusien rakennustyyppien lisääminen sen alaluokiksi pakottaa muuttamaan olemassa olevaa määritelmää, josta aiemmin sen alaluokiksi määritellyt rakennustyyppit ovat riippuvaisia.

NFR5 (ontologiset sitoumukset)

Ontologisia sitoumuksia (engl. ontological commitments, modelling precision) koskevan vaatimuksen mukaan

- RYYO:n tulee perustua pienimpään joukkoon määritelmiä, jolla sen kuvaaman tietäalueen sisällöt pystytään peittävästi kuvaamaan.
- RYYO:n ei tule rajata määritelmiä liian kapeiksi asettamalla niille liiallista määrää reunaehtoja, eikä myöskään määritellä sellaisia rakenteita, jotka ovat selvästi aihepiirin ulkopuolisia.

Ontologisten sitoumusten minimointi maksimoi sen käytettävyyden, sillä tällöin hyödyntävä taho joutuu mukauttamaan omaa käsitteistöään mahdollisimman vähän voidakseen ottaa RYYO:n käyttöön. Tämä toteutuu, kun RYYO:n määritelmiä on minimaalinen määrä ja niillä on maksimaalinen peittävyys.

Esimerkki: rakennusta ei tule määritellä yleispätevästi olioksi, jonka ominaisuuksiin täytyy sisältyä kaikki rakennusluvituksessa tarvittavat tiedot, sillä kaikista rakennuksista näitä tietoja ei ole saatavilla, eivätkä tiedot ole kaikissa rakennuksissa koskevilla käyttötapauksissa relevantteja.

NFR6 (yhteentoimivuus)

Yhteentoimivuutta koskeva vaatimus on peräisin Ympäristöministeriön yhteentoimivuusmenetelmän käyttöä koskevasta linjauksesta, Tiedonhallintalaista, EIF-kehikosta sekä eri aloilla tehdystä empiirisestä tutkimuksesta. Vaatimuksen mukaan

- RYYO:n tulee toteuttaa määrittelyprojektissa linjatun mukaisesti rakennetun ympäristön tiedon yhteentoimivuuden edistämistä, huomioiden laajemman julkishallinnon yhteentoimivuuden edistämisen tavoitteet.
- RYYO:n toteutus tulee olla vapaa tiettyjen teknologioiden tai palveluntarjoajien monopolista (engl. vendor lock-in).
- RYYO:n tulisi hyödyntää jo tehtyjä vakaita määrittelyjä mahdollisuuksien mukaan referenssiaineistona tai oman rakenteensa pohjana.
- RYYO tulisi olla määritelty avoimien standardien varaan.
- RYYO:n soveltamiseen tulisi olla saatavilla laadukkaita ja ajantasaisia avoimen lähdekoodin kirjastoja, alustoja, ja työkaluja.

Vaatimuksen tavoitteena on varmistaa yhteentoimivuuden laaja-alainen toteutuminen. Tavoite on realistinen vain tilanteessa, jossa toimijat eivät jakaudu niihin, jotka kykenevät investoimaan

kaupallisiin teknologioihin ja standardeihin sekä niiden koulutukseen, ja niihin, jotka eivät pysty kehittämään riittäviä kyvykkyksiä investointikynnyksen vuoksi.

2.2 Kuvauskielten kartoitus

Tässä kappaleessa kuvataan karkealla tasolla vaatimusmäärittelyistä johdettu ratkaisuavaruus, eli niiden potentiaalisten ratkaisujen joukko, jolla vaatimuksiin voidaan jollain tasolla vastata.

2.2.1 Käytettävissä olevat kielet

Mahdollisia standardoituja, suosituksiin pohjautuvia ja laajalti käytössä olevia tietorakenteiden (engl. information model) tai tiedon kuvaamiseen (engl. knowledge representation) soveltuvia kieliä ovat mm.:

- UML (Unified Modeling Language), määritelty standardiperheessä ISO 15959 ja saatavilla uudempaan OMG:n julkaisemana määrittelynä (Object Management Group, 2017) sekä sen yhteydessä mahdollisesti käytettävä OCL (Object Constraint Language), määritelty standardissa 15959 ja saatavilla uudempaan OMG:n julkaisemana määrittelynä (Object Management Group, 2014b)
- UML-profiilina määritelty SysML (System Modeling Language), määritelty standardissa ISO 15959 ja saatavilla uudempaan OMG:n julkaisemana määrittelynä (Object Management Group, 2019)
- UML-profiilina määritelty GFM (General Feature Model), määritelty standardissa ISO 15959, ja saatavilla laajennettuna uudempaan OMG:n julkaisemana määrittelynä (Herring, 2020).
- RDF (Resource Description Framework), määritelty avoimessa W3C-standardissa
- RDFS (Resource Description Framework Schema), määritelty avoimessa W3C-standardissa (Brickley & Guha, 2014)
- OWL (Web Ontology Language), määritelty avoimessa W3C-standardissa (Bock ym., 2012)
- SKOS (Simple Knowledge Organization System), määritelty avoimessa W3C-standardissa (Miles & Bechhofer, 2009)
- GeoSPARQL core, määritelty avoimessa OGC-standardissa (Perry & Herring, 2012)
- SHACL (Shapes Constraint Language), määritelty avoimessa W3C-standardissa (Knublauch & Kontokostas, 2017)

Kuvauskielten piiriin kuuluvat myös *ad-hoc*-tyyppiseen mallintamiseen käytetyt ER-diagrammien (engl. entity-relationship diagram) Crow's foot ja Chen's -notaatiot, sekä NoUML-paradigman mukaiset kuvaukset.

Lisäksi on olemassa edellä listattuihin kieliin liittyviä eriasteisen yhteentoimivuuden määrittelyyn tarkoitettuja kehikoita, joiden olemassaolo on hyvä tiedostaa:

- O-DEF (Open Data Element Framework), saatavilla The Open Groupin määrittelemänä (The Open Group, 2020)
- ODM (Ontology Definition Metamodel), saatavilla OMG:n julkaisemana määrittelynä (Object Management Group, 2014a)
- DOL (Distributed Ontology, Modeling and Specification Language), saatavilla OMG:n julkaisemana määrittelynä (Object Management Group, 2018)

- MFI (Metamodel framework for interoperability), määritetty standardiperheessä ISO 19763
- XMI (XML Metadata Interchange), määritelty standardissa ISO 19509 ja saatavilla uudempana OMG:n julkaisemana määrittelynä (Object Management Group, 2015)

Listat eivät ole kattavia, mutta sisältävät olennaisimmat keskeiset kielet, joita määrittelyyn voitaisiin käyttää (liian tiukasti tiettyyn perspektiiviin rajatut kielet kuten BPMN, ArchiMate jne., sekä harvoin käytössä nähdyt kuten IDEF1X on tiputettu pois vertailusta).

2.2.2 Kielten välisistä eroista

Edellä mainitut kielet eivät ole yhteismitallisia, eli niiden suunnitellut käyttökohteet ja ilmaisuvoima vaihtelevat huomattavasti. Osa kielistä sisältää vahvoja reunaehtoja, jotka perustuvat niiden suunniteltuihin käyttökohteisiin.

Kielet käsitellään vaatimuskehikon kautta jaettuna kahteen ryhmään: metamallipohjaiset kielet (Model Driven Architecture -paradigman mukaan) ja linkitetyn datan kielet (Ontology Engineering -paradigman mukaan). Jako on tehty näiden kahden paradigman dominanssin pohjautuen (Kiko & Atkinson, 2008). Muita kieliperheitä ja yksittäisiä kieliä ei tämän projektin aikataulun puitteissa pystytä kattavasti ottamaan huomioon.

2.2.2.1 Metamallikielet

Metamallikielillä (engl. meta-modeling languages) tarkoitetaan sellaisia kuvauskieliä, jotka itse perustuvat johonkin toiseen niiden rakennetta määrittävään kieleen. Metamallikielistä ylivoimaisesti tunnetuin on MOF-pohjainen UML.

UML on pitkään käytössä ollut ja laajalti tunnettu ohjelmistokehityksen tarpeisiin kehitetty mallinnuskieli, jolla pystytään esimerkiksi tuottamaan ohjelmiston lähdekoodia tai tietokantakuvausvisuaalisia kaaviomuotoisen suunnitelman pohjalta. UML sisältää 7 rakenteita kuvaavaa kaaviotyyppiä sekä 8 käyttäytymistä kuvaavaa kaaviotyyppiä, jotka yhdessä kattavat kehitettävän ohjelmiston lähes kaikki osa-alueet. Ylivoimaisesti useimmin hyödynnetään kuitenkin vain yhtä kaaviotyyppiä (luokkakaavio). UML:n systemaattinen käyttö yhdistyy vahvasti vesiputousmalliseen kehittämiseen.

UML:a esimerkkinä käyttäen, malleja ja metamalleja on neljässä tasossa:

- M0) taso, jolla kuvataan tiettyjä yksittäisiä entiteettejä (esimerkiksi tietyn asemakaavan sisältö tietomallimuotoisena)
- M1) taso, jolla määritellään rakenne, jolla edellisen tason sisältö tulee kuvata (asemakaavan tietomallin rakenne UML-luokkakaaviona)
- M2) taso, jolla määritellään rakenne, jolla mm. UML-luokkakaavioita tulee kuvata (UML:n kielioppi)
- M3) taso, jolla määritellään rakenne, jolla UML:n kielioppi on kuvattu (nk. MOF, Meta-Object Facility)

Asiaa monimutkaistaa se, että M1-tasolla voidaan käyttää nk. UML-profiileja, jotka ovat UML:lla määriteltyjä kokoelmia valmiita mallirakenteita (erikoistamisen kautta lisättyjä rajoitteita) tiettyjä käyttökohteita varten. Niillä voidaan yhdenmukaistaa ja helpottaa tietyn tietoaalueen mallintamista joutumatta määrittämään MOF-tasolla uutta UML:n kaltaista kieltä vastaavilla rakenteilla. Esimerkkejä UML-profiileista ovat mm. järjestelmäsuunnitteluun luotu SysML, ja paikkatiedon

kuvailuun mm. INSPIRE-direktiivin yhteydessä käytettävä General Feature Model (GFM, ISO 19109). Siinä missä UML ei itse ota kantaa mallinnettavaan sisältöön, GFM ottaa kuvauksen lähtökohdaksi paikkatietoperspektiivin ja SysML järjestelmän osien sekä niiden välisen toiminnan. Profiileita on kehitetty runsaasti eri käyttötarkoituksiin aina OWL tai Archimate-kuvauksista TOGAF:iin.

UML:n dominoivan roolin vuoksi seuraavat vaatimukset käsitellään pääosin sen sekä sitä tarkentavien profiilien (joita voisi tässä kontekstissa kutsua ”UML-pohjaisiksi sanastoiksi”) kautta.

Toiminnalliset vaatimukset

UML kykenee tarjoamaan sanastojen käyttöön formaaleja tietorakenteita, mutta yhtenäistä tapaa yhdistää käsitteellistä ja loogista mallia itse UML ei sisällä.

UML-rakenteet määritellään nk. teknisillä nimillä, joilla ei ole tunnisteena toimimisen lisäksi mitään yhteyttä itse mallin sisältöön (nk. weak semantics), mutta jotka auttavat esimerkiksi tietoarkkitehteja tunnistamaan skeemasta rakenteet, jotka vastaavat tiettyjä substanssitarpeita (esimerkiksi luokka nimeltä *Rakennus* jonka metaluokka on *FeatureType*).

UML kykenee tarjoamaan luokkakaavioiden muodossa sekä pohjan tietomallien määrittelyyn että niiden yhdistämiseen sillä ehdolla, että yhdistettävät mallit ovat UML-muotoisia. Yhdistämiseen voidaan rajatuissa määrin käyttää UML-profiileja, jotka mahdollistavat jonkinasteisen käänteiskuvauksen UML-muotoon, mutta UML-mallien harmonisointi on käytännön tasolla realismia vain, kun mallit tuotetaan käsin.

Korkean tason ilmaisuvoimaisena kielenä UML:n varaan voidaan määritellä lähes mitä tahansa toiminnallisuutta, mutta UML sellaisenaan ei tarjoa suoraviivaista tukea muodoltaan hajanaisten ja eri rajapintojen kautta tarjottavien tietojen kuvailuun, ja tarjolla olevien eri asteisten metatietokehikoiden hyödyntäminen vaatisi merkittävää investointia niiden maturiteetin ja sovellettavuuden arviointiin.

Rakennetun ympäristön piirissä yksi tunnetuimmista tietoaaluekohtaiseen mallintamiseen käytetyistä UML-profiileista on edellä mainittu GFM. Se on myös keskeinen INSPIRE-direktiivin hyödyntämä malli tietorakenteiden määrittelyyn. MOF-kehikon mukaisesti sen alla käytettävät kielet ovat pääasiallisesti erikoistavia, mihin liittyen INSPIRE:n oma käsitelmä (versio 3.4rc2) kappale 6.1 toteaaakin (korostus oma):

”Another important principle is that the scope of INSPIRE is spatial data. I.e., most non-spatial aspects of the themes are out-of-scope of INSPIRE data specifications. INSPIRE should not aim at being the information infrastructure for all data relevant for environmental applications. Much of the data in this context will have some relation to space and time, but the scope of INSPIRE is mainly focussed on the spatial aspects.”

Haasteeksi muodostuukin tiukasti erikoistamiseen perustuvaan mallintamisparadigmaan liittyvä kiinteä kytkentä erityisesti XML-muotoisiin soveltamisprofiileihin, joiden rakenne ja ilmaisuvoima eivät vastaa UML-profiileja (Chojka, 2020; Herring, 2020; Jetlund, 2020).

NFR1 (selkeys)

UML:n määrittelyn tulkinnanvaraisuus on pitkään tiedossa ollut ongelma (Ciccozzi ym., 2019; Evans ym., 2003; Pham ym., 2018; Seidewitz, 2014; Shan & Zhu, 2008), joka on kuitenkin tyypillisesti sivuutettu olemassa olevien manuaalisten konventioiden vuoksi, joilla mallien ja niiden pohjalta tuotettujen järjestelmien välistä dissonanssia hallitaan kehitystyön aikana. Usein UML-kaavioita tulkitaankin sovellettaessa pitkälti vain visuaalisena ohjeena varsinaisen teknisen toteutuksen aikana:

”We attribute the mismatch between UML being formally verifiable and UML being executable to the original goal of UML to be both human centred and a tool for communication between human designers, allowing a significantly loose semantics, with little intention to be executable” (Carrillo ym., 2021).

NFR2 (johdonmukaisuus)

Edellä kuvattu tulkinnanvaraisuus liittyy kiinteästi sisäiseen ristiriidattomuuteen, jota UML:ssa ei ole voitu varmentaa formaalisti sen eteen tehdystä kehitystyöstä huolimatta. Osittainen formalisointi (nk. executable UML) on tuotettu, mutta se kattaa vain rajatun osan koko määrittelystä (Object Management Group, 2021).

NFR3 (kuvaustapavinouma ja vastuiden erottelu)

UML:n geneerisyyden vuoksi se sisältää hyvin vähän kuvaustapavinoumia; esimerkiksi primitiivityypit määritellään hyvin yleispätevästi. Vinoumien määrä kasvaa tyypillisesti radikaalisti hyödynnettäessä alakohtaisia UML-profileja.

NFR4 (laajennettavuus ja peittävyys)

UML on kielenä erittäin laajennettava, mutta sen laajennettavuus kärsii UML-profiilien sisältämistä vahvoista ontologisista sitoumuksista ja useamman profiilin yhteisestä käytöstä potentiaalisesti seuraavasta semanttisesta epäselvyydestä.

UML-profiilien tietalueiden rajaukset voivat mennä osittain päällekkäin, jolloin käyttäjälle voi tarjoutua useampi kuin yksi tapa mallintaa leikkauskohtaan kuuluvia entiteettejä. Ei voida myöskään lähtökohtaisesti olettaa, että kyseisten rakenteiden välille voisi muodostaa johdonmukaisesti kuvaussuhteen (yksi-, tai kaksisuuntaisen). Useiden UML-profiilien yhtäaikainen käyttö on haastavaa, ja malleista voi tulla hyvin monimutkaisia validoitavia, verifioitavia ja ylläpidettäviä.

NFR5 (ontologiset sitoumukset)

UML sisältää määrittelydokumentissa vahvan eksplisiittisen sitoumuksen ohjelmistopohjaisten järjestelmien näkökulmaan. Määrittelyn kappaleessa 6.3. kuvataan selkeästi, että kohdetta, jota UML:lla mallinnetaan, tarkastellaan sitä käsittelevän järjestelmän näkökulmasta.

Merkittäviä muita sitoumuksia ovat mm. UML:n identiteettirakenteen puute (Object Management Group, 2014a), suljetun maailman oletus (engl. Closed-World Assumption), moninperinnän toteutustavan riippuvuus valitusta UML-profilista, sekä perinnällä erikoistamisen vahva rooli läpi koko M0...M3 kielihierarkian.

NFR6 (yhteentoimivuus)

Useiden eri lähteiden mallien yhdistäminen ei ole täysin ongelmaton, sillä UML:n rakenteen määrittely on ensi sijassa graafiseen notaatioon ja sanalliseen kuvaukseen pohjautuva, ja mallien siirtämisen mahdollistava XMI-kuvaus UML:sta on toissijainen suhteessa standardin määrittämiin rakenteisiin. XMI ei ole määrittelyiltään kattava eikä ristiriidaton, eikä siksi pysty tarjoamaan kaksisuuntaista yksi yhteen -kuvausta MOF-pohjaisista kielistä⁴. Ohjelmistojen välistä validoitua yhteentoimivuutta ei siis käytännössä tällä hetkellä ole kuin rajatun ominaisuusjoukon osalta. Varteenotettavalla tasolla yhteentoimivaan mallintamiseen kelpaava ohjelmistotarjonta on suhteellisen rajattua (QPR EA, Sparx EA, Eclipse Papyrus, ym.).

2.2.2.2 Linkitetyn datan kielet

Linkitetyn datan (LD) kielillä tarkoitetaan kieliperhettä, jonka kieliä on kehitetty mm. semanttisen Webin tarpeisiin viimeiset päälle 20 vuotta. Kielien tehtävänä on mahdollistaa useiden itsenäisten toimijoiden verkostoitunut tiedonvaihto siten, että tietoa ei kysellä sen rakenteen vaan merkityksen perusteella, tietoa voidaan yhdistellä tehokkaasti, ja siitä voidaan tehdä kielestä riippuen enemmän tai vähemmän edistynyttä konepäätelyä.

Siinä missä edellä esitelty MOF-kieliperhe rakentuu nk. suljetun maailman periaatteelle, linkitetyn datan kielissä lähtökohtana on nk. avoin maailma. Avaamatta tarkemmin aiheen taustoja, erot voidaan käytännössä tiivistää siihen, että suljetussa maailmassa tietorakenteiden oletetaan olevan kattavia (kaikki tiettyä entiteettiä koskevat tiedot sisältyvät sitä kuvaavaan luokkaan). Avoimessa maailmassa oletuksena on, että tietorakenteita voidaan aina täydentää, sillä oletuksena on, ettei kenelläkään toimijalla ole kaikkea tietyn entiteetin määrittelyä koskevaa tietoa hallussaan.⁵

Kuten edellä kerrotusta voi päätellä, linkitetyn datan kielet on kehitetty varta vasten tiedon yhteentoimivuuden edistämiseen. Yksi keskeisistä lähtökohdista on ollut oletus siitä, ettei monen toimijan ympäristössä voida aina (eikä välttämättä ole tarpeen) muodostaa ja ylläpitää yhtä harmonisoitua auktoriteettina toimivaa määritelmää jokaiselle kaikkien toimijoiden käsittelemälle entiteetille: yhteentoimivuus voidaan saavuttaa entiteettien linkittämällä keskenään joko löyhästi heuristiikkaan pohjautuen, tai täsmällisesti formaaliin logiikkaan pohjaten. Linkitettyä dataa voidaan kuitenkin luonnollisesti soveltaa myös kontrolloiduissa rajatuissa ympäristöissä, jolloin myös yhtenäisten harmonisoitujen määritelmien käyttökin on helpompaa.

Keskeinen linkitetyn datan peruskivi on Resource Description Framework (RDF), WWW-standardeihin nojaava formaali kieli ja avoin standardi, josta on tullut yleinen tapa siirtää ja käsitellä graafimuotoista (eli linkitettyä) tietoa. RDF on hyvin simppelempi kielioppi, ja tarkoitettu käytettäväksi etenkin ilmaisuvoimaisempien formaalien kielten (kuten RDF Schema, SKOS tai OWL) kanssa, jotka mahdollistavat rikkaan konepäätelyn (kuten uuden tiedon johtamisen tai aineiston lauseiden totuudenmukaisuuden tarkistamisen) RDF-muotoisesta aineistosta.

⁴ XMI-spesifikaatiossa on UML/MOF:n spesifikaation sekä XML-spesifikaation monimutkaisuuden vuoksi lukuisia avoimia virheitä (<https://issues.omg.org/issues/spec/XMI>).

⁵ Tämä ei tarkoita, etteikö avoimessa maailmassa voitaisi määrittää tiettyä käyttötarkoitusta varten tiedon rakennetta sääteleviä skeemoja XSD:n tapaan, vaan että esimerkiksi luokat eivät oletuksena ole suljettuja, ellei niin erikseen todeta. Suurin ero paradigmojen välillä on loogisessa päätelyssä: suljetussa maailmassa tieto, jota ei tiedetä, oletetaan automaattisesti epätodeksi – avoimessa maailmassa sen totuusarvoon ei oteta kantaa (koska se voi olla kumpi vain).

RDF:ssä tietoa ilmaistaan lauseina nk. kolmikoilla (engl. triples), jotka ovat muotoa <subjekti, predikaatti, objekti>. Näistä subjekti ja objekti ovat entiteettejä ja predikaatti niitä yhdistävä assosiaatio. Subjekti ja predikaatti ovat aina URI-resursseja (esimerkiksi `https://uri.suomi.fi/datamodel/rak/Rakennus`), objekti voi olla joko URI-resurssi tai literaaliarvo (esimerkiksi merkkijono). Useimmiten luettavuuden vuoksi URI-resurssit esitetään URI-nimiavaruuden avulla lyhennetyssä muodossa (esimerkiksi `rak:Rakennus`). URI:t voivat olla myös URN-muotoisia, eli tietyn nimeämiskäytännön (UUID, ISBN, DOI, jne.) mukaisia pysyviä tunnuksia.

URI-resurssien ei täydy osoittaa toimivaan osoitteeseen (ts. ”resolvoitua”), mutta tyypillisesti niihin kuitenkin halutaan kytkeä tietoa, joka kytkeytyy tietomallissa kyseiselle URI:lle annettuun rooliin. Mallin tietoa käsitteleville toimijoille ja ohjelmistoille halutaan usein tarjoilla eri esitys (tai sisältö) HTTP-otsikkotietojen perusteella: käyttäjälle voidaan esimerkiksi näyttää visualisointi tai muu dokumentaatio tietosisällöstä, ohjelmistolle taas tarjoilla vastaava koneluettavana RDF-sanomana.

Kuten yllä todettiin, kolmikoilla ilmaistaan tietoa. Esimerkiksi `kunta:<UUID> rdf:type rak:Rakennus` kertoo, että kyseisen UUID-arvon sisältävä URI-resurssi on yhdistetty predikaatilla `rdf:type` resurssiin `rak:Rakennus`. Aineistoa parsivat ohjelmistot osaavat tunnistaa tiettyihin URI-arvoihin liitettyjä loogisia sääntöjä, esimerkiksi sen, että em. lauseen predikaatilla halutaan kertoa UUID:n kuvaavan rakennusta. Laajentamalla kuvausta muilla lauseilla, on mahdollista rikastaa kuvausta rajattomasti.

On olennaista ymmärtää, että toisin kuin mm. UML:ssa, itse UUID-arvoon päättyvä subjektin URI tai sen edustama tietue ei millään tavoin ”omista” objektina olevaa URI:a (tai sen sisältöä), saati niitä yhdistävää assosiaatiota. RDF:ssä kaikki resurssit ovat itsenäisiä (atomisia), myös assosiaatiot ja attribuutit – niiden olemassaolo ei ole siis sidottu luokkien tai niiden instanssien olemassaoloon.

Uudelleenkäytettävyyden mahdollistava resurssien atomisuus on merkittävä yhteentoimivuuden kulmakivi. Sen ansiosta ydintietomäärittelyissä voidaan keskittyä tiedon toisteisuuden minimointiin, mikä osaltaan edesauttaa hallittavan tiedon pitämisessä eheänä.

Esimerkiksi erisnimeä kuvaava resurssi voidaan määrittää vain kerran, ja käyttää sen URI:a kaikissa tilanteissa, joissa jollain toisella entiteetillä (oli kyseessä sitten henkilö, rakennus, organisaatio) on tarvetta erisnimen kaltaiselle tunnisteelle. Näin toimiessa ei synny tilannetta, jossa erisnimeä ilmaiseva attribuutti olisi nimetty eri tavoin eri entiteeteillä, ja kaikki erisnimeä käyttävät entiteetit on mahdollista saada selville niitä kuvaavista malleista yhdellä yksinkertaisella kyselylausekkeella.

Keskeisiä RDF-pohjaisia kieliä ovat yksinkertainen RDF Schema, formaalin konepäättelyn mahdollistavien ontologioiden rakentamiseen suunniteltu Web Ontology Language (OWL), koneluettavien sanastojen ja käsittemallien määrittelyyn suunniteltu ja OWL:n varaan rakennettu Simple Knowledge Organization System (SKOS), ja tietorakenteiden sisällön validointiin sekä itse rakenteiden muodon sääntelyyn suunniteltu Shapes Constraint Language (SHACL).

Tyypillisesti RDF-muotoista tietoa välitetään XML, JSON-LD, tai Turtle-muotoon serialisoituna.

Toiminnalliset vaatimukset

Itse RDF:ää käytetään tuskin koskaan sellaisenaan tietorakenteiden määrittelyyn, sillä se ei sisällä kuin hyvin alkeellisen kieliopin. Rakenteet määritellään käytännössä aina em. ilmaisuvoimaisempien kielten varaan. Sanastojen määrittelyyn voidaan käyttää suoraan SKOS:ia, ja ne voidaan yhdistää

käsitteiden URI-tunnisteiden kautta minkä tahansa muun RDF-kielen mukaisiin tietorakenteisiin. Muiden mallien mukaista tietoa voidaan yhdistää, mikäli kyseiset tiedot ovat tarjolla esimerkiksi REST-resursseina, jolloin niihin voidaan viitata suoraan resurssien URI-tunnisteilla.

URI-tunnisteilla ei UML-nimien tapaan ole mitään yhteyttä itse tietosisältöihin, mutta URI:t takaavat sen, että jokainen saman URI:n ilmentymä kuvaa samaa rakennetta.

RDF sekä sen varaan määriteltyt kielet tarjoavat UML:n tapaan erittäin suuren ilmaisuvoiman, lisäksi ne mahdollistavat hyvin monimuotoisten tietojen kuvailun. Tietoaluekohtaiseen mallintamiseen LD-puolella on hyödynnetty usein RDF-pohjaisten kielten varaan määriteltyjä paikallisia rakenteita, eikä selkeitä standardoitujen UML-profilien tapaisia määrittelyitä ole laajalti käytössä. Tällaisia on kuitenkin tarjolla; esimerkiksi ISO:n ylläpitämästä paikkatieto-ontologioiden varannosta löytyy mm. GFM:n OWL-kuvaus⁶, ja IFC:stä on BuildingSMARTin kautta tarjolla IFC4x1-version OWL-kuvaus⁷.

NFR1 (selkeys)

RDF ja RDF-pohjaiset kielet on aksiomatisoitu ja ne ovat tämän vuoksi täysin määritelmällisesti läpinäkyviä. Määritelmät on dokumentoitu sanallisesti, mutta niiden korkea formaliteetin aste heikentää saavutettavuutta laajemmassa käyttäjäjoukossa ja vaatii tuekseen selventävää kirjallisuutta tai tutoriaaleja. Tämä muodostaa merkittävimmän kynnyksen RDF-pohjaisten kielten laajalle hyödyntämiselle.

NFR2 (johdonmukaisuus)

Edellä kuvatun mukaisesti kielet ovat sisäisesti johdonmukaisia. Avoimen maailman oletuksen vuoksi niiden varaan on kuitenkin mahdollista rakentaa malleja, jotka sallivat ristiriitojen lausumisen, tai sellaisten rakenteiden määrittelyn, jotka eivät ole konepäättelviä (engl. undecidable). Tämä lisää mallintajiin kohdistuvia osaamisvaatimuksia.

NFR3 (kuvaustapavinouma ja vastuiden erottelu)

Merkittävin potentiaalinen kuvaustapavinouma liittyy literaalien tyypittämiseen yleisesti käytettyjen XSD-datatyyppeiden määritelmiin ja niiden väliseen hierarkiaan mm. arvoja vertaillessa. XSD-datatyyppeiden hyödyntäminen on vapaaehtoista, mutta niille ei ole tällä hetkellä olemassa uskottavaa laajasti käytettyä vaihtoehtoa.

NFR4 (laajennettavuus ja peittävyys)

Linkitetyn datan esittelyn ohessa kerrotun mukaisesti määrittelyjä on mahdollista laajentaa triviaalisti linkittämällä resursseja keskenään. Merkittävimmät haasteet laajennettavuudelle muodostuvat tilanteessa, jossa määrittelyissä hyödynnetään pitkälle jalostettua OWL-pohjaista konepäättelyä. Resurssien välisiä yhteyksiä koskevat triviaalit kyselyt onnistuvat laajentamisen tavasta riippumatta, mutta virheellisesti toteutettu laajentaminen saattaa konepäättelyn osalta johtaa herkästi joko epäintuitiivisiin (vaikkakin loogisesti johdonmukaisiin) tai ristiriitaisiin päättelyn tuloksiin.

⁶ <https://def.isotc211.org/>

⁷ https://standards.buildingsmart.org/IFC/DEV/IFC4_1/OWL/

NFR5 (ontologiset sitoumukset)

Merkittävin mallintamistapaan vaikuttava sitoumus on assosiaatioiden binäärisyys, sekä kömpelö tapa kuvata lausuttua sisältöä koskevaa metatietoa. RDF:llä ei siis voida esimerkiksi ilmaista kolmen osapuolen vuorovaikuttavan keskenään yhden assosiaation kautta, vaan tähän tarvitaan kolme lausetta (yksi jokaisen parin yhdistämiseen).

Lisäksi RDF ohjaa *reiffoimaan* eli luomaan uuden rakenteen (hieman relaatiomallisen yhdistelmätaulun tapaan) tilanteissa, joissa halutaan esimerkiksi todeta tietyn assosiaation yhdistävän kahta osapuolta vain tietyn ajan. Jälkimmäiseen on ratkaisuksi kehitteillä RDF-star -määrittely⁸, joka mahdollistaa minkä tahansa kolmikon käyttämisen kolmikon subjektina tai objektina, ja kasvattaa RDF:n ilmaisuvoimaa merkittävästi.

NFR6 (yhteentoimivuus)

Esittelyssä kuvatun mukaisesti yhteentoimivuus on linkitetyn datan keskiössä. Yksinkertaisen perusrakenteen vuoksi rakenteiden siirtymisessä ei ole epäyhteensopivuutta työkalujen välillä.

Ohjelmistotarjonta on laaja, mutta julkisesti saatavilla olevasta tarjonnasta merkittävä osa on akateemista nk. abandonwarea. Kaupallisella puolella useat mallinnuskyvykkyyden tarjoavat tuotteet ovat laajoja kokonaisratkaisuja ja lisenssihinnoiltaan UML-mallintimia kertaluokkaa kalliimpia (Stardog, RDFox, TopBraid Suite, jne.). Ilmaisia ja avoimen koodin järjestelmäratkaisuja on saatavilla (GraphDB, RDF4j, TopBraid Composer, Apache Jena, AnzoGraph, jne.).

2.2.3 Yhteenvedo

Edellä tehtyä vertailua ei tule pitää kattavana tai aukottomasti toisen vaihtoehdon poissulkevana. Tällä hetkellä rakennetun ympäristön määrittelyjä on tuotettu ja tuotetaan molempien mukaisina, joten on tärkeää tämentää, mihin käyttöön kumpainenkin vaihtoehto parhaiten soveltuu. Esimerkkinä UML:n laajasta soveltamisesta, myös merkittävä osa linkitetyn datan standardeista hyödyntää UML-notaatiota perusrakenteidensa esittämiseen formaalin notaation ohella.

Puhtaasti loogisen tason mallintamisen, sanastotyön ja kansallisen (sekä ylikansallisen) yhteentoimivuuskehikon asettamissa puitteissa linkitetyn datan paradigma on perusperiaatteiltaan UML:a lähempänä toiminnallisten vaatimusten tarpeita, ja hyvin positioitunut kansainvälisessä yhteentoimivuuden kehityksessä (vrt. EU tai esimerkiksi Australian hallituksen Linked Data WG).

Lisäksi linkitettyyn dataan pohjautuen on tuotettu viime vuosina merkittävä määrä rakennetun ympäristön tietomalleja, koodistoja ja sanastoja Yhteentoimivuusalustalle. Alustan rooli tulee kasvamaan entisestään, ja sille ollaan tällä hetkellä tekemässä merkittäviin parannuksiin tähtäävää uudistusta niin käyttöliittymän, toiminnallisuuden kuin hallintamallin osalta.

⁸ <https://w3c.github.io/rdf-star/cg-spec/>

3 Määrittely

RYYO:n sisällön määrittely aloitettiin kartoittamalla, millaisen käsitteellisen vakaan pohjan päälle määrittelyä voitaisiin ryhtyä tekemään. Kilpailevia – ja enemmän tai vähemmän selkeästi ilmaistuja – periaatteita taksonomiseen luokitteluun (luokka-aliluokka -suhteet), mereologiaan (formaali koostumussuhteiden määrittely) ym. löytyy runsaasti niin standardeista, sanastoista, kuin yksittäisistä tietomalleistakin. Määrittelyt eivät kuitenkaan ole valtaosassa tapauksia perusteiltaan keskenään yhteensopivia, tai määriteltä läpinäkyvästi siten että yhteensopivuudesta voitaisiin varmistua.

Keskeisenä lähtöaineistona tälle projektille olivat Iso-Britanniassa hallituksen käynnistämän digitaalisen rakennetun Britannian ohjelman puitteissa tuotetut selvitykset yläontologioista (Partridge ym., 2020) ja teollisuudessa käytetyistä tietomalleista (Leal ym., 2020). Selvitysten tehtävänä oli tukea kansallisesti yhtenäisen tiedonhallinnan kehityksen (engl. information management framework) ja ”perustietomallin” (engl. foundational data model) määrittelyä.

Kyseisten selvitysten rinnalla perehdyttiin myös meneillään olevaan ISO 21838-standardiperheen määrittelyyn. Julkaistuista osista ISO 21838-1 määrittää yleiset kriteerit yläontologioiden tuottamiseen. Osat ISO 21838-2 ja ISO 21838-3 ovat alun perin EU-rahoitteisessa WonderWeb-ontologiaprojektissa tuotettuja ja sen jälkeen lukuisissa muissa projekteissa sekä teollisuuden käytössä jatkokehitettyjä yläontologioita. Näistä ISO 21838-2 eli BFO (Basic Formal Ontology) on valmistunut standardiksi, ja ISO 21838-3 eli DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) on vielä standardointivaiheessa.

3.1 DOLCE

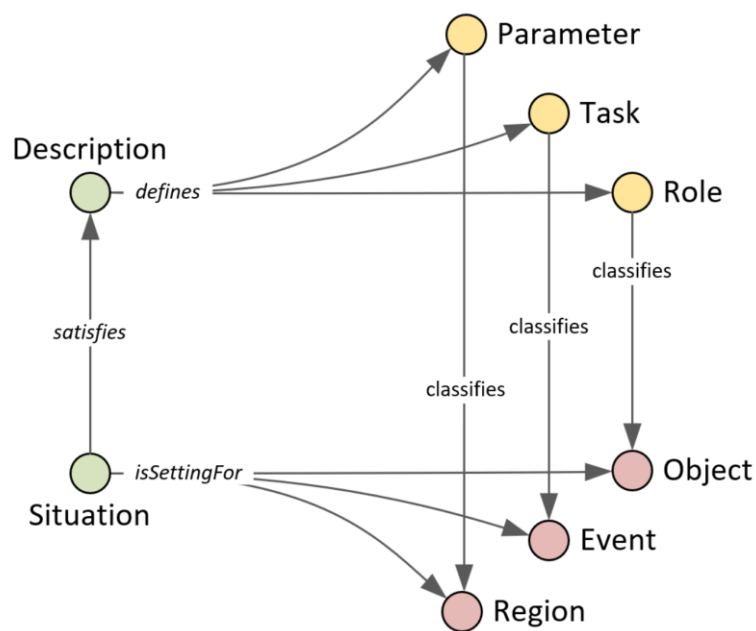
Määrittely päädyttiin lopulta perustamaan DOLCE:n varaan sekä sen tulevan standardoinnin, rakenteen että kansallisen käytön vuoksi: Kansalliskirjaston lukuisia laajoja ontologioita sisältävä Finto-palvelu⁹ on harmonisoinut ne DOLCE:n perusrakenteen mukaisesti. BFO:n valintaa olisi puoltanut sen erittäin laaja kansainvälinen käyttäjäkunta (genetiikka, biolääketiede ja lääketeollisuus) sekä tuore laajentuminen aiemman käyttäjäkuntansa ulkopuolelle (mm. Yhdysvaltojen ja Iso-Britannian puolustusministeriöiden tiedonhallinnan kehittäminen).

DOLCE:en päädyttiin kuitenkin siitä syystä, että rakennettu ympäristö on em. aloja heterogeenisempi ja monelta osin käsitteellisesti ”pehmeämpi” toimintaympäristö, ja DOLCE tukee etenkin käsitteellistä moninaisuutta BFO:ta paremmin, kuten käy ilmi sen suunnittelun lähtökohdista (korostus oma):

⁹ <https://finto.fi/fi/>

*DOLCE is a foundational ontology developed with the vision that a **unique universal ontology for knowledge representation cannot exist**. The idea behind DOLCE is that an ontology should be philosophically consistent and transparent (i.e., embrace a clear ontological perspective) and promote its correct application (e.g., by describing explicitly the basic assumptions on which it relies). Furthermore, **DOLCE puts much emphasis on interoperability, in particular with other ontological systems**, and exploits the “no hidden choice” principle: if a philosophical or applicative position is compatible with the explicit commitments of an ontology, then this ontology can indeed be extended to formalize that position. **DOLCE goes even further in this view by allowing coexistence of alternative ontological views via parametrization and other formal techniques.** (Staab & Studer, 2009)*

Projektin aikana DOLCE:sta valittiin käyttöön kevyt ”kansantajuistettu” variantti DOLCE+D&S Ultralite¹⁰, joka sekä piilottaa soveltajalta standardoitavan DOLCE:n tarpeettoman korkeaa formalismin ja abstraktion tasoa, ja tarjoaa DOLCE:n oheen modulaarisesti neljä muuta yläontologiaa useiden eri mallintamisperspektiivien tukemiseen. Versio sisältää myös runsaasti käytössä opastavia annotointeja.



Kuva 3. DOLCE:n sekä sen keskeisten luokkien väliset yhteydet. Vibreät luokat ovat D&S-ontologiasta, punaiset DOLCE:sta ja keltaiset Plan Ontology -ontologiasta. Kuva sovellettu lähteestä (Presutti & Gangemi, 2016).

¹⁰ Versio on kehitetty Italian kansallisen tutkimuskomitean (Il Consiglio Nazionale delle Ricerche) alaisen sovelletun ontologian laboratoriossa (The Laboratory for Applied Ontology).

Alla kuvataan tiiviisti ja mahdollisimman selkokielisesti DOLCE+D&S Ultralite -yläontologian tarjoamat mallinnusrakenteet sekä niiden käyttökohteet esimerkinomaisesti. Varsinaisia kielitieteeseen, logiikkaan tai filosofiaan pohjautuvia formaaleja perusteita ei tässä yhteydessä esitetä.

Itse DOLCE:n keskeisimmät luokat ovat oliot, tapahtumat ja ominaisuudet. Näistä ominaisuudet voivat esiintyä vain osana olioita tai tapahtumia. Jako on kuvattu YSO:ssa¹¹ näin:

Oliot

Olemassa oleva yhtenäinen kokonaisuus. Olioilla on useimmiten suhteellisen selvät spatiaaliset, mutta epäselvät temporaaliset rajat. Oliot ovat olemassa tiettynä ajanhetkenä kokonaisuudessaan. Useampi konkreettinen olio ei voi yleensä olla samaan aikaan samassa paikassa. Oliot osallistuvat tapahtumiin ja toimintaan.

Tapahtumat

Luokkaan kuuluvista entiteeteistä sanotaan, että ne tapahtuvat tai ilmaantuvat. Niillä on usein suhteellisen selvät temporaaliset, mutta epäselvät spatiaaliset rajat. Tietty tapahtuma on vain osittain läsnä tiettynä ajanhetkenä, esim. Ranskan vallankumous ei ollut kokonaisuutena läsnä Pallohuoneen valan hetkenä. Voidaan myös sanoa, että tapahtuman tietynhetkinen osa voi olla itsenäinen tapahtuma. Luokan entiteeteillä sanotaan usein olevan syy- ja/tai seuraussuhde, toisin kuin olioilla. Useampi tämän luokan entiteetti voi tapahtua samassa paikassa samaan aikaan. Tapahtumiin ja toimintaan osallistuvia entiteettejä kutsutaan Olioiksi.

Ominaisuudet

Ominaisuudet luonnehtivat olioita ja tapahtumia. Ominaisuudet eivät ole olemassa itsenäisinä, vaan ne ”realisoituvat” muissa entiteeteissä. Esimerkkejä ominaisuuksista ovat muoto, paino, pituus, väri. Ominaisuuden instanssi (esim. tietty punaisuuden sävy) voi olla läsnä useassa objektissa samanaikaisesti.

Tämän jaon lisäksi DOLCE sisältää myös alla kuvatut yläontologiat:

3.1.1 Tilanteet ja kuvaukset

Descriptions and Situations (D&S) -ontologia on yksi ilmaisuvoimaisimmista ja myös käsitteellisesti monimutkaisimmista DOLCE:n laajennuksista. D&S tarjoaa mahdollisuuden erottaa tietty asioiden tunnistettu tila tai *tilanne* ja tietystä perspektiivistä tehty *kuvauk*s tai tulkinta siitä (Gangemi & Mika, 2003). Lääketieteellinen diagnoosi muodostaa mahdollisen kuvauksen (tulkinnan) potilaan fysiologisesta tilasta. Harjaterästä, sementtiä ja muottilautoja ym. tietyllä tavoin yhdistävä toiminta muodostaa tilanteen, jonka voidaan katsoa tähtäävän tietyn rakentamista koskevan suunnitelman (kuvauk) täyttämiseen (engl. *satisfy*). Samoin tietyn elimistön tilan voidaan katsoa ilmentävän tiettyä sairauden diagnoosia (kuvauk).

DOLCE:n rakennetta kuvaavassa kaaviossa (Kuva 3) havainnollistetaan, että tilanteet (*Situation*) luo kontekstin (*isSettingFor*) olioille, tapahtumille ja erilaisille abstrakteille arvokokonaisuuksille, joita parametrit, tehtävät ja roolit luokittelevat. Tilanne vastaa tai täyttää (*satisfies*) kuvauksen (*Description*), joka taas voi määritellä parametreja, tehtäviä ja rooleja. Kuvauk siis ikään kuin antaa

¹¹ <https://finto.fi/ysso/fi/>

kontekstin (tulkinnan) jollekin tilanteelle ja siihen liittyville entiteeteille. Rakenne mahdollistaa sen, että sama asioiden tila voi kuvata eri toimijoiden näkökulmasta eri tilanteita.

3.1.2 Suunnitelmat

Ontology of Plans -ontologia tarjoaa käyttöön elementit suunnitelmien ja niiden toteutuksen mallintamiseen edellä kuvatun D&S:n varassa (Oberle ym., 2006). Ontologia määrittelee ja hyödyntää osin muiden ontologioiden määrittelemiä keskeisiä luokkia *Plan*, *Plan Execution*, *Goal*, *Task*, *Role*, sekä *Parameter* (ks. Kuva 3 luokkien välisistä suhteista).

3.1.3 Tietokohteet

Information Objects -ontologia tarjoaa mahdollisuuden kuvata tietueita, sekä niiden ilmentymiä D&S:n varassa (Behrendt ym., 2005). Tämä tapahtuu kahden keskeisen luokan avulla: *InformationObject* kuvaa toteutustavasta riippumatonta oliota, *InformationRealization* taas sen toteutusta tai ilmentymää. Asemakaava hyväksyttynä maankäyttösuunnitelmana on *InformationObject*, sen ilmentymä esim. tietomallin mukaisena sanomana tai paperimuodossa on *InformationRealization*. Valitusta perspektiivistä riippuen asemakaavan ilmentymäksi voitaisiin myös laskea sen mukaisesti rakennettu alue.

3.1.4 Kokoelmat

Collection Ontology -ontologian tarkoituksena on mahdollistaa yksien ja samojen entiteettien luokittelu D&S:n varassa modulaarisesti ja monin eri tavoin tilanteesta riippuen (Bottazzi ym., 2006).

3.2 Mallinnusperiaatteet

Pelkkä joustava kehikko ei kuitenkaan riitä, vaan mallintamiseen tarvitaan myös periaatteita, joilla uusia määrittelyjä tuotetaan ja vanhoja harmonisoidaan johdonmukaisesti. Näitä kutsutaan ohjelmistokehityksen suunnittelumallien (engl. Software Design Pattern) mukaisesti ontologisiksi suunnittelumalleiksi (engl. Ontology Design Pattern, ODP).

Mallin tarkoituksena on tarjota yleinen, uudelleenkäytettävä ratkaisu usein toistuvaan mallinnusongelmaan. Vaatimusmäärittelyn mukaisesti ontologioiden kehittämiseen saatavilla olevista suunnittelumalleista tehtiin kevyt kartoitus (Gangemi, Borgo, ym., 2004; Hoekstra, 2009; Janowicz, 2016; Presutti & Gangemi, 2016; Staab & Studer, 2009).

Suuri osa RYYO:n määrittelyssä tarvittavasta perustasoisesta luokittelusta pystyttiin ratkaisemaan DOLCE:n perusrakenteen puitteissa (onko määrittelyn kohteena toiminta, ominaisuus, fyysinen tai käsitteellinen olio). Osa käytetyistä käsitteistä on kuitenkin määritelty lähdeaineistoissa hyvin laveasti tai puutteellisesti, tai samaa termiä käytetään monen toisistaan eroavan käsitteen nimeämiseen (polysemia) – kummassakin tapauksessa yhden täsmällisen kokoavan käsitteen määrittely sekä luokittelu DOLCE:n rakenteeseen on useimmiten mahdotonta. Näissä tapauksissa työryhmä on hyödyntänyt metodia, jossa jokin konkreettinen (alakohtainen) esimerkki pyritään ilmaisemaan yleistetyssä ja kontekstistaan vapaassa (engl. instance-free) muodossa (Blomqvist ym., 2016). Tällä tavoin voidaan esimerkiksi pyrkiä selvittämään, missä määrin infra- ja talonrakennuspuolen käyttämät käsitteet kuvaavat samoja rakenteita.

Muita hyödynnettyjä keskeisiä malleja olivat DOLCE:a varten kuvatut osallistumista, ominaisuuksia, tehtäviä, rooleja, jäsenyyttä, osa-kokonaisuutta, sijaintia, järjestystä, kuvailua ym. koskevat ODP:t (Presutti & Gangemi, 2016).

Käytetyt keskeisimmät mallit on kuvattu alla:

- Entiteettien (toimijoiden ja toiminnan kohteiden) osallisuus tapahtumiin: `Object` -luokan aliluokka tai instanssi yhdistetään `Event` -luokan aliluokkaan tai instanssiin `participatesIn` -assosiaatiolla tai sen aliassosiaatiolla.
- Entiteettien ominaisuuksien ilmaisu: `Quality` -luokan aliluokka tai instanssi yhdistetään `hasQuality` -assosiaatiolla tai sen aliassosiaatiolla ominaisuuden haltijaan (entiteettiin). Kuvattaessa ominaisuuksia, joiden arvot on määritelty jossain itsenäisesti kuvattavassa arvoavaruudessa (metrinen avaruus pituuksien ilmaisuun, väriavaruus, jne.), määritellään arvoavaruus `Region` -luokan aliluokaksi, ja yhdistetään ominaisuuteen `hasRegion` -assosiaatiolla tai sen aliassosiaatiolla.
- Entiteettien kategorisointi eri luokittelutavoin käsitteillä, joihin voidaan liittää luokittelua koskevia muita tietoja: `Concept` -luokan aliluokka tai instanssi yhdistetään `classifies` -assosiaatiolla tai sen aliassosiaatiolla luokiteltavaan entiteettiin. Tarkennettuja luokittelutapoja tarjoavat `Role` -luokka entiteettien roolien ilmaisemiseksi tietyissä tilanteissa, `Parameter` -luokka mm. kvalitatiivisten ominaisuuksien arvojen ilmaisuun tilanteessa, jossa itse arvot ovat käsitteitä, joilla on merkityssisältö (esim. ”korkea”, ”arvokas”), ja `Task` -luokka toiminnan luokitteluun niiden tavoitteena olevan päämäärän mukaisesti.
- Entiteettien käsitteellinen (summittainen) sijainti: `Place` -luokan aliluokka tai instanssi yhdistetään `isLocationOf` -assosiaatiolla tai sen aliassosiaatiolla entiteettiin, jonka sijainti halutaan ilmaista. `Place` -luokan määrittelemät sijainnit ovat paikkoja, alueita tai tiloja, joilla ei välttämättä ole tarkasti määriteltyjä rajoja, ja joiden määrittely ja tunnistaminen tapahtuu niiden olemuksen ja/tai niissä tapahtuvan toiminnan perusteella (mm. erilaiset maisemat, ympäristöt).
- Entiteettien jakautuminen ne muodostaviin osiin: `hasPart` -assosiaatiolla tai sen aliassosiaatiolla yhdistetään koostuva ja koostavat entiteetit toisiinsa. Jos halutaan ilmaista eitransitiivisiä koostumussuhteita, käytetään `hasComponent` -assosiaatiota tai sen aliassosiaatioita.
- Entiteetin sisällön ja toteutuksen välisen suhteen ilmaisu: `InformationObject` -luokan aliluokka tai instanssi ilmaisee jonkin tietueen, joka voidaan realisoida (`isRealisedBy` -assosiaatio tai sen aliassosiaatio) eri tavoin. Esimerkiksi asiakirja on määritelmä tietystä tietosisällöstä ja sen muodosta ja riippumaton siitä, onko asiakirja realisoitu paperimuotoisena dokumenttina vai esimerkiksi XML-sanomana.

3.3 Lähtöaineiston analyysi

RYYO:n sisältöä tuotettaessa käytiin läpi suuri määrä määritelmiä sisältäviä valtiohallinnon julkaisuja, RT-kortteja, standardeja, sanastoja ja aiempien määrittelyprojektien tuloksia. Näiden lisäksi käytiin läpi Y-alustalle jo tuotettujen tietomallien sisältö ja niiden väliset suhteet.

Analyysin tavoitteena oli pyrkiä sekä kvalitatiivisin että kvantitatiivisin menetelmin rajaamaan RYYO:aan kuuluvat keskeisimmät määritelmät. Työryhmän sisäisiin sekä sidosryhmien kanssa toteutettuun työpajaan pohjaten taulukoitiin noin 150 käsitettä käsiteltäväksi.

Toinen joukko valikoitiin Y-alustan tietalueilta: asuminen, kulttuuri, liikenne, luonnonvarat, eläimet ja kasvit, paikkatieto, rakennettu ympäristö, yleiset tieto- ja hallintopalvelut, sekä ympäristö. Valituiksi tuli 65 tietomallia, joista 7 oli merkitty käytöstä poistetuiksi tai korvatuiksi ja täten rajattiin pois. Otokseen rajautuneista malleista 36 on soveltamisprofiileja ja 19 tietokomponenttikirjastoja. Malleista hyödynnettiin analyysissa versioita, jotka olivat saatavilla lokakuussa 2022.

Malleja on luotu välillä 2/2016–9/2022 ja päivitetty välillä 8/2019–10/2022. Tarkempaa analyysia mallien ikä- ja muutosjakaumasta ei tässä tehdä, mutta on huomioitava, että ikähaitari on laaja, eikä yhteentoimivuusmenetelmän soveltaminen ole ollut mallien välillä identtistä. Tällä on osaltaan vaikutusta analyysin lopputulokseen.

Malleissa on huomattava määrä aineistoa: yhteensä noin 139000 kolmikkoo, joista merkittävä osa metadattaa. Analyysimenetelmänä on käytetty mm. nk. keskeisyysmetriikkaa (engl. centrality metric), joka mittaa graafimuotoisen aineiston entiteettien “tärkeyttä”, joka riippuu niitä yhdistävien assosiaatioiden määrästä.

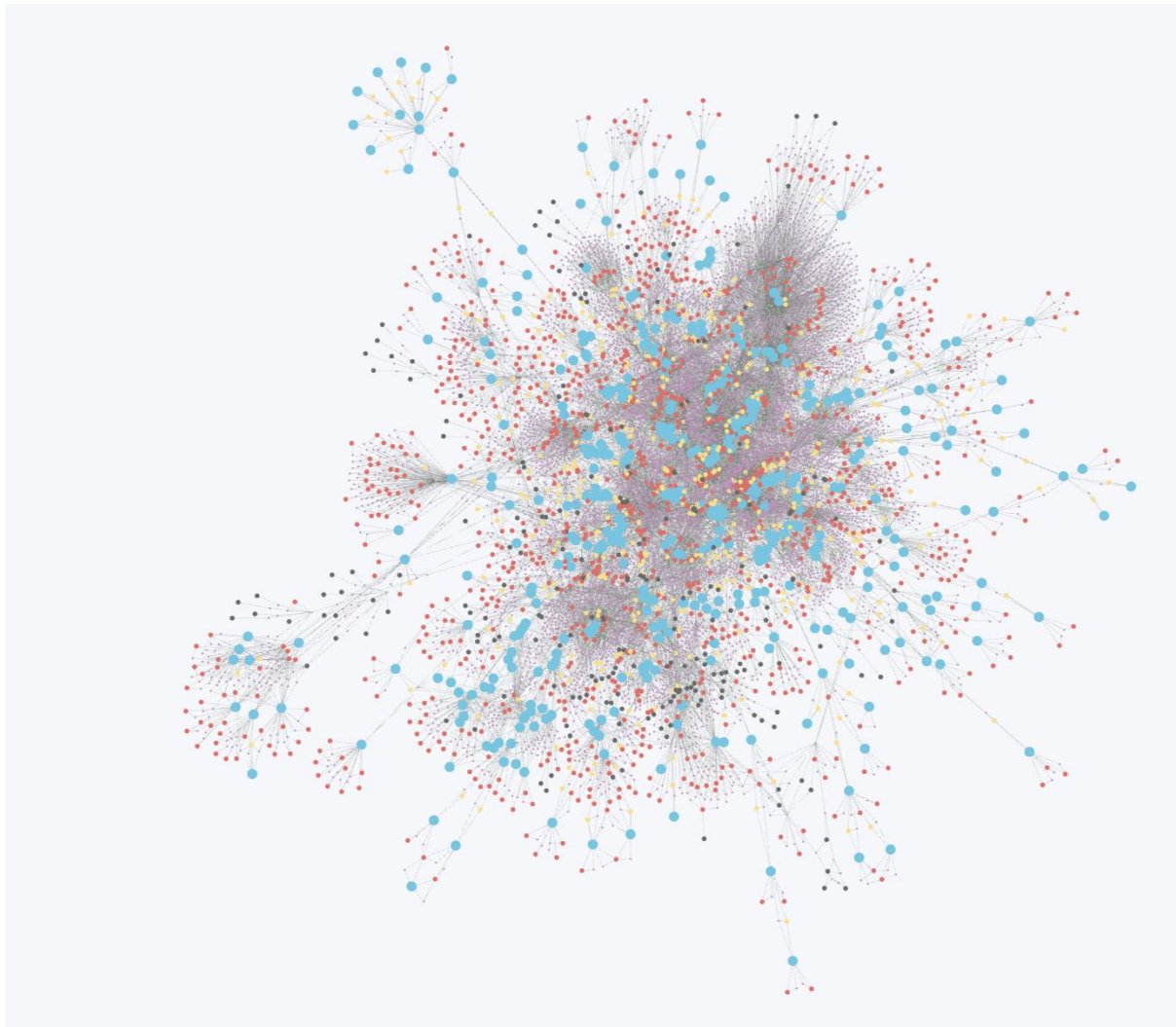
Yksi tässä analyysissa käytetty mitta on välillisyysskeskeisyys (engl. betweenness centrality), joka nimensä mukaisesti mittaa, kuinka tärkeässä yhdistävässä roolissa kukin entiteetti on, ts. kuinka suuri osa graafin muiden entiteettien välisistä yhteyksistä on siitä riippuvaisia. Esimerkiksi jos kahta graafitietorakennetta yhdistää vain yksi molemmissa käytetty resurssi (esimerkiksi `jhs:Nimi`), on kyseisellä resurssilla erittäin korkea välillisyyysluku, sillä ilman kyseistä resurssia rakenteet eivät olisi lainkaan yhteydessä toisiinsa.

Toinen käytetty mitta on ominaisvektorikeskeisyys (engl. eigenvector centrality), joka antaa entiteetille sitä suuremman arvon, mitä enemmän sillä on yhteyksiä muihin entiteetteihin joilla on lukuisia yhteyksiä. Laskentapa nostaa siis esiin sellaisia entiteettejä jotka eivät ole vain itsessään hyvin kytkeytyneitä, vaan ovat hyvin kytkeytyneitä toisiin hyvin kytkeytyneisiin entiteetteihin.

Nämä kaksi mittaa täydentävät toisiaan nostaen YTA-tietomallien aineistosta esiin entiteettejä eri painotuksilla. Analyysissa on tutkittu myös sisältöjen klusteroitumista eli aineiston jakautumista tietyin ehdoin itsenäisiin toisistaan irrallisiin kokonaisuuksiin.

Analyysi on toteutettu pääosin Neo4j-graafialustalla, jolle analysoitava aineisto viedään Pythonilla kirjoitetulla ad-hoc ETL-työkalulla. Työkalu lataa YTA-rajapinnan kautta halutut mallit, sanitoi ne poistamalla ja yhdistämällä aineistoa, ja lataa Neo4j-rajapinnan kautta sen Neo4j-graafitietokantaan. Loppu analyysista on tehty Cypher-kyselylausekkeilla Neo4j-alustalla GDS (Graph Data Science) ja APOC (Awesome Procedure On Cypher) -kirjastojen avulla. Lisäksi analyysissa on hyödynnetty GraphDB-graafialustaa, joka tukee natiivisti RDF-aineistoa.

Analyysin lopputuloksena luokiteltiin noin 8700 resurssia, joista keskeisimmät nostettiin työryhmän taulukkoon. Työryhmän valintojen päällekkäisyydet huomioden lopulliseksi käsiteltäväksi määräksi muodostui 206 käsitettä. RYYO:aan päätyi kuitenkin myös kyseisen talukon ulkopuolisia käsitteitä, jotka havaittiin tarpeellisiksi aiemmin valikoituja käsitteitä lisättäessä.



Kuva 4. Visualisointi Y-alustalta ladatuista resursseista sekä niiden välisistä kytköksistä.

3.4 Hallintamalli

Projektisuunnitelman mukaisesti lopuksi kuvataan ehdotus aiemmin tuotetun rakennetun ympäristön tietokomponenttikirjaston sekä RYYO:n välisestä hallintamallista.

Yksiselitteinen suositus on, että RYYO ja tietokomponenttikirjasto integroidaan yhdeksi rakennetun ympäristön ontologiaksi. Tällöin poistuu tämänhetkinen epäselvä vastuunjako mallien sekä niiden rajausten välillä (mitkä osat kuuluvat yläontologiaan, mitkä tietokomponenttikirjastoon).

Integroidusta ontologiasta tulisi poistaa kaikki rakennetun ympäristön aihepiiriin yksiselitteisesti kuulumattomat määrittelyt (kuten mm. yleiset julkishallinnon tasoiset henkilö-, organisaatio tai prosessikuvaukset), ja viitata niihin vain uudelleenkäytön ja erikoistamisen kautta. Tätä muutosta ei toisaalta kannata tehdä ennen kuin Y-alustan Tietomallit-työkalun työn olevat muutokset on saatu vietyä tuotantoon, ja mallien välisten viittausten tekeminen sekä mallien yleinen hallinta helpottuvat nykyiseen verrattuna merkittävästi.

Tulevaisuudessa mallin päivitysten tulisi tukeutua nykyistä suuremmassa määrin läpinäkyvään ja jäljitettävään dokumentointiin (esimerkiksi Y-alustan tarjoaman mallien versionhallinnan ja

annotoinnin kautta), sekä suunnittelumallien (ODP) hyödyntämiseen sekä sidosryhmäkeskustelujen tukena, että pyrittäessä johdonmukaiseen mallintamisen lopputulokseen. Keskeistä on pyrkiä minimoimaan mallin muokkauksessa epäjohdonmukaisten ratkaisujen kautta syntyvä tekninen velka. (Hoekstra, 2009)

3.4.1 Sisällön päivittäminen

Tietokomponenttikirjaston sisällön osalta hallintamallissa suositellaan käyttöön otettaviksi seuraavia periaatteita:

- Määriteltävän resurssin osalta tulisi selvittää, onko kyseessä polysemia, sillä tiedostamattomana tämä voi muodostaa erityisesti sidosryhmätyössä merkittäviä ongelmia. Polysemian kaikki tunnistetut määritelmät tulisi esitellä mahdollisimman varhain.
- Määriteltävän aihealueen perustana toimivia rakenteita (erityisesti yläontologiatasoisia, kuten ”olio”, ”ominaisuus”, ”arvo”, jne.) ei tule tehdä, sillä näiden rakenteiden määrittely kuuluu pohjana käytetyn yläontologian vastuulle. Määrittelemällä perustasoisia rakenteita uudelleen eri projekteissa, syntyy kilpailevia ristiriitaisia käsitteistyyksiä, joita ei ole aksiomatisoitu johdonmukaisesti.
- Tietokomponenttien mallintamisessa keskeisenä mallintamisen tavoitteena on luokittelu ja identifiointi: kirjaston tarkoituksena on tarjota määrittelyitä, joita soveltamisprofiileissa muodostetut rakenteet hyödyntävät. Soveltamisprofiilien roolina on ainoastaan määrittää tiettyä käyttötapausta varten tarvittavien tietojen rakenne, hyödyntäen tietojen tunnistamisessa yläontologiassa määriteltyjä rakenteita.
- Tietokomponenttien määrittelyn osalta tulisi varmistaa, että komponenttiin liitetty sanaston käsite todella vastaa määritelmältään tietokomponentin rakennetta (esimerkiksi luokan tapauksessa sanaston käsitteen tulisi vastata luokkaan liitettyjen assosiaatioiden ja attribuuttien tulkinnan kautta muodostuvaa käsitystä sen määritelmästä).

3.4.2 Teknisen rakenteen päivittäminen

Tietokomponenttikirjaston teknisen rakenteen osalta hallintamallissa suositellaan käyttöön otettaviksi seuraavia periaatteita:

- Tietokomponenttikirjasto tulisi Y-alustan päivittymisen myötä muokata (refaktoroida) puhtaaksi OWL-ontologiaksi. Tämä tulisi tehdä siksi, että komponenttikirjastoa voitaisiin jatkojalostaa myös Y-alustan ulkopuolella alalla laajasti käytössä olevissa työkaluissa (mm. TopBraid Composer, Protégé).
- Tietokomponenttikirjasto tulisi ISO 21838-3 julkaisun myötä validoida suhteessa kyseisen yläontologian standardoituun versioon.

4 Lähteet

Aarnio, S., Kauppi, M., Tammi, I., Hytönen, M., Rinne, I., Mäkelä, J., & Rainio, A. (2020). *Tulevaisuuden maankäyttöpäätökset*. 244.

Adams, K. M. (2015). Introduction to Non-functional Requirements. Teoksessa K. MacG. Adams (Toim.), *Nonfunctional Requirements in Systems Analysis and Design* (ss. 45–72). Springer International Publishing. https://doi.org/10.1007/978-3-319-18344-2_3

Blomqvist, E., Hammar, K., & Presutti, V. (2016). *Engineering Ontologies with Patterns – The eXtreme Design Methodology*. 28.

Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., & Smith, M. (2012). *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)* (B. Motik, P. F. Patel-Schneider, & B. Parsia, Toim.). World Wide Web Consortium. https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Appendix:_Change_Log_.28Informative.29

Brickley, D., & Guha, R. V. (Toim.). (2014). *RDF Schema 1.1*. World Wide Web Consortium. <https://www.w3.org/TR/rdf-schema/>

Chojka, A. (2020). Considerations about the quality of UML and GML application schemas. *Reports on Geodesy and Geoinformatics*, 109, 17–23. <https://doi.org/10.2478/rgg-2020-0003>

Ciccozzi, F., Malavolta, I., & Selic, B. (2019). Execution of UML models: A systematic review of research and practice. *Software & Systems Modeling*, 18(3), 2313–2360. <https://doi.org/10.1007/s10270-018-0675-4>

Cloutier, R. J., & Hoboken, N. (Toim.). (2022). *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 2.6. SEBoK Editorial Board.

Colomb, R. M. (2007). Quality. Teoksessa *Ontology and the Semantic Web*. IOS Press.

Dick, J., Hull, E., & Jackson, K. (2017). *Requirements Engineering*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-61073-3>

Euroopan komissio. (2017, maaliskuuta 23). *KOMMISSION TIEDONANTO EUROOPAN PARLAMENTILLE, NEUVOSTOLLE, EUROOPAN TALOUS- JA SOSIAALIKOMITEALLE JA ALUEIDEN KOMITEALLE: Eurooppalaiset yhteentoimivuuksperiaatteet – täytäntöönpanostrategia*. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2017:134:FIN>

Euroopan parlamentin ja neuvoston direktiivi 2019/1024, OJ L (2019). <http://data.europa.eu/eli/dir/2019/1024/oj/fin>

Evans, A., Bruel, J.-M., France, R., Lano, K., & Rumpe, B. (2003). *Making UML Precise*.

Frisky & Anjoy & Netum Oy. (2022). *Käynnissä Ryhti-liike. Ryhti-hankkeen kehittävä arviointi. Loppuraportti*. https://api.hankeikkuna.fi/asiakirjat/cf9e2af1-05b9-4a5a-93f6-aaa3efbd7a77/7a31dd8d-9c50-4a49-9d20-071f2e8cc6aa/MUISTIO_20220603091503.PDF

Gangemi, A., Borgo, S., Catenacci, C., & Lehmann, J. (2004). *Task Taxonomies for Knowledge Content* (Deliverable Nro D07; Methodology and Tool Infrastructure for The Creation of Knowledge Units (METOKIS), s. 102).

Gangemi, A., Catenacci, C., & Battaglia, M. (2004). Inflammation ontology design pattern: An exercise in building a core biomedical ontology with descriptions and situations. *Studies in health technology and informatics*, 102, 64–80. <https://doi.org/10.3233/978-1-60750-945-5-64>

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). Theoretical Foundations of Ontologies. Teoksessa *Ontological Engineering* (ss. 1–45). Springer-Verlag. https://doi.org/10.1007/1-85233-840-7_1

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5–6), 907–928. <https://doi.org/10.1006/ijhc.1995.1081>

Herring, J. R. (Toim.). (2020). *Features and geometry – Part 1: Feature models*. Open Geospatial Consortium. <https://docs.ogc.org/as/17-087r13/17-087r13.html>

Hoekstra, R. J. (2009). *Ontology Representation: Design patterns and ontologies that make sense*. IOS Press. <https://hdl.handle.net/11245/1.317612>

Jackson, M. (2009). Some Notes on Models and Modelling. Teoksessa A. T. Borgida, V. K. Chaudhri, P. Giorgini, & E. S. Yu (Toim.), *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos* (ss. 68–81). Springer. https://doi.org/10.1007/978-3-642-02463-4_5

Janowicz, K. (2016). Modeling Ontology Design Patterns with Domain Experts – A View From the Trenches. Teoksessa *Ontology Engineering with Ontology Design Patterns: Foundations and Applications* (s. 11).

Jetlund, K. (2020). A Structure of UML Profiles For Modelling of Geospatial Information In GIS, ITS and BIM. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VI-4/W1-2020, 101–108. <https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-101-2020>

Kiko, K., & Atkinson, C. (2008). *A Detailed Comparison of UML and OWL*. 58.

Knublauch, H., & Kontokostas, D. (Toim.). (2017). *Shapes Constraint Language (SHACL)*. World Wide Web Consortium. <https://www.w3.org/TR/shacl/>

Laki julkisen hallinnon tiedonhallinnasta, Pub. L. No. 906 (2019). <https://www.finlex.fi/fi/laki/ajantasa/2019/20190906>

Leal, D., Cook, A., Partridge, C., Sullivan, J., & West, M. (2020). *A Survey of Industry Data Models and Reference Data Libraries—To identify requirements for, and provide input to, a Foundation Data*

Model. Apollo - University of Cambridge Repository.

https://www.cdbb.cam.ac.uk/files/industry_data_models_and_reference_data_libraries_0.pdf

Miles, A., & Bechhofer, S. (2009). *SKOS Simple Knowledge Organization System Reference*. World Wide Web Consortium. <https://www.w3.org/TR/skos-reference/>

Mäkelä, P., Paavilainen, P., & Andersson, A. (2018). *Rakennusluokitus 2018: Käyttäjän käsikirja*. Tilastokeskus.

Object Management Group. (2014a). *Ontology Definition Metamodel*.

<https://www.omg.org/spec/ODM/1.1>

Object Management Group. (2014b). *Object Constraint Language*.

<https://www.omg.org/spec/OCL/2.4>

Object Management Group. (2015). *XML Metadata Interchange (XMI) Specification*.

<https://www.omg.org/spec/XMI/2.5.1>

Object Management Group. (2017). *OMG® Unified Modeling Language®*.

<https://www.omg.org/spec/UML/2.5.1>

Object Management Group. (2018). *The Distributed Ontology, Modeling, and Specification LanguageTM*.

Object Management Group. (2019). *OMG Systems Modeling Language*.

<https://www.omg.org/spec/SysML/1.6>

Object Management Group. (2021). *Semantics of a Foundational Subset for Executable UML Models (fUML), v1.5*. 432.

Oinonen, K., Vartiainen, K., Nurmio, K., Nieminen, H., Wegelius, J., & Skwarek, J. (2017). *Tulevaisuuden yhteiskunnan johtaminen – Tieto käyttöön, parempia päätöksiä*. 44.

Partridge, C., Mitchell, A., Cook, A., Sullivan, J., & West, M. (2020). *A Survey of Top-Level Ontologies—To inform the ontological choices for a Foundation Data Model*. Apollo - University of Cambridge Repository. <https://doi.org/10.17863/CAM.58311>

Perry, M., & Herring, J. (2012). *OGC GeoSPARQL - A Geographic Query Language for RDF Data*. 75.

Pham, V. C., Radermacher, A., Gérard, S., & Li, S. (2018). A Framework for UML-Based Component-Based Design and Code Generation for Reactive Systems. Teoksessa L. F. Pires, S. Hammoudi, & B. Selic (Toim.), *Model-Driven Engineering and Software Development* (ss. 300–327). Springer International Publishing. https://doi.org/10.1007/978-3-319-94764-8_13

Presutti, V., & Gangemi, A. (2016). Dolce+D&S Ultralite and its main ontology design patterns. *Ontology Engineering with Ontology Design Patterns*, 81–103. <https://doi.org/10.3233/978-1-61499-676-7-81>

Rautiainen, J. (2021). *Ryhti-hankesuunnitelma*. https://api.hankeikkuna.fi/asiakirjat/cf9e2af1-05b9-4a5a-93f6-aaa3efbd7a77/35518d34-60b7-4448-8a16-e88934a6a5b2/KIRJE_20210520070543.PDF

Riekkinen, J. (2022). *Selvitys viranomaisen tiedon hallinnasta ja julkisuudesta rakennetun ympäristön tietojärjestelmästä annettavaa lakia varten* (VN/12032/2022; s. 34).

Rissanen, O.-P., & Lahti, T. (2022). *Valtioneuvoston periaatepäätös tiedon hyödyntämiseksi ja avaamiseksi*.

Seidewitz, E. (2014). UML with meaning: Executable modeling in foundational UML and the Alf action language. *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology - HILT '14*, 61–68. <https://doi.org/10.1145/2663171.2663187>

Shan, L., & Zhu, H. (2008). A Formal Descriptive Semantics of UML. Teoksessa S. Liu, T. Maibaum, & K. Araki (Toim.), *Formal Methods and Software Engineering* (Vsk. 5256, ss. 375–396). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-88194-0_23

Staab, S., & Studer, R. (Toim.). (2009). *Handbook on Ontologies*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-92673-3>

The Open Group. (2020). *O-DEFTM, the Open Data Element Framework, Version 2.0*. <https://publications.opengroup.org/c202>

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(2), 93–136. <https://doi.org/10.1017/S0269888900007797>

Valtioneuvosto. (2022). *Hallituksen esitys eduskunnalle rakentamislaki ja siihen liittyviksi laeiksi*.

Wang, X., Almeida, J. S., & Oliveira, A. L. (2008). Ontology Design Principles and Normalization Techniques in the Web. Teoksessa A. Bairoch, S. Cohen-Boulakia, & C. Froidevaux (Toim.), *Data Integration in the Life Sciences* (ss. 28–43). Springer. https://doi.org/10.1007/978-3-540-69828-9_5