



# Tiedonhallinta ja tietomallinnus kone- netuettavassa muodossa talotekniikan suunnittelusta käyttö- ja ylläpi- tovaiheeseen

Metropolia Ammattikorkeakoulu

Insinööri (YAMK)

Tutkinto-ohjelman nimi

Opinnäytetyö

24.10.2025

## Tiivistelmä

Tekijä:	Markku Vuori
Otsikko:	Tiedonhallinta ja tietomallinnus koneluettavassa muodossa talotekniikan suunnittelusta käyttö- ja ylläpitovaiheeseen
Sivumäärä:	75 sivua + 15 liitettä
Aika:	24.10.2025
Tutkinto:	Insinööri (YAMK)
Tutkinto-ohjelma:	Talotekniikan tutkinto-ohjelma
Ammatillinen pääaine:	Sähköinen talotekniikka
Ohjaajat:	Johtaja Heikki Ihasalo Yliopettaja Rauno Holopainen

---

Rakennusten tietomallit (BIM/IFC) kuvaavat geometrian ja rakenteet, mutta eivät kata ilmanvaihtokoneiden (IV-koneiden) laite- ja ohjaustason semantiikkaa. Tämän YAMK-opinnäytetyön tavoitteena oli kehittää menetelmä, jolla IV-koneen rakenteet ja ominaisuudet voidaan esittää koneluettavassa ja semanttisesti yhteentoimivassa muodossa suunnittelusta ylläpitoon saakka.

Aineistona käytettiin rakennusautomaation mallikaavioita, RAU/LVI-pistelista ja RAVA3PRO-sanastoa. Menetelmässä IV-kone kuvattiin järjestelmätasolla SysML v2 -mallinnuskielellä, josta johdettiin inventaario (CSV), käsitteavastavuuskirjasto (SSSOM) sekä ontologinen tietomalli (RDF/TTL). Kansalliset koodistot (RAU, LVI, RAVA) linkitettiin kansainvälisiin ontologioihin (Brick, BOT, SAREF).

Tuloksena syntyi menetelmä, joka mahdollistaa järjestelmärakenteen ja ontologisen tiedon yhdistämisen vaiheittain. Validoinnissa todettiin, että SysML v2 -malli toimii tehokkaana rakenteellisena lähtökohtana semanttiselle mallinnukselle, ja SHACL/SPARQL-säännöt mahdollistavat tiedon eheyden automaattisen tarkistamisen. Menetelmä parantaa tiedon jäljitettävyyttä, vähentää manuaalista työtä ja luo perustan rakennusautomaation, suunnittelun ja analytiikan yhteiselle tietopohjalle.

Johtopäätöksenä SysML v2:n ja semanttisten teknologioiden yhdistäminen täydentää IFC:tä ja mahdollistaa talotekniikan tietojen siirrettävyyden, validoinnin ja jatkokohdyntämisen eri elinkaarivaiheissa. Menetelmä tarjoaa skaalautuvan ja avoimiin standardeihin perustuvan lähestymistavan rakennusten digitaalisen kaksosen kehittämiseen.

Avainsanat: digital twin, rakennusautomaatio, graafitietokanta SysML v2, ontologiat, Brick, BOT, SAREF, RDF, SHACL, IV-kone, tietomallinnus

## Abstract

Author: Markku Vuori  
Title: Information management and information modeling in machine-readable form, from the design of building technology to the operation and maintenance phase  
Number of Pages: 75 pages + 15 appendices  
Date: 24 October 2025

Degree: Master of Engineering  
Degree Programme: Building Service Engineering  
Professional Major: Electrical Building Services Engineering  
Supervisors: Heikki Ihasalo, Innovation department leader  
Rauno Holopainen, Principal Lecturer

---

Building Information Models (BIM/IFC) describe geometry and structures but do not capture the semantic details of devices and control systems, such as those in air handling units (AHUs). The objective of this Master's thesis was to develop a method that enables representing the structures and properties of an AHU in a machine-readable and semantically interoperable form, covering all phases from design to operation and maintenance.

The study utilized building automation diagrams, RAU/LVI point lists, and the national RAVA3PRO vocabulary. In the proposed method, the AHU was modeled at the system level using the SysML v2 language, from which an inventory (CSV), a semantic mapping library (SSSOM), and an ontology-based data model (RDF/TTL) were derived. National code systems (RAU, LVI, RAVA) were linked to international ontologies (Brick, BOT, SAREF).

The results indicate that the developed method enables a stepwise integration between structural and semantic data. Validation confirmed that SysML v2 provides an effective structural foundation for semantic modeling, while SHACL and SPARQL rules support automated verification of data consistency. The method improves data traceability, reduces manual work, and creates a unified knowledge base connecting building automation, design, and analytics.

In conclusion, combining SysML v2 with semantic technologies complements the IFC standard and enables data interoperability, validation, and reuse across all lifecycle stages. The method provides a scalable and standards-based approach to developing digital twins for building systems.

Keywords: digital twin, building automation, graph database, SysML v2, ontologies, Brick, BOT, SAREF, RDF, SHACL, air handling unit, building information modeling

# Sisällys

## Lyhenteet

1	Johdanto	1
1.1	Tausta ja tutkimuskysymykset	1
1.2	Työn tavoitteet	3
1.3	Aineisto ja menetelmät	5
2	Talotekniikan suunnittelua ohjaavat asetukset ja ohjeet	7
2.1	Talotekniikan suunnittelun koneluettavuus ja sääntöpohjaiset ohjeet	8
2.2	Rakentamismääräykset ja rakennuslaki (2025)	9
3	Suunnittelussa tuotetun tiedon koneluettavuus ja sen hyödyntäminen elinkaaren aikana	11
3.1	Rakennuksen tietomallit ja niiden hyödyntäminen	12
3.2	Digitaalinen kaksonen	15
3.3	Mallinnuskielet ja metamallit	17
3.4	Tietämysgraafit tiedon tallennusmuotona	19
3.5	Ontologiat rakennetussa ympäristössä	24
3.6	Yhteenveto	25
4	Tietomallien ja nimikkeistöjen semanttinen integraatio	27
4.1	Johdanto prosessin kehittämiseen	27
4.2	Viitekehykset ja käsitteet	28
4.3	Menetelmäprosessin yleiskuva	29
4.4	Työkalut ja ohjelmistot	30
4.5	SysML v2-mallin käsittely ja ontologiapohjaisen käsittemallin muodostaminen.	32
5	Tapaustutkimus: IV-koneen semanttinen mallinnus	35
5.1	Aineisto ja rajaus	35
5.2	Komponenttiluettelo ja alustava käsitevastaavuuskirjasto	36
5.3	Käsitevastaavuuskirjasto ja ontologinen linkitys	37
5.4	Validointi ja laadunvarmistus	39
5.4.1	Rakenteen tarkistus (SHACL ja SPARQL)	39
5.4.2	Merkitysten tarkistus ja semanttinen arviointi	40

5.4.3	Iteratiivinen validointi ja menetelmän kehitys	40
5.4.4	Tekoäly ja asiantuntija yhdessä (GPT-5)	41
5.5	Käyttöönotto ja sovellettavuus	42
5.6	Tulokset ja pohdinta	43
6	Menetelmän sovellettavuus ja hyödynnettävyys	48
6.1	Suunnitteluvaihe	48
6.2	Rakentamisvaihe	50
6.3	Käyttö- ja ylläpitovaihe	51
6.4	Hyödyt eri sidosryhmille	52
6.5	Yhteenveto	52
7	Pohdinta	54
8	Yhteenveto	58
	Lähteet	60

## Liitteet

### Liite 1. BIM–AI-integraation pääsovellukset ja tekoälyteknologiat

- Kuva L1. Pääsovellukset BIM–AI-integraatioissa (Zhang ym. 2022, s. 7).
- Kuva L2. Tekoälyteknologioita BIM-mallien yhteydessä (Zhang ym. 2022, s. 10).

### Liite 2. Menetelmässä käytetyt keskeiset käsitteet ja viitekehukset

### Liite 3. RDF- ja property-graafien vertailu.

### Liite 4. Taulukko 2. Menetelmäprosessin vaiheet, syötteet, työkalut, tuotokset ja hyödyt

### Liite 5. RAU-kaavio ja SysML v2-mallin yleiskuva (kuvat L1–L2)

### Liite 6. SysML v2 IBD -osakaavio (komponentit + portit).

### Liite 7. Inventaario (ahu\_inventory.csv – ote)

### Liite 8. Ehdokkaat (candidates.csv)

### Liite 9. Käsitevastaavuuskirjasto (ahu\_mappings.sssom.tsv)

### Liite 10. AML ja DeepOnto – verifiointitulokset

### Liite 11. RDF/TTL-graafi (ahu\_graph.ttl – ote)

### Liite 12. SHACL-säännöt ja SPARQL-kyselyt (+ tulokset)

### Liite 13. YAML-konfiguraatio ja rajapinta-asetukset

### Liite 14. Suorituskomennot ja käyttöohje (GraphDB, toistettavuus)

### Liite 15. Tekoälyn käyttö opinnäytetyössä

## Lyhenteet

- AADL: *Architecture Analysis & Design Language* – arkkitehtuurin analysoinnin ja suunnittelun kieli, jolla mallinnetaan sulautettujen järjestelmien rakenteita.
- AHU: *Air Handling Unit* – ilmanvaihtokone (IV-kone), joka käsittelee rakennuksen tulo- ja poistoilman.
- AI: *Artificial Intelligence* – tekoäly. Työssä viitataan AI-tekniikoihin BIM-integraatioiden yhteydessä (esim. koneoppiminen, ennakoiva analytiikka).
- API: *Application Programming Interface* – sovellusohjelmointirajapinta, sääntöjen ja määrittelyjen joukko, jonka avulla ohjelmistot keskustelvat keskenään.
- ArchiMate: Enterprise-arkkitehtuurin mallintamiskieli (The Open Groupin standardi), jolla kuvataan organisaation prosesseja, tietoja ja teknologia-arkkitehtuuria yhtenäisesti.
- BEKG: *Built Environment Knowledge Graph* – Rakennetun ympäristön tietämysgraafi, joka kokoaa tutkimustietoa graafimuotoon.
- BIM: *Building Information Modeling* – rakennuksen tietomallinnus; digitaalinen rakennusmalli, joka sisältää geometrian lisäksi tietoa rakennusosista ja ominaisuuksista.
- BOT: *Building Topology Ontology* – rakennuksen topologiaontologia; ontologia, joka mallintaa rakennuksen fyysisen rakenteen, tilahierarkian ja osien väliset sijaintisuhteet.

- Brick: Rakennusautomaation laitteiden ja datapisteiden ontologia (Brick-ontology). Brick tarjoaa standardoidun tavan kuvata esim. anturit, toimilaitteet ja niiden väliset yhteydet. (Lyhenne ei varsinaisesti tule sanoista, "Brick" on ontologian nimi.)
- CI/CD: *Continuous Integration / Continuous Deployment* – jatkuva integraatio/jatkuva käyttöönotto; ohjelmistokehityksen käytäntö, jossa koodi integroidaan tiheästi ja toimitetaan automaattisesti tuotantoon.
- CityGML: *City Geography Markup Language* – kaupungin tietomallien kuvaukseen tarkoitettu XML-pohjainen formaatti, jolla esitetään 3D-kaupunkimalleja ja niihin liittyviä tietoja.
- COBie: *Construction-Operations Building information exchange* – rakentamisen ja ylläpidon aikaisen tiedon siirtoformaatti (yleensä Excel/XML-muotoinen), jolla rakennuksen tilat, laitteet ja huoltotiedot voidaan välittää esimerkiksi suunnittelusta kiinteistön ylläpitoon.
- CSV: *Comma-Separated Values* – pilkkueroteltu tiedostomuoto taulukkotiedolle. CSV-tiedostossa sarakkeiden arvot erotellaan pilkuilla.
- DT: *Digital Twin* – digitaalinen kaksonen; fyysisen järjestelmän virtuaalinen vastine, joka yhdistää suunnittelumallin ja reaaliaikaisen sensordatan.
- EPBD: *Energy Performance of Buildings Directive* – rakennusten energiatehokkuusdirektiivi (EU); asettaa vaatimuksia mm. energiatodistuksille ja olemassa olevien rakennusten seurantaan.
- FTP: *File Transfer Protocol* – tiedostonsiirtoprotokolla, jota käytetään tietojen siirtämiseen verkon yli (perinteinen protokolla esim. palvelimelle tiedostojen lataamiseen).

- FMI: *Functional Mock-up Interface* – standardi rajapinta ja tiedostomuoto erilaisten simulointimallien yhteiskäyttöön (mahdollistaa esim. Modella-mallien liittämisen muihin simulointeihin).
- gbXML: *Green Building XML* – rakennusten energiamallinnuksessa käytetty XML-pohjainen tiedostoformaatti, jolla voidaan siirtää tietoja esim. BIM-mallista energiasimulointiohjelmiin.
- IFC: *Industry Foundation Classes* – rakennusalan tietomallistandardi (ISO-standardisoitu), joka mahdollistaa 3D-mallien ja niihin liittyvien tietojen vaihdon eri ohjelmistojen välillä.
- IoT: *Internet of Things* – esineiden internet; laitteiden ja sensorien verkottuminen internetin yli. Viittaa työssä mm. kentälaitteisiin, jotka tuottavat dataa rakennusautomaatioon.
- JSON: *JavaScript Object Notation* – kevyt tiedonesitysmuoto, joka on ihmisluettava ja koneluettava (yleinen formaatti API-rajapinnoissa ja tietojen siirrossa).
- LBD: *Linked Building Data* – viittaa linkitetyn datan tekniikoihin rakennusalan tietomallien yhteydessä. LBD-palvelin tarkoittaa ratkaisua, jossa BIM-tietomallit altistetaan RDF-graafina linkitetyn datan periaattein.
- LLM: *Large Language Model* – suuri kielimalli, esimerkiksi tekoälymalli (kuten GPT-3/4) joka kykenee luonnollisen kielen käsittelyyn. Työssä viitataan LLM:ään ehdotusten tarkentamisen apuna.
- LVI: Lämpö, vesi ja ilmanvaihto – talotekniikan osa-alue (HVAC vastaava englanniksi). Työssä LVI viittaa myös LVI-suunnittelun nimikkeistöön.
- ORM: *Object-Relational Mapping* – olio-relaatiomuunnos; menetelmä, jossa oliopohjaisen ohjelmiston tietomalli muunnetaan relaatiotietokannan taulumuotoon.

- OWL: *Web Ontology Language* – ontologioiden kuvauskieli (W3C-standardi), jolla voidaan määritellä käsitteitä, niiden suhteita ja loogisia sääntöjä semanttisessa webissä.
- PoC: *Proof of Concept* – konseptin toteennäyttö; kokeellinen toteutus tai prototyyppi, jolla osoitetaan idean toimivuus käytännössä.
- RAU: Rakennusautomaatio (esim. RAU-nimikkeistö viittaa rakennusautomaation urakkarajaukseen ja laitekoodeihin). Työssä RAU viittaa Granlundin tietokannassa käytössä olevaan rakennusautomaation laiteluokitukseen.
- RAVA: Viitataan RAVA3PRO-hankkeeseen; RAVA tulee sanoista “Rakennusvalvonta” tms., käytännössä kyse on valtakunnallisesta talotekniikan nimikkeistö- ja tietosisältöprojektistä. (Lyhenteelle ei ole avattu virallista muotoa julkisesti, mutta asiayhteydessä tarkoitetaan Rakennusvalvonnan kehityshanketta nimikkeistöjen yhtenäistämiseksi.)
- REST: *Representational State Transfer* – verkkopalvelujen arkkitehtuurityyli ja rajapintamalli, jossa resurssit (tiedot tai toiminnot) esitetään tiloja hyödyntäen HTTP-protokollan kautta. REST-rajapinta (REST API) on yleinen tapa toteuttaa web-sovellusrajapintoja.
- RDF: *Resource Description Framework* – linkitetyn datan tietomallirakenne (W3C-standardi), jossa tieto esitetään kolmoisina (subjekti–predikaatti–objekti). Mahdollistaa erilaisten tietolähteiden yhdistämisen semanttisesti.
- SAREF: *Smart Applications REFerence ontology* – IoT-laitteiden ja -toimintojen viiteontologia (kehitetty alun perin EU:ssa kodin älylaitteiden yhteentoimivuuteen), jota voidaan hyödyntää kuvaamaan laitteiden piirteitä yhtenäisesti.

- SEAS: *Smart Energy Aware Systems* – ontologiakokonaisuus/projekti, joka pyrkii mahdollistamaan eri energiajärjestelmien (rakennukset, verkot, tuotanto) semanttisen yhteentoimivuuden. Työssä mainittu yhtenä esimerkkinä laajoista ontologiaviitekehyksistä.
- SKOS: *Simple Knowledge Organization System* – sanastojen ja luokittelujärjestelmien kuvauskieli (W3C-suositus), jolla voidaan esittää käsitteitä, termistöjä ja niiden hierarkioita koneymmärrettävästi.
- SSSOM: *Simple Standard for Sharing Ontology Mappings* – yksinkertainen standardi ontologioiden ja sanastojen käsitevastaavuuksien jakamiseen. Käytännössä TSV-taulukkona esitetty vastaavuusformaatti, jolla voidaan dokumentoida esimerkiksi RAU-käsitteiden ja Brick-luokkien vastaavuudet.
- SHACL: *Shapes Constraint Language* – kieli RDF-tietomallien rakenteellisten sääntöjen ja eheysehtojen määrittelyyn. SHACL-määrittelyjen avulla voidaan validoida, että graafi noudattaa tiettyjä sääntöjä (esim. "IV-koneessa oltava vähintään yksi lämpötila-anturi").
- SPARQL: *SPARQL Protocol and RDF Query Language* – RDF-tietojen kyselykieli (W3C-standardi), jonka avulla voidaan hakea ja yhdistellä tietoa RDF-tietomalleista samaan tapaan kuin SQL relaatiotietokannoista.
- SysML v2: *Systems Modeling Language* – järjestelmämallinnuskieli, laajennus UML-kielestä. Mahdollistaa järjestelmän rakenteen (lohkokaaviot), toiminnan (tilakaaviot, sekvenssikaaviot), vaatimusten ja parametrien kuvaamisen yhdenmukaisesti.
- TKHJ: Tietokannan hallintajärjestelmä – ohjelmisto tiedon tallentamiseen ja hakemiseen (esim. SQL-relaatiotietokanta). Vastaa englanninkielistä lyhennettä DBMS.
- UML: *Unified Modeling Language* – yhtenäinen mallinnuskieli, jota käytetään erityisesti ohjelmistojen ja järjestelmien suunnitteluun (kaavioita

mm. luokkarakenteista, sekvensseistä, tapauksista). SysML v2 on osittain johdettu UML:stä.

- URI: *Uniform Resource Identifier* – yhtenäinen resurssitunniste; merkkijono, joka yksilöi resurssin (kuten tietyn laitteen instanssin graafitietokannassa). RDF-maailmassa jokaisella solmulla on yksilöllinen URI-tunnus.
- XMI: *XML Metadata Interchange* – XML-pohjainen formaatti mallinnustietojen vaihtoon (esim. UML/SysML v2-mallin voi viedä XMI-muodossa yhteensopivuuden varmistamiseksi).
- YTV: Yleiset tietomallivaatimukset – rakennushankkeiden tietomallinnuksen ohjeistus (esim. YTV 2012, päivitetty 2022). Määrittelee, mitä tietoja BIM-malleihin tulee tuottaa suunnittelijoiden toimesta eri suunnittelualoilla.

# 1 Johdanto

## 1.1 Tausta ja tutkimuskysymykset

Rakennusten energiatehokkuus ja sisäympäristön palvelutasot eivät ole pysyviä vaan muuttuvat rakennuksen elinkaaren aikana esimerkiksi käyttötarkoituksen, käyttöaikojen tai tilamuutosten seurauksena. Kiinteistön arvo määräytyy osittain energialuokan perusteella, mikä luo painetta energiatehokkuuden jatkuvalle seurannalle ja parantamiselle (European Union 2024). Käytännössä energiatoistuluokan parantaminen voidaan saavuttaa esimerkiksi lyhentämällä ilmanvaihtokoneiden käyntiaikoja tai säätämällä jäähdytyksen ja lämmityksen tehoa, mikä voi heikentää sisäympäristön laatua. Tämä asettaa talotekniikan suunnittelulle ja ylläpidolle uusia haasteita: palvelutason on säilyttävä, vaikka energiatehokkuutta parannetaan.

Kiinteistöalalla korostuu siirtymä järjestelmäkeskeisyydestä palvelukeskeisyyteen. Tilojen ja rakennusten toiminnallisuus, turvallisuus ja käyttäjäkokemus ovat nousseet tärkeiksi arvoja määrittäviksi tekijöiksi. Uudet toimintamallit voivat sisältää esimerkiksi lämpöolosuhdetakuun, energiankulutusvastuun tai ilmanlaatutakuun. Tämä kehitys lisää tarvetta todentaa rakennusten toimivuus ja taloteknisten järjestelmien palvelutaso entistä tarkemmin.

Käytännön ongelma on, että rakennuksesta kerättävä tieto on edelleen pirstoutunut useisiin järjestelmiin: energiankulutuksen mittauksiin, vedenmittaukseen, rakennusautomaatioon, kulunvalvontaan, valaistukseen ja muihin palveluihin. Standardinomaiset tiedonsiirtorajapinnat ovat yleistyneet (esim. REST-API), mutta ne eivät ratkaise käsitteiden yhteismitallisuuden puutetta. Tämä tekee palveluiden tuotteistamisesta kallista ja hidasta, koska eri järjestelmien rajapinnat on räätälöitävä tapauskohtaisesti (Balaji ym. 2016).

Rakennusten talotekniset järjestelmät ovat myös aina jossain määrin uniikkeja. Jokaisen rakennuksen suunnitelmat, työpiirustukset ja toteutuneet laitteet poikkeavat toisistaan, vaikka ratkaisuja pyritään monistamaan. Usein analysointiin tarvittava järjestelmätieto joudutaan syöttämään käsin analytiikkaohjelmistoihin, koska suunnitelmissa oleva tieto ei ole koneluettavassa muodossa. Tämä nostaa kustannuksia ja tekee analytiikan soveltamisen vaikeaksi (Rasmussen ym. 2018).

Nykyiset ontologiat, kuten Brick ja BOT, tarjoavat yhtenäisiä sisällöllisiä kuvauksia rakennusautomaation laitteista, datapisteistä ja rakennusten tilahierarkiasta. Brick soveltuu hyvin esimerkiksi ilmanvaihtokoneen komponenttien, pisteiden ja ohjaussuhteiden kuvaamiseen, kun taas BOT täydentää sitä rakennuksen tilojen ja topologian mallintamisessa. Kumpikaan ei kuitenkaan yksin riitä koko rakennuksen elinkaaren analytiikan tarpeisiin, minkä vuoksi niiden yhdistäminen ja laajentaminen muihin ontologioihin on välttämätöntä. Tätä osoittavat myös uudet tutkimukset, joissa on kehitetty menetelmiä BIM- ja Brick-mallien integroimiseksi (Vittori ym. 2023) sekä Brickin linkittämiseksi muihin rakennusalan ontologioihin (Garrido-Hidalgo ym. 2023). Lisäksi on kehitetty kokonaisvaltaisia suorituskäyttöontologioita, kuten HBPO, joissa huomioidaan energia-, akustiikka- ja valaistusnäkökulmia (Utkucu ym. 2025). Näiden esimerkkien perusteella voidaan todeta, että eri ontologiat on integroitava, jotta rakennuksen rakenteet, tilat ja toiminnot voidaan kuvata analytiikkaa varten.

Lisäksi rakentamislaki (voimaan 1.1.2025) ja tietomallimuotoinen lupaprosessi (pakollinen 1.1.2026) vahvistavat, että tietomallien käyttö ja yhteismitallisuus ovat jatkossa välttämättömiä (Rakentamislaki 2023; Rakennustieto 2024).

Tämän opinnäytetyön tutkimuskysymykset ovat:

Miten talotekniikan järjestelmärakenteet ja komponentit voidaan kuvata koneluettavassa muodossa niin, että niitä voidaan hyödyntää analytiikassa ja elinkaaren aikaisessa hallinnassa? Tutkimus jakautuu osakysymyksiin:

- Miten talotekniikan järjestelmärakenne (esimerkiksi ilmanvaihtokone) voidaan kuvata SysML v2 v2 -kielellä siten, että suunnittelu- ja ohjaustieto on muokattavissa koneluettavaan, semanttiseen RDF/ontologiamuotoon analytiikkaa ja elinkaaren aikaista hallintaa varten?
- Kuinka SysML v2:n elementit ja niiden väliset suhteet voidaan kytkeä keskeisiin rakennusalan ontologioihin (Brick, BOT, SA-REF sekä kotimaiset sanastot ja luokitukset) siten, että syntyvä tietomalli säilyttää rakenteellisen ja käsitteellisen yhteneväisyyden?
- Millainen vaiheistettu prosessi (esimerkiksi inventaario, vastuu, RDF-muunnos ja validointi) tarvitaan, jotta tiedon yhteensovitus ja muunnos voidaan toteuttaa menetelmätasolla ilman, että kaikkia vaiheita automatisoidaan loppuun asti?
- Millä kriteereillä kehitetyn menetelmän kattavuutta, toistettavuutta ja sovellettavuutta voidaan arvioida, ja miten menetelmä tukee suunnittelu- ja käyttövaiheen tiedon yhteensovitusta myös tulevia digitaalisia palveluja varten?

Näiden yleisten tutkimuskysymysten lisäksi työssä toteutetaan tapaustutkimus ilmanvaihtokoneesta (IV-kone). Erityiskysymys on:

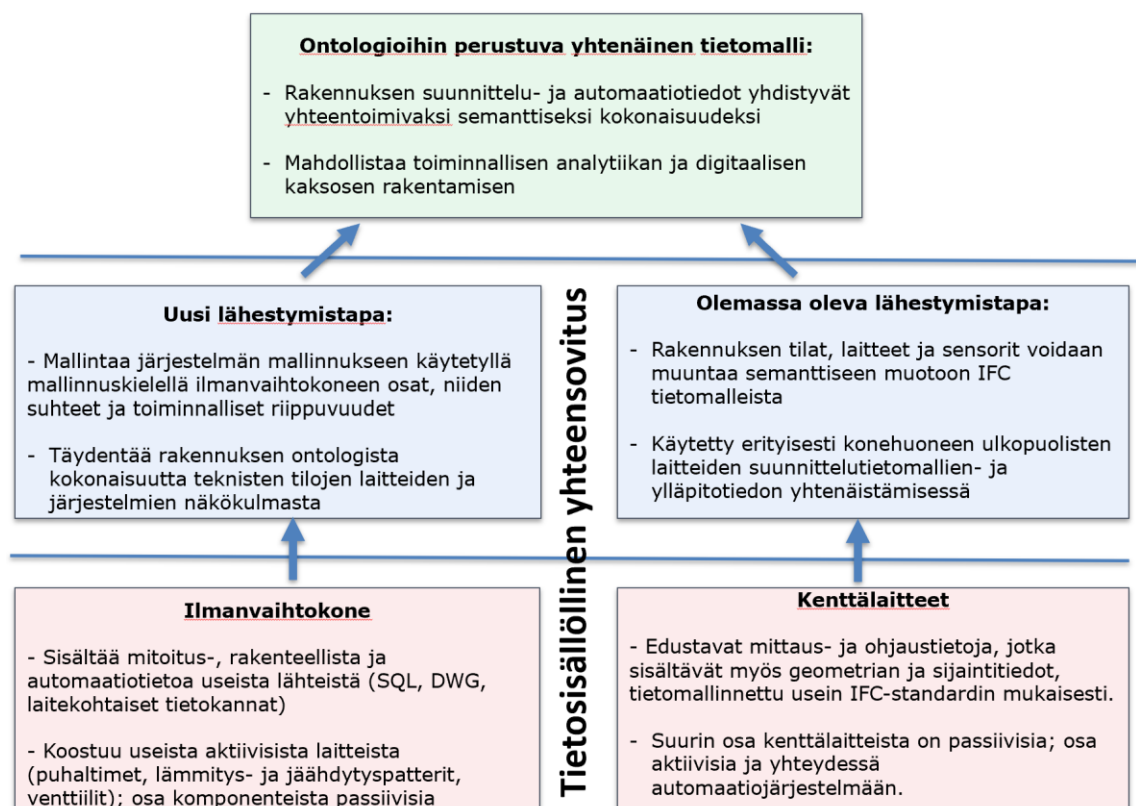
- Miten IV-koneen järjestelmätason semantiikka voidaan kuvata SysML v2:llä ja linkittää ontologioihin siten, että se on toistettavasti validoitavissa ja hyödynnettävissä suunnittelusta ylläpitoon asti?

## 1.2 Työn tavoitteet

Tämän YAMK-opinnäytetyön tavoitteena on kehittää ja kuvata menetelmä, jolla suunnitteluvaiheen IV-koneen järjestelmärakenteista, komponenteista ja niiden suhteista voidaan luoda ontologiapohjainen, koneluettava malli.

Tavoitteena on tietosisällön yhteensovitus, jossa rakennuksen suunnittelu-, ohjaus-, IoT- ja automaatiotieto voidaan liittää osaksi ontologisesti yhtenäistä mallia muiden laitteiden kanssa. Menetelmän toimivuus ja sovellettavuus todennetaan kokeellisella toteutuksella, joka toimii esimerkkinä käytännön sovelluksesta. Järjestelmän mallinnusta voidaan soveltaa muihin talotekniikan järjestelmiin, mutta työ kohdistuu IV-konehuoneen ilmanvaihtokoneen toiminnalliseen kokonaisuuteen.

Kuva 1 esitetään tutkimuksessa käytetty lähestymistapa, jossa ilmanvaihtokonehuoneen suunnittelu- ja automaatiotieto jalostetaan toiminnallisen mallin kautta semanttisesti yhtenäiseksi tietomalliksi. Alimmalla tasolla kuvataan suunnitteluvaiheen rakenteellinen tieto. Keskitasolla esitetään järjestelmämallinnukseen perustuva lähestymistapa ja olemassa oleva käytäntö tietosisällön yhteensovittamiseksi IFC-tiedoista, jota pääosin käsittävät konehuoneiden ulkopuoliset laitteet. Ylimmällä tasolla kuvataan tavoitetila, ontologioihin perustuva yhtenäinen tietomalli, joka mahdollistaa analytiikan ja digitaalisen kaksosen rakentamisen koko järjestelmästä.



Kuva 1. Tutkimuksessa käytetty lähestymistapa IV-konehuoneen tietosisällölliseen yhteensovitukseen.

Työn onnistumista arvioidaan seuraavien kriteerien avulla:

- Menetelmän sisällöllinen kattavuus: Arvioidaan kuinka hyvin kehitetty prosessi kattaa ilmanvaihtokoneen järjestelmärakenteet ja niiden suhteet SysML v2-ontologia-mallinnuksessa.
- Menetelmän toimivuus: Arvioidaan kuinka hyvin SysML v2-pohjainen malli voidaan muuntaa RDF/TTL-muotoon ja validoida SHACL-sääntöjen avulla.
- Työelämäarvo ja sovellettavuus: Arvioidaan miten menetelmä tukee suunnittelu- ja ylläpitovaiheen tiedon yhdistämistä, vähentää manuaalista työtä ja parantaa tiedon jäljitettävyyttä.

### 1.3 Aineisto ja menetelmät

Tämän tutkimuksen aineisto koostuu talotekniikan suunnitteluaineistoista, tietomalleista sekä rakennusautomaation kenttälaitetiedoista. Keskeisenä tarkastelun kohteena ovat ilmanvaihtokoneen (IV-koneen) suunnitelmat. Tapaustutkimus on rajattu yhteen IV-koneeseen, jota hyödynnettiin menetelmän soveltamisen ja arvioinnin lähtökohtana.

Tutkimusmenetelmä perustuu koneluettavan tiedon esittämisen tapojen tarkasteluun ja niiden soveltamiseen talotekniikan järjestelmien mallintamisessa. Tavoitteena on kehittää lähestymistapa, jolla järjestelmien rakenteet, komponentit ja niiden suhteet voidaan esittää semanttisesti yhteneväisessä muodossa analytiikkaa varten. Tutkimuksessa arvioidaan erityisesti, miten olemassa olevia mallinnuskieliä ja standardeja voidaan hyödyntää tässä tehtävässä.

Yhtenä lähtökohtana tarkastellaan IFC-tiedostoja, jotka ovat rakennusalan vakiintunut tiedonsiirtostandardi, mutta tutkimus laajennetaan myös muihin järjestelmätason mallinnuskieliin ja semanttisiin lähestymistapoihin. Lopullinen mallinnuskielen valinta perustellaan kirjallisuuskatsauksen perusteella (luvut 2–3).

Opinnäytetyöntekijä tuo tähän lisäarvon tarkastelemalla, miten semanttinen harmonisointi mahdollistaa IV-koneen datan kytkemisen osaksi digitaalista kaksosta, jolloin voidaan osoittaa menetelmän soveltuvuus konkreettisiin kohteisiin.

Menetelmän kehittämisessä yhdistetään kolme toisiaan täydentävää lähestymistapaa:

- Kirjallisuuskatsaus – analysoitiin aiempia menetelmiä (esim. IFC–RDF-muunnokset, ontologiat, mallinnuskielet) ja arvioitiin niiden soveltuvuutta talotekniikan järjestelmien kuvaamiseen ja analysointiin.
- Tapaustudkimukset – testattiin käytännössä, miten kehitetty menetelmä toimii ilmanvaihtokoneen (IV-koneen) mallintamisessa ja semanttisessa muunnoksessa.
- Menetelmäkehitys ja validointi – kehitettiin vaiheistettu prosessi, jossa SysML v2-malli muunnettiin RDF/ontologiamuotoon ja validoitiin SHACL-sääntöjen avulla. Prosessi sekä sen osat — SysML v2-mallit, käsitevastaavuuskirjasto ja käytetyt ontologiat — suunniteltiin keskenään yhteensopiviksi ja tukemaan myöhempiä pilvipalvelu- ja rajapintatoteutuksia.

## 2 Talotekniikan suunnittelua ohjaavat asetukset ja ohjeet

Talotekniikan suunnittelua ohjaavat Suomessa useat eri säädökset, standardit ja ohjeet. Tämän opinnäytetyön näkökulmasta keskeisimmiksi lähteiksi on valittu rakentamislaki ja rakentamismääräykset, tekniset standardit (SFS, Eurokoodit), Rakennustiedon ohjekortit ja YTV 2022 -ohjeet, tietomallit.suomi.fi-portaali sekä suurimpien kaupunkien ja keskeisten tilaajaorganisaatioiden, kuten Senaatti-kiinteistöjen, ohjeistukset. Valinta perustuu siihen, että nämä lähteet muodostavat yhdessä kolmitasoisen kokonaisuuden: lakisääteisen perustan, tekniset mitoitus- ja suunnitteluohjeet sekä tilaajien ja viranomaisten konkreettiset vaatimukset. Näin voidaan varmistaa, että suunnittelussa tuotettu tieto on yhteismitallista, koneluettavaa ja hyödynnettävissä koko rakennuksen elinkaaren ajan.

Keskeiset lähteet ja dokumentit ovat:

- Rakentamislaki ja rakentamismääräykset, jotka sisältävät talotekniikan kannalta olennaisia säädöksiä energiatehokkuudesta, paloturvallisuudesta ja sisäilmasta (Rakentamislaki 2023: §33–34; §37–38)
- SFS-standardit ja Eurokoodit, joissa määritellään teknisiä ratkaisuja ja mitoitusperusteita .
- Rakennustiedon RT-kortit ja YTV 2022 -ohjeet, jotka ohjaavat käytännön suunnittelua ja tietomallien käyttöä(KIRAHub 2021: 1–4 ).
- Tietomallit.suomi.fi -portaali, joka kokoaa yhteen valtakunnallisia tietomalleja (esim. 3Dkunta, alueidenkäytön rajoitusten tietomalli) ja tarjoaa perustan yhteen toimivuudelle.
- RAVA3PRO-hankkeen määrittelyt, joissa on kehitetty kansallinen talotekniikan nimikkeistö ja tietosisältöjen vakiointi. Tämä työ tukee erityisesti rakennuslupaprosessin digitalisointia ja varmistaa, että tietomalleissa olevat talotekniset tiedot ovat yhteismitallisia ja hyödynnettävissä elinkaaren eri vaiheissa (Luoma 2023: 60–65).
- Suurimpien kaupunkien ohjeet (esim. Helsingin tietomalliohje 2024, Espoon rakennusvalvonnan ohjeet) sekä Senaatti-kiinteistöjen ylläpidon tietomalliohjeet, jotka määrittelevät konkreettisia tietosisältöjä suunnittelu- ja ylläpitomalleille.

Yhdessä nämä lähteet kattavat sekä normatiiviset vaatimukset, tekniset standardit että käytännön soveltamisen, ja ne on valittu, koska juuri tämä kokonaisuus mahdollistaa tutkimuskysymysten kannalta keskeisen näkökulman: tiedon yhteismitallisuuden, koneluettavuuden ja semanttisen yhdenmukaisuuden varmistamisen koko rakennuksen elinkaaren ajan.

## 2.1 Talotekniikan suunnittelun koneluettavuus ja sääntöpohjaiset ohjeet

Koneluettavuus tarkoittaa sitä, että suunnitelmien tietosisältö on esitetty sellaisessa muodossa, että tietoa voidaan käsitellä automaattisesti ohjelmistojen avulla. Pelkkä 3D-geometria ei riitä, vaan mallista on käytävä ilmi myös järjestelmätiedot, laitteiden ja pisteiden suhteet sekä tilasijoittelu.

Rakentamislaki tuo vaatimuksen, että lupaprosessissa on käytettävä koneluettavia tietomalleja, mikä tekee asiasta keskeisen myös talotekniikan suunnittelussa (Rakentamislaki 2023: 60–61 §).

Rakennustarkastusyhdistyksen (2024:1–6) mukaan tietomallien hyödyntäminen vähähiilisessä rakentamisessa perustuu siihen, että mallien tietosisältö on riittävän jäsenneltyä, jotta voidaan automatisoida esimerkiksi hiilijalanjälkilaskenta ja energiatehokkuuden arviointi.

Opinnäytetyöt ja tutkimukset tukevat tätä näkemystä. (Kolari (2021: 45–52) toteaa, että rakenteellisten IFC-mallien tietosisällön vakiointi on välttämätöntä, jotta suunnitelmat voidaan tarkistaa automaattisesti ja viranomaisten käsittely nopeutuu. (Vesalainen (2017: 37–40) osoittaa diplomityössään, että talotekniikan nimikkeistöjen yhdenmukaisuus parantaa suunnitelmien siirrettävyyttä ja mahdollistaa tietojen hyödyntämisen myös ylläpitovaiheessa.

Tämän työn kannalta koneluettavuus ymmärretään siten, että tieto kuvataan linkitettyinä datana (RDF), laitteet ja pisteet merkitään ontologioilla (Brick, BOT, tarvittaessa Haystack/RAVA), ja mallin rakennetta validoidaan sääntöpohjaisesti

(esim. SHACL-validoinnilla). Näin koneluettavuus ei ole pelkkä tekninen lisäominaisuus, vaan se on perusta, jonka avulla suunnittelu voidaan liittää osaksi viranomaisprosesseja, elinkaaren aikaista tiedonhallintaa ja rakennuksen digitalisaatiota.

## 2.2 Rakentamismääräykset ja rakennuslaki (2025)

Rakentamismääräykset sisältävät talotekniikan suunnittelun kannalta olennaisia ohjeita mm. ilmanvaihdosta, veden laadusta, paloturvallisuudesta ja energiatehokkuudesta. Lisäksi korjausrakentamista koskevat asetukset ohjaavat suunnittelua vanhoissa rakennuksissa (Eksergia 2013: 12–20).

Vuonna 2025 voimaan tullut rakentamislaki uudistaa merkittävästi lupaprosessia. Laki tuo yhden yhtenäisen rakennusluvan ja velvoittaa huomioimaan hiilijalanjälkilaskennan sekä tietomallimuotoisen lupaprosessin, joka on pakollinen 1.1.2026 alkaen (Rakentamislaki 2023: 38 §; 42 §; 60–61 §). Rakennustiedon (2024) mukaan tämä tarkoittaa, että tietomalliin on sisällytettävä järjestelmätiedot ja materiaalien elinkaaritietoja, jotta lupahakemukset onnistuvat.

Tocomanin (2024: 5–8) raportin mukaan pelkkä 3D-geometriamalli ei enää riitä, vaan lupaprosessissa vaaditaan koneluettavaa ja semanttisesti määriteltyä tietoa mm. taloteknisistä järjestelmistä ja energiankulutuksesta. Tämä edellyttää suunnittelijoilta ja ohjelmistotoimittajilta uusia toimintamalleja.

Luoma (2023: 60–65) osoittaa diplomityössään, että RAVA3PRO-hanke määrittää tietosisältöjä, joilla rakennuslupaprosessi voidaan vakioida. Hänen mukaansa yhtenäiset tietomallivaatimukset ovat välttämättömiä, jotta lupaprosessin digitalisointi onnistuu käytännössä. Samoin Saari (2023: 33–40) esittää opinnäytetyössään, että tietomallipohjainen lupaprosessi voi nopeuttaa erityisesti korjausrakentamisen lupahakemuksia ja parantaa suunnitelmien laatua.

VTT:n tutkimuksessa Lavikka & Kallinen (2024: 12–19) kuvaavat, että Suomen siirtymäpolku BIM-pohjaisiin lupaprosesseihin riippuu teknologisten ratkaisujen

lisäksi kuntien osaamisesta ja resursseista Solibri (2024: 2–6) mukaan tietomallien käyttö lupaprosessissa on jo helpottanut automaattista sääntöjen tarkistusta käytännön hankkeissa.

Yhteenvetona voidaan todeta, että rakentamislaki vahvistaa tietomallien asemaa keskeisenä dokumenttina talotekniikan suunnittelussa. Sen myötä suunnittelijoiden on hallittava sekä tekniset standardit että sisällöllisesti rikastettu tietomallipohjainen raportointi. Tämä työ vastaa vaatimukseen kehittämällä menetelmän, jossa järjestelmätiedot kuvataan ontologisesti määriteltynä graafeina ja tarkastetaan sääntöpohjaisesti.

### 3 Suunnittelussa tuotetun tiedon koneluettavuus ja sen hyödyntäminen elinkaaren aikana

Koneluettavan tiedon hyödyntäminen ei yksin riitä varmistamaan rakennusten teknisten järjestelmien tehokasta hallintaa. Tieto voi olla muodollisesti oikeassa formaatissa, mutta sen merkitys jää usein epäselväksi, jos eri järjestelmät tulkitsevat samoja käsitteitä eri tavoin. Käytännössä tämä merkitsee, että esimerkiksi sama ilmanvaihtokoneen puhallin voidaan kuvata eri tavoin eri tietomalleissa tai dokumentaatioissa, mikä johtaa ylimääräiseen käsityöhön metatietojen yhdistämisessä.

Useat tutkimukset ovat osoittaneet, että pelkkä koneluettavuus ei takaa yhteentoimivuutta ilman semanttista vakiointia ja ontologioiden käyttöä. Carlos ym. (2025: 185–186) korostavat, että eri toimijoiden käyttämät käsitteet ja ominaisuudet poikkeavat toisistaan, mikä vaikeuttaa automaattista tietojen yhdistämistä. Tan ym. (2024: 14–16) osoittavat, että valmistajien tuotetietokannat ja BIM-ympäristöt kärsivät metatietojen nimeämisen epäyhtenäisyydestä, mikä heikentää tiedon siirrettävyyttä. Vastaavasti Abanda ym. (2025: 7–9) ehdottavat BIM-OIM-ontologiaa ratkaisuksi siihen, että eri projektivaiheiden tiedot voidaan esittää yhdenmukaisesti.

Myös rakennusten energiasimuloinnissa on havaittu, että ilman semanttisia malleja eri lähteistä peräisin olevan datan integraatio johtaa epäyhtenäisyyksiin ja rajoittaa automaattisen analytiikan luotettavuutta (Aniakor ym. 2024: 3–5). Tämän ongelman ratkaisemiseksi on välttämätöntä, että suunnitelmien ja mallien sisältö on sekä koneluettavaa että semanttisesti yhdenmukaista, jolloin järjestelmät voivat automaattisesti ymmärtää laitteiden ja prosessien merkityksen.

Koneluettavuudella viitataan rakenteisiin, joita ohjelmistot voivat käsitellä automaattisesti ilman manuaalista tulkintaa (Pérez, Arenas & Gutierrez 2009). Semanttinen integraatio tarkoittaa sitä, että eri lähteistä tuleva tieto sidotaan yhteiseen käsitemalliin ja eksplisiittisiin suhteisiin, jolloin merkitys säilyy järjestelmärajapintojen yli Rasmussen ym. 2018). Tietämysgraafi tarjoaa tähän sopivan rakenteen: solmut kuvaavat olioita (esim. laite, tila, mittapiste) ja kaaret niiden

suhteita, ja kokonaisuus voidaan varustaa ontologioilla päättelyä ja kyselyjä varten (Angles & Gutierrez 2008). Ontologiat, kuten Brick (laitteet, datapisteet ja ohjaussuhteet) ja BOT (rakennuksen topologia ja tilahierarkia), antavat formaalin sanaston ja säännöt, joiden avulla eri tietolähteet voidaan tehdä yhteismitalliseksi (Balaji ym. 2016a; Rasmussen ym. 2018). Lisäksi SHACL-säännöt tukevat RDF-graafien laadun ja rakenteen validoimista (Pérez ym. 2009).

Edellä mainittu kirjallisuus määrittelee keskeiset käsitteet täsmällisesti, mutta useimmat lähteet keskittyvät joko yleiseen semanttiseen kehykseen tai yksittäisen ontologian soveltamiseen. Talotekniikan laite- ja ohjaustason mallinnus (esimerkiksi IV-koneen komponenttien järjestys ja ohjausketjut) jää kuitenkin sovellustason tehtäväksi. Tämän vuoksi opinnäytetyöntekijä yhdistää semanttisen mallinnuksen, järjestelmätason kuvaukset ja sääntöpohjaisen validoinnin, ja arvioi lähestymistavan toimivuutta kahdessa käyttötapauksessa (IV-kone ja kiihtyvyysanturi).

### 3.1 Rakennuksen tietomallit ja niiden hyödyntäminen

Rakennuksen tietomallit (Building Information Models, BIM) muodostavat perustan digitaalisen suunnittelutiedon hallinnalle ja siirrolle. BIM ei rajoitu pelkästään rakennuksen geometrian esittämiseen, vaan se sisältää myös tietoa rakennusosien ominaisuuksista, järjestelmistä ja suhteista.

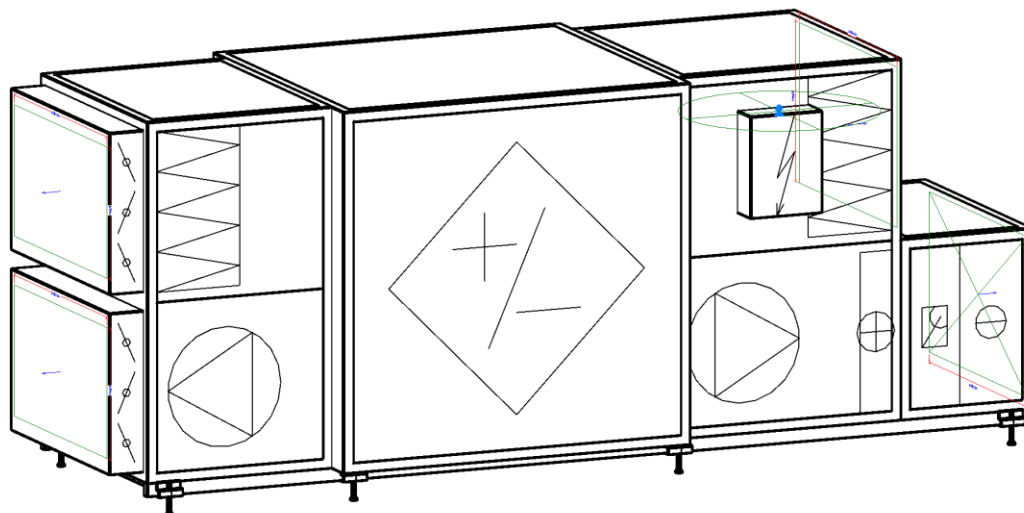
IFC (Industry Foundation Classes) on yleisimmin käytetty avoin standardi, jonka avulla tietoa voidaan siirtää ohjelmistojen välillä (Pauwels & Terkaj 2016: 126–135). IFC:n vahvuus on sen laaja-alainen sovellettavuus, mutta sen hyödyntäminen edellyttää lisämäärittelyjä ja soveltamisohjeita, jotta tiedon semanttinen sisältö säilyy yhdenmukaisena eri hankkeissa (Kolari 2021: 45–52). IFC sisältää luokkia antureille ja toimilaitteille, mutta laitetason sisäiset komponentit eivät ole erillisinä tietorakenteina, mikä rajoittaa käyttö- ja ylläpitovaiheen soveltamista (buildingSMART International 2023: 43).

Myös muita tietomalleja on käytössä. COBie (Construction Operations Building information exchange) on suunniteltu erityisesti rakennuksen ylläpitovaiheen tiedonsiirtoon. gbXML (Green Building XML) tukee energiasimulointeja, ja CityGML mahdollistaa rakennusten liittämisen laajempiin kaupunkimalleihin. Nämä täydentävät IFC:tä, mutta niiden yhteentoimivuus edellyttää semanttista harmonisointia.

Zhang ym. (2022: 7) esittävät tekoälyn ja AI:n (artificial intelligence) integraatioiden pääsovelluksia rakennuksen elinkaaren eri vaiheissa. Tietomallit voivat toimia tietovarastona ja päätöksenteon tukena suunnittelusta aina käyttö- ja ylläpitovaiheeseen sekä purkuun saakka. Niitä voidaan hyödyntää esimerkiksi riskienhallinnassa, kunnossapidossa, energiatehokkuuden optimoinnissa ja kestävä kehityksen arvioinnissa. Näin BIM ei ole ainoastaan suunnitteluvaiheen työkalu, vaan koko rakennuksen elinkaaren kattava tiedonhallinnan alusta (Zhang ym. 2022: 7). He esittävät kattavan tiedonhallinnan alustan, jossa yhdistyvät BIM ja tekoälyratkaisut. Tätä havainnollistetaan liitteessä 1 (kuva L1-L2), joka esittää pääsovellukset BIM–AI-integraatioissa.

Kuva L2 (liite 1) puolestaan täydentää tätä näkökulmaa esittämällä tekoälyteknologioita, joita voidaan yhdistää BIM-malleihin. Näitä teknologioita hyödynnetään esimerkiksi energian optimoinnissa, vikojen ennakoinnissa ja käyttöympäristöjen mukauttamisessa (Zhang ym. 2022: 10).

Nykyinen suunnittelukäytäntö ei kuitenkaan sisällä kaikkia järjestelmiä ja niiden komponentteja. Kuva 2 on esimerkki Recairin ilmanvaihtokoneen BIM-mallista, jossa ei ole mukana koneen ohjaamiseen liittyviä aktiivilaitteita, antureita tai väyläpisteitä. Tämä havainnollistaa IFC-formaatin rajoitteita: vaikka malli kuvaa laitteen geometrian, sen toiminnalliset komponentit jäävät usein mallintamatta. Tämä rajoittaa mallin hyödynnettävyyttä käyttö- ja ylläpitovaiheessa, jolloin juuri näiden komponenttien data on keskeistä (Kähärä 2023, henkilökohtainen viestintä).



Kuva 2. Recairin ilmanvaihtokoneen BIM-malli ilman aktiivilaitteita (Kähärä 2023).

BIM:n tulevaisuus kytkeytyy yhä vahvemmin tekoälyyn. Tekoälyteknologioiden avulla tietomalleja voidaan rikastaa semanttisella tiedolla, automatisoida tarkistusta ja parantaa käytön aikaista analytiikkaa (Zhang ym. 2022; Xue ym. 2021). Tämä kehitys tukee siirtymää staattisista suunnittelumalleista kohti digitaalisia kaksosia, jotka mahdollistavat ennakoivan kunnossapidon ja kestäväen kehityksen tavoitteiden seurannan.

IFC-standardin vahvuutena on avoimuus ja laaja kansainvälinen käyttö, mutta sen rajallisuus näkyy erityisesti laitetason sisäisten komponenttien kuvaamisessa. COBie, gbXML ja CityGML täydentävät IFC:tä, mutta semanttisen yhteentoimivuuden ongelma jää ratkaistavaksi. Opinnäytetyöntekijä hyödyntää tätä havaintoa osoittaakseen, että pelkkä formaatti ei riitä, vaan tarvitaan semanttisia lisäkerroksia ja soveltamisohjeita, jotta talotekniikan järjestelmät voidaan kuvata analytiikan kannalta mielekkäällä tavalla.

## 3.2 Digitaalinen kaksonen

Digitaalinen kaksonen (Digital Twin, DT) on käsite, joka laajentaa rakennuksen tietomallin (BIM) käyttöä yhdistämällä staattisen suunnittelutiedon ja dynaamisen operatiivisen datan yhdeksi kokonaisuudeksi. Se ei ainoastaan kuvaa rakennuksen nykyistä tilaa, vaan tarjoaa myös kyvyn simuloida, analysoida ja ennakoita tulevaa käyttäytymistä. Abdelrahman ym. (2024: 3–5) määrittelevät digitaalisen kaksosen rakentamisen kontekstissa järjestelmäksi, joka sisältää staattisen mallin, semanttisen kerroksen, kaksisuuntaisen tiedonsiirron sekä ennustavan kyvykkyyden. Li ym. (2022: 6–9) puolestaan ehdottavat digitaalisen kaksosen perusmallia, jossa yhdistyvät fysikaaliset mekanismit, vuorovaikutusmallit ja reaaliaikainen data, mahdollistaen rakennuksen sisäisten tilojen ja toimintojen seurannan sekä hallinnan.

Rakennusalan sovelluksissa digitaalinen kaksonen voidaan nähdä BIM-mallin luonnollisena jatkumona. BIM tarjoaa rakenteellisen perustan rakennuksen geometriasta, materiaaleista ja järjestelmistä. Sen sijaan digitaalisen kaksosen lisäarvo syntyy reaaliaikaisesta datasta – sensoreista, rakennusautomaatiosta ja käyttäjäjärjestelmistä. Käytännössä tämän datan integrointi ei kuitenkaan ole ongelmattonta: vaikka IFC-standardi sisältää luokkia antureille ja toimilaitteille, laitetason sisäiset komponentit (esimerkiksi ilmanvaihtokoneen toimilaitteet, anturit ja väyläpisteet) eivät yleensä ole mallissa erillisinä tietorakenteina (BuildingSMART International 2023: 43). Tämä johtaa siihen, että käyttö- ja ylläpito-vaiheessa juuri kriittisin osa – rakennusautomaation laitetieto – on integroitava jälkikäteen, mikä vaikeuttaa digitaalisen kaksosen toteutusta (Kähärä 2023, henkilökohtainen viestintä).

Tutkimuskirjallisuus korostaa lisäksi, että digitaalisen kaksosen onnistunut hyödyntäminen edellyttää useiden rinnakkaisten osa-alueiden hallintaa: ajantasainen BIM-malli, kattava ja luotettava mittausdata, toimiva datainfrastruktuuri sekä semanttisesti yhdenmukaiset standardit. Nguyen ym. (2023: 5–8) ja Cespedes-Cubides & Jradi (2024: 4–7) osoittavat, että yleisimmät käyttötapaukset liittyvät

energiatehokkuuden parantamiseen, komponenttien tilaseurantaan, poikkeamien tunnistukseen ja ennakoivaan kunnossapitoon. Samalla heidän tutkimuksensa osoittavat, että vanhoissa rakennuksissa digitaalisen kaksosen toteutus on erityisen haastavaa, koska suunnitteluaineisto on puutteellista tai vanhentunutta (Cespedes-Cubides & Jradi 2024).

Näin ollen digitaalinen kaksonen ei ole vain tekninen työkalu, vaan se vaatii semanttisen rakenteen, joka takaa eri järjestelmien välisen yhteentoimivuuden. Karabulut ym. (2024: 240–243) osoittavat systemaattisessa katsauksessaan, että ontologiat muodostavat keskeisen välineen digitaalisten kaksosten semanttisen yhtenäisyyden ja automaattisen päättelyn mahdollistamisessa. Ghorbani ym. (2025: 7–10) kehittivät erityisesti digitaalisen kaksosen käyttö- ja ylläpitovaiheisiin suunnatun ontologian, joka rakentuu BIM-ontologian pohjalle ja tukee operatiivisia päätöksentekosovelluksia. Ramonell ym. (2023: 17–19) puolestaan esittävät tietämysgraafeihin perustuvan dataintegraatiojärjestelmän, joka yhdistää useista lähteistä tulevat tiedot digitaalisen kaksosen hallintaan. Samansuuntaisesti Chen ym. (2025) demonstroivat, miten digitaalisen kaksosen ja tietämysgraafin yhdistelmä voi parantaa reaaliaikaista riskienhallintaa rakennus- ja infrastruktuurikohteissa.

Näiden tutkimusten perusteella digitaalinen kaksonen voidaan nähdä siltaana staattisen tietomallin ja semanttisen tietämysgraafin välillä. Se yhdistää BIM-mallien koneluettavan tiedon käyttö- ja ylläpitovaiheen tarpeisiin, mutta edellyttää samalla semanttista harmonisointia ja graafipohjaisia rakenteita, jotta eri lähteistä tuleva data voidaan yhdistää ja hyödyntää tehokkaasti. IFC:n rajoitukset rakennusautomaation mallintamisessa korostavat tarvetta kehittää myös uusia mallinnuskieliä ja metamalleja, joilla voidaan kuvata järjestelmät ja niiden aktiiviset laitteet tarkemmin. Näin digitaalinen kaksonen toimii johdantona seuraavaan lukuun, jossa tarkastellaan mallinnuskieliä ja metamalleja ja niiden roolia digitaalisen kaksosen täydentäjänä.

### 3.3 Mallinnuskielet ja metamallit

Edellisessä luvussa todettiin, että digitaalisen kaksosen toteutus edellyttää enemmän kuin pelkkää IFC-mallia. Erityisesti IV-konehuoneissa olevat rakennusautomaation laitteet ja niiden vuorovaikutus jäävät usein mallin ulkopuolelle. Tämä luo tarpeen hyödyntää formalisoituja mallinnuskieliä ja metamalleja, joilla voidaan kuvata järjestelmien rakenteet, toiminnot ja vuorovaikutussuhteet täsmällisemmin ja koneellisesti käsiteltävällä tavalla.

Mallinnuskieli on kieli, jonka avulla voidaan kuvata järjestelmien rakennetta, toimintaa ja vaatimuksia ennalta määritellyillä käsitteillä ja symboleilla. Metamalli puolestaan määrittelee, mitä käsitteitä mallissa voidaan käyttää ja millä tavoin niitä voidaan yhdistellä. Rakennettuun ympäristöön sovellettuna tämä tarkoittaa, että esimerkiksi laitteet, tilat ja anturit voidaan kuvata yhdenmukaisella ja koneellisesti käsiteltävällä tavalla.

Yksi merkittävimmistä mallinnuskielistä on SysML v2 (Systems Modeling Language), joka on kehitetty erityisesti järjestelmätason mallintamiseen. SysML v2 mahdollistaa sekä rakenteellisten ominaisuuksien (lohko- ja komponenttikaa- viot) että toiminnallisten piirteiden (sekvenssi- ja tilakaaviot) kuvaamisen. Lisäksi sen avulla voidaan esittää vaatimuksia ja parametreja, mikä tekee siitä käyttökelpoisen myös rakennusautomaation ja talotekniikan mallintamisessa. Baras ym. (2015: 18–24) ja Morkevicius (2024: 6–15). Tutkimuksissa SysML v2 on osoittautunut soveltuvaksi esimerkiksi ilmanvaihtokoneiden toiminnan mallintamiseen ja liitettäväksi simulointikehyksiin, kuten Modelicaan (Baras & others 2015: 18–24). Viimeaikainen kehitys on lisäksi mahdollistanut SysML v2-dia- grammien muuntamisen OWL-ontologioiksi, jolloin ne voidaan yhdistää semant- tisiin tietomalleihin ja RDF-graafeihin.

SysML v2:n rinnalla on olemassa muita mallinnuskieliä ja metamalleja, joilla on erityyppisiä sovelluksia. UML (Unified Modeling Language) on ohjelmisto- ja jär- jestelmäkehityksessä laajasti käytetty kieli, josta SysML v2 on osittain johdettu.

AADL (Architecture Analysis & Design Language) on kehitetty erityisesti sulautettujen järjestelmien analysointiin ja arkkitehtuurin kuvaamiseen, kun taas ArchiMate on tarkoitettu yritysarkkitehtuurin mallintamiseen ja tarjoaa välineet liiketoiminnan, prosessien ja teknologian välisen suhteen kuvaamiseen. Vaikka AADL ja ArchiMate eivät ole rakennustekniikassa yhtä yleisiä kuin SysML v2, ne tarjoavat hyödyllisiä viitekehyksiä erityisesti silloin, kun järjestelmiä tarkastellaan osana laajempaa organisaatiokontekstia. IFC:n metamalli puolestaan toimii rakennustiedon pohjana ja määrittelee rakennusosat, tilat, materiaalit ja niiden suhteet, kun taas mallinnuskielet kuten SysML v2 täydentävät sitä tuomalla mukaan toiminnallisen ja käyttäytymisen näkökulman Pauwels & Terkaj (2016: 100–133) sekä Pauwels & Roxin (2020).

SysML v2 tarjoaa vahvan välineistön järjestelmätason kuvaamiseen, mutta suurin osa tutkimuksista kohdistuu ohjelmisto- tai yleiseen järjestelmäkehitykseen. Talotekniikan konteksti on vähemmän käsitelty, mikä jättää aukon soveltamiseen rakennusautomaation laite- ja ohjaustasolla ja järjestelmätiedon kuvaajana. Opinnäytetyössä tarkastellaan, voisiko SysML v2:n integrointi RDF/OWL-pohjaisiin tietämysgraafeihin tarjota keinon tehdä mallinnuksesta täsmällisempää ja yhteen toimivampaa talotekniikan ja rakennusautomaation tietomallien laadinnassa, joka parantaisi elinkaaren aikaista analytiikkaa ja tiedonhakua.

Yhteenvetona voidaan todeta, että mallinnuskielet ja metamallit tarjoavat välttämättömän välineistön järjestelmä- ja laitetason yksityiskohtien kuvaamiseen, joita pelkkä IFC ei tavoita. Ne muodostavat keskeisen perustan digitaalisen kaksoson kehittämiseksi. Kuitenkin, jotta niiden sisältämä tieto voisi palvella rakennuksen koko elinkaarta ja olla yhdistettävissä suunnittelutietoon, niiden tulisi olla osa semanttisesti rikastettua tietorakennetta. Tämä edellyttää, että myös mallinnuskielillä tuotetut metamallit voidaan muuntaa tietämysgraafeihin ja ontologioihin, jolloin eri tietolähteet voidaan integroida yhtenäiseksi kokonaisuudeksi. Seuraavassa luvussa tarkastellaan tätä kysymystä tarkemmin analysoimalla tietämysgraafeja tiedon tallennusmuotona.

### 3.4 Tietämysgraafit tiedon tallennusmuotona

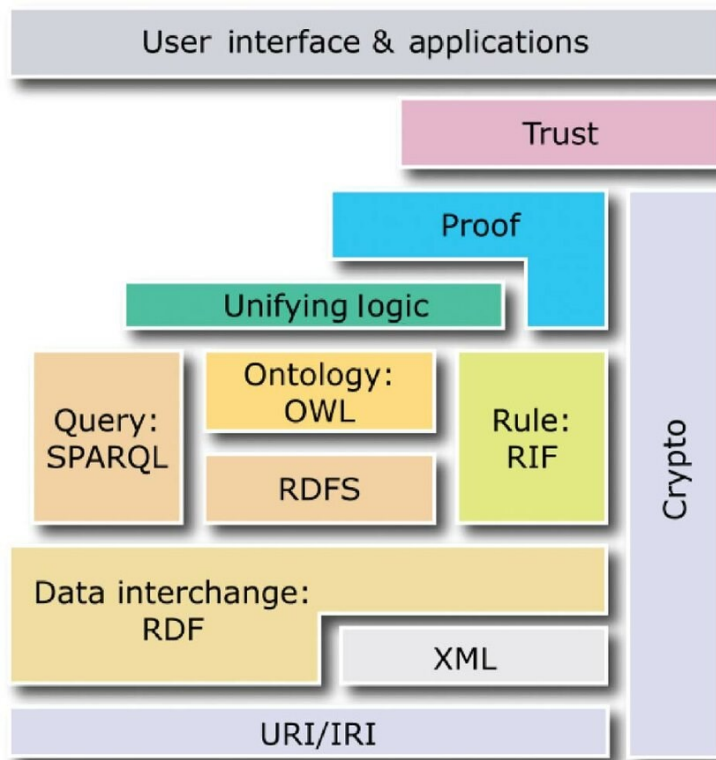
Edellisessä luvussa todettiin, että mallinnuskielet ja metamallit, kuten SysML v2, UML ja AADL, tarjoavat välineitä järjestelmien rakenteiden ja toimintojen kuvaamiseen. Jotta nämä kuvaukset eivät jäisi erillisiksi mallinnuksen tuloksiksi (artefakteiksi), tarvitaan tallennusmuoto, joka mahdollistaa tietojen merkitysten ja suhteiden kuvaamisen sekä tietojen yhdistämisen eri lähteistä, kuten suunnittelumalleista ja IFC-tietomalleista.

Tässä yhteydessä voidaan hyödyntää semanttisen webin ekosysteemiä, joka muodostaa arkkitehtuurin koneellisesti ymmärrettävän tiedon kuvaamiseen ja yhdistämiseen. Semanttinen web ei ole yksittäinen teknologia, vaan kerroksittain rakentuva viitearkkitehtuuri, joka määrittelee, miten tieto tunnistetaan, esitetään, linkitetään ja kysellään eri järjestelmien välillä. Semanttisen webin kerrosarkkitehtuuri on esitetty Kuva 3. Semanttisen webin kerrosarkkitehtuuri. Tietämysgraafit hyödyntävät erityisesti RDF-, OWL- ja SPARQL-kerroksia, joilla tieto voidaan esittää merkityssisällöltään jäsennellyssä muodossa (Trellet ym. 2018)

Alimmilla tasoilla määritellään tiedon tunnistaminen ja siirto. URI/IRI (Uniform Resource Identifier / Internationalized Resource Identifier) antaa jokaiselle tietoelementille yksilöllisen tunnisteen. XML ja RDF (Resource Description Framework) tarjoavat rakenteen ja formaatin tiedon esittämiseen ja vaihtamiseen.

Keskimmäisillä tasoilla määritellään tiedon merkitys ja kyselylogiikka. RDFS (RDF Schema) ja OWL (Web Ontology Language) kuvaavat käsitteiden ja suhteiden merkityksiä, jolloin tieto ei ole pelkkää dataa vaan myös semanttisesti tulkittavaa. SPARQL mahdollistaa tiedon kyselyn ja yhdistämisen eri lähteistä, kun taas RIF (Rule Interchange Format) mahdollistaa sääntöpohjaisen päättelyn.

Ylemmillä tasoilla, kuten Unifying Logic, Proof, Trust ja Crypto, määritellään tiedon eheyteen, todennettavuuteen ja luotettavuuteen liittyviä mekanismeja, joita käytetään erityisesti avoimen webin ympäristöissä.



Kuva 3. Semanttisen webin kerrosarkkitehtuuri. Tietämysgraafit hyödyntävät erityisesti RDF-, OWL- ja SPARQL-kerroksia, joilla tieto voidaan esittää merkitysisällöltään jäsenellyssä muodossa (Trellet ym. 2018).

Tietämysgraafi on tämän ekosysteemin käytännön toteutus – konkreettinen tietorakenne, joka muodostuu, kun RDF-, OWL- ja SPARQL-kerroksia käytetään yhdessä. Se toimii yhtenäisenä tallennus- ja kyselykerroksena, johon voidaan muuntaa sekä suunnittelumallit että järjestelmämallit ja liittää ne toisiinsa merkitysisällön perusteella. Tämä tekee tiedosta yhteismitallista, koneluettavaa ja laajennettavaa, mikä on erityisen tärkeää, kun rakennuksen eri järjestelmistä peräisin olevaa tietoa halutaan tarkastella kokonaisuutena.

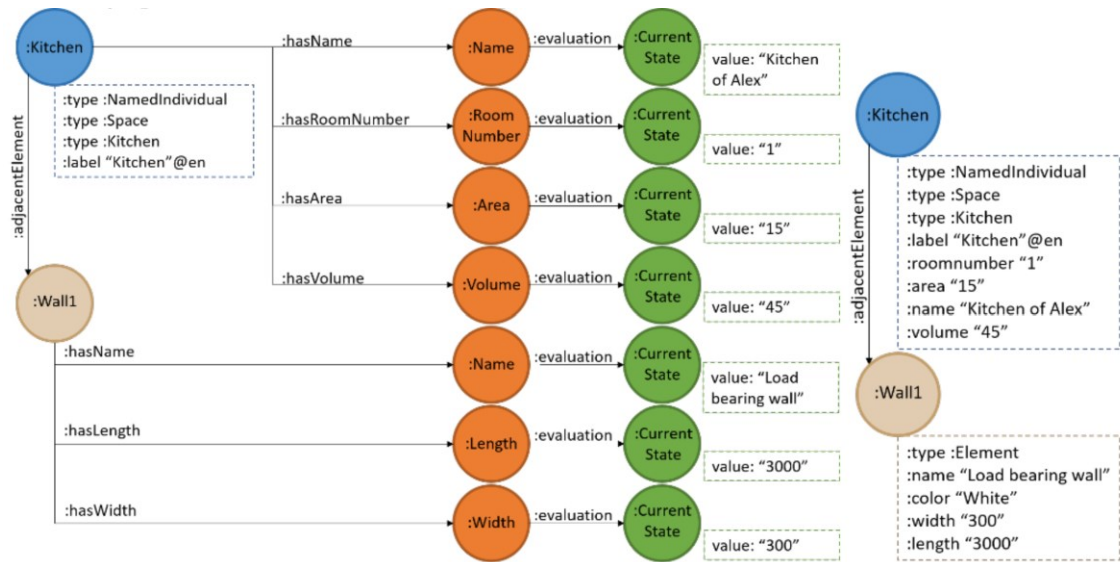
Tutkimukset osoittavat, että mallinnuskielet voidaan kytkeä tietämysgraafeihin. Boronat ym. (2021: 4–6) esittävät lähestymistavan, jossa SysML v2-kaavioita muunnetaan OWL-ontologiaksi, jolloin mallien semanttinen sisältö voidaan liittää suoraan tietämysgraafiin. Vastaavasti Gasevic ym. (2009: 201–210) kuvaavat, kuinka UML-malleja voidaan muuntaa RDF- ja OWL-rakenteiksi, mikä mah-

dollistaa ohjelmistoarkkitehtuurien ja järjestelmäkuvausten yhdistämisen avoimiin semanttisiin kyselyihin. Samansuuntaisesti Bader ym. (2025: 1–2, 4, 6, 9–10) osoittavat, että SysML v2- ja tietämysgraafien yhdistelmä tarjoaa tehokkaan keinon hallita monimutkaisia teknisiä järjestelmiä, erityisesti silloin, kun eri lähteistä peräisin olevaa tietoa on integroitava yhteen.

Tämä lähestymistapa vahvistaa tietämysgraafien roolia rakennusalan tiedonhallinnassa. Ne eivät ainoastaan kuvaa rakennuksen osia ja niiden suhteita, vaan toimivat myös mallinnuskielten ja metamallien semanttisena tallennusmuotona. RDF tarjoaa standardoidun ja yhteentoimivan tavan liittää ontologioita, kun taas ominaisuuksiin perustuvat graafimallit (property-graafit) tukevat tehokasta analytiikkaa ja sovelluskohtaista laajennettavuutta (Pérez ym. 2009; Robinson, Webber & Eifrem 2015; Angles & Gutierrez 2008).

Tietämysgraafit ovat nousseet keskeiseksi lähestymistavaksi, kun rakennuksiin liittyvää tietoa halutaan tallentaa ja käsitellä joustavasti. Perinteiset tietokantarakenteet eivät sovellu hyvin monimutkaisten suhteiden mallintamiseen, sillä ne pakottavat tiedon taulukkomuotoon, jossa suhteet jäävät usein implisiittisiksi. Graafipohjaisessa lähestymistavassa suhteet ovat ensiluokkaisia tieto-olioita: huone voi sisältää laitteen, anturi voi mitata suuretta tietyssä tilassa ja puhallin voi syöttää ilmaa tiettyyn järjestelmään. Tämä tekee graafimallista erityisen luontevan rakennusten teknisten järjestelmien kuvaamiseen, sillä juuri suhteet muodostavat olennaisen osan järjestelmän toimintaa (Zhu ym. 2021: 1–8).

Tämä tekee property-graafista rakenteeltaan yksinkertaisemmän ja kevyemmän esimerkiksi reaaliaikaiseen analytiikkaan ja sovelluskehitykseen, kun taas RDF-malli tarjoaa laajemman semanttisen integraation ja yhteentoimivuuden eri järjestelmien välillä. Kuva 4. RDF- ja property-graafien esitystapojen vertailu älykötidatan kontekstissa (Donkers ym. 2020).



**B:** LPG graph using the NSMTX RDF import in Neo4j

**C:** Custom LPG graph

Kuva 4. RDF- ja property-graafien esitystapojen vertailu älykotidatan kontekstissa (Donkers ym. 2020).

Vasemmalla puolella esitetään RDF-malli, jossa tieto rakentuu kolmikantaisista lausumista (tripleteistä). Jokainen ominaisuus – esimerkiksi huoneen nimi, numero, pinta-ala tai tilavuus – muodostaa oman solmunsa ja liitetään muihin tietoihin relaatioiden avulla. Oikealla puolella on property-graafi, jossa samat tiedot liitetään suoraan solmuihin ominaisuuksina (properties).

Taulukko 1 on esitetty liitteessä 3 ja täydentää vertailua osoittamalla, kuinka nämä kaksi lähestymistapaa palvelevat eri tarkoituksia: RDF korostaa merkitysisältöön perustuvaa tietojen yhdistämistä, kun taas property-graafit tukevat tehokasta ja käytännönläheistä analytiikkaa (Donkers ym. 2020, LDAC Conference).

RDF (Resource Description Framework) on vakiintunut tapa esittää tietämysgraafeja. Siinä tieto kuvataan tripleinä, joissa jokainen lausuma koostuu subjektista, predikaatista ja objektista. RDF:n vahvuus on sen standardointi ja semanttinen yhteentoimivuus, sillä siihen voidaan liittää ontologioita, jotka määrittelevät käsitteiden merkityksen. Näin tieto ei ole vain dataa, vaan se voidaan myös ymmärtää ja yhdistää eri lähteistä peräisin oleviin tietoihin. Kyselykielenä

käytetään SPARQLia, joka mahdollistaa monimutkaisten kyselyiden tekemisen ja päättelyn tiedon pohjalta (Pérez ym. 2009).

Property-graafit tarjoavat toisen lähestymistavan. Niissä solmuihin ja kaariin voidaan liittää ominaisuuksia joustavasti, mikä tekee niistä käytännöllisiä interaktiiviseen analyysiin ja dynaamiseen tiedonhallintaan. Neo4j on tunnettu property-graafeihin perustuva tietokanta, jossa voidaan hyödyntää esimerkiksi Cypher-kieltä (Robinson ym. 2011: ch 3). Property-graafeissa painopiste on enemmän käytännön tehokkuudessa ja analysoitavuudessa, kun taas RDF:n vahvuus on semanttinen syvyys ja kansainvälinen yhteen toimivuus. Käytännössä nämä lähestymistavat täydentävät toisiaan: RDF soveltuu erityisesti merkityssisällöltään jäsennellyin ja yhteismitallisen tiedon esittämiseen, kun taas ominaisuuksiin perustuvat graafimallit, eli niin sanottu *property-graafit*, toimivat hyvin nopeassa analytiikassa ja käytännön sovelluksissa (Angles & Gutierrez 2008: 1–39).

RDF tarjoaa vahvan semanttisen perustan, mutta sen käyttöönotto vaatii laajaa ontologioiden ja standardien hallintaa. Property-graafit taas tarjoavat käytännöllisyyttä, mutta menettävät osan semanttisesta syvyydestä. Suurin osa kirjallisuudesta käsittelee näitä erillisinä lähestymistapoina (Pérez ym. 2009; Robinson ym. 2015). Opinnäytetyöntekijä keskittyy erityisesti RDF-pohjaiseen merkityssisällön integraatioon, joka mahdollistaa suunnittelu- ja järjestelmämallien yhdistämisen ontologioiden avulla. Samalla kuitenkin tunnistetaan property-graafien tarjoamat mahdollisuudet käytännön analytiikan ja sovellustason optimoinnin tukena.

Tämä lähestymistapa tukee opinnäytetyön kokonaismallia, jossa RDF-pohjainen tietämysgraafi toimii eri tietomallien yhdistävänä rajapintana ja varmistaa, että rakennuksen elinkaaren aikana syntyvä tieto säilyy merkityssisällöltään yhtenäisenä ja koneellisesti hyödynnettävänä.

### 3.5 Ontologiat rakennetussa ympäristössä

Ontologiat muodostavat keskeisen osan rakennetun ympäristön tietomallinnusta, sillä ne tarjoavat semanttisen kerroksen, jonka avulla rakennusten osia, järjestelmiä ja laitteita voidaan kuvata yhdenmukaisella tavalla. Ilman ontologioita tietämysgraafit jäisivät pelkiksi solmuiksi ja suhteiksi, joiden merkitys ei ole selvä eri sovelluksille. Ontologiat siis mahdollistavat sen, että esimerkiksi “ilmanvaihtokone”, “anturi” ja “tila” ymmärretään samalla tavalla riippumatta siitä, tuleeko tieto IFC-mallista, rakennusautomaation järjestelmästä vai IoT-sensoriverkosta.

Rakennetussa ympäristössä tietämysgraafeja on hyödynnetty monin tavoin. IFC-mallien muuntaminen IFC-Graphiksi on parantanut tiedon kyseltävyyttä ja mahdollistanut implisiittistenkin suhteiden paljastamisen (Pauwels & Roxin 2020). Brick-ontologia on tarjonnut standardoidun mallin rakennusautomaation pisteiden ja laitteiden kuvaamiseen, kun taas BOT on keskittynyt rakennuksen topologiaan suhteisiin ja tilojen väliseen hierarkiaan (Balaji ym. 2016, s. 1276–1284; Rasmussen ym. 2018, s. 95–101). Project Haystack on kehittänyt käytännönläheisen tagausjärjestelmän ja API-tuen, joka helpottaa järjestelmien välistä integraatiota (Project Haystack 2016). Lisäksi BEKG, rakennetun ympäristön tutkimusjulkaisuihin perustuva tietämysgraafi, on osoittanut graafimuotoisen tiedon potentiaalin myös akateemisessa tiedonhallinnassa (Yang ym. 2022).

Ontologioiden soveltaminen ei rajoitu vain rakennusautomaation ja tilahierarkioiden mallintamiseen. Esimerkiksi robotiikan ja automaation sovelluksissa on tutkittu, kuinka BIM-malleja voidaan muuntaa semanttiseksi dataksi reaaliaikaisen robotin navigoinnin tueksi, jolloin arviointikriteereinä ovat toteutettavuus, skaalautuvuus ja tiedonsiirron luotettavuus (Pauwels ym. 2023). Myös laajempia viitekehyksiä, kuten RealEstateCore ja SEAS, on kehitetty tukemaan kiinteistöjen ja kaupunkien semanttista yhteentoimivuutta (Hammar ym. 2021; Lefrançois ym. 2020).

Näiden esimerkkien perusteella voidaan todeta, että ontologiat muodostavat olennaisen välineen suunnittelutiedon, rakennusautomaation ja käyttöaikaisen datan semanttisessa yhdistämisessä. Ne toimivat linkkinä eri tietomallien ja tietämysgraafien välillä, mahdollistaen tietojen yhteen toimivuuden koko rakennuksen elinkaaren ajan. Ontologioiden avulla voidaan siirtyä pelkästä koneluettavuudesta semanttiseen ymmärrettävyyteen, mikä luo pohjan digitaalisille kaksoille, reaaliaikaiselle analytiikalle ja kestäväen kehityksen tavoitteiden saavuttamiselle.

Brick ja BOT tarjoavat selkeän lähtökohdan laitteiden ja tilojen kuvaamiseen, mutta tutkimukset osoittavat, että niiden soveltaminen jää usein erillisiin pilotteihin. Project Haystack tuo käytännönläheisyyttä, mutta sen semanttinen formaalius on rajallinen. RealEstateCore ja SEAS laajentavat soveltuvuutta, mutta niiden käyttöönotto vaatii huomattavaa resursointia. Opinnäytetyöntekijä yhdistää nämä havainnot ja testaa, kuinka ontologiat voidaan konkreettisesti ankkuroida case-esimerkkeihin: IV-koneen komponentit kuvataan Brickillä ja tilasuhteet BOT:illa, kun taas kiihtyvyyssanturin datapisteet linkitetään Haystackin tai Brickin tageihin.

### 3.6 Yhteenveto

Luvussa 3 on osoitettu, että pelkkä koneluettavuus ei riitä rakennusten teknisten järjestelmien hallinnassa. Tieto voi olla muodollisesti oikeassa formaatissa, mutta sen hyödyntäminen elinkaaren aikana edellyttää semanttista yhtenäisyyttä ja eri mallien välistä yhteentoimivuutta. Tämä luo tarpeen yhdistää suunnittelumallit, käyttöaikainen data ja semanttiset rakenteet kokonaisuudeksi, joka mahdollistaa automaattisen analytiikan ja päätöksenteon. Keskeiset havainnot voidaan tiivistää seuraavasti:

- BIM tarjoaa perustan rakennuksen geometrian ja järjestelmien kuvaamiseen, mutta sen rajoitteet, kuten aktiivilaitteiden ja anturien puutteellinen mallintaminen, tekevät siitä riittämättömän käyttö- ja ylläpitovaiheessa (buildingSMART International 2023).

- Digitaalinen kaksonen (DT) toimii siltana staattisen suunnittelumallin ja dynaamisen operatiivisen tiedon välillä. Sen onnistunut hyödyntäminen edellyttää reaaliaikaista dataa, semanttisia malleja ja tietojen yhdistämistä eri lähteistä (Nguyen ym. 2023; Cespedes-Cubides & Jradi 2024).
- Mallinnuskielet ja metamallit (esim. SysML v2, UML, AADL) täydentävät BIM:iä ja DT:tä tarjoamalla välineet järjestelmien ja laitteiden tarkan rakenteen ja käyttäytymisen kuvaamiseen. Ne ovat kuitenkin hyödynnettävissä täysimääräisesti vasta, kun niiden sisältö voidaan muuntaa tietämysgraafeihin ja ontologioihin (Boronat ym. 2021; Gasevic ym. 2009).
- Tietämysgraafit mahdollistavat mallien, järjestelmien ja mitausdatan yhdistämisen yhtenäiseksi kokonaisuudeksi. RDF tarjoaa standardoidun ja semanttisesti rikastetun lähestymistavan, kun taas property-graafit tukevat tehokasta analytiikkaa ja sovelluskohtaista joustavuutta (Pérez ym. 2009; Robinson ym. 2015; Angles & Gutierrez 2008).
- Ontologiat muodostavat tietämysgraafien semanttisen kerroksen: niiden avulla käsitteet kuten "tila", "anturi" ja "ilmanvaihtokone" saavat täsmällisen merkityksen eri järjestelmissä. Ontologiat kuten Brick, BOT ja Project Haystack ovat jo osoittaneet käytännön sovellettavuutensa rakennusautomaation, tilahierarkioiden ja integraatioiden hallinnassa (Balaji ym. 2016; Rasmussen ym. 2018; Project Haystack 2016).

Kokonaisuutena luvun 3 tarkastelu osoittaa, että rakennusten tiedonhallinta kehittyy staattisista malleista kohti elinkaarisia, sisällöllisesti rikastettuja digitaalisia ekosysteemejä, joissa suunnittelumallit, reaaliaikainen data, mallinnuskielet, tietämysgraafit ja ontologiat muodostavat yhtenäisen tiedonhallinnan perustan.

Luvun tutkimukset vahvistavat, että tekninen yhteistoimivuus ilman semanttista yhtenäisyyttä jää riittämättömäksi. Kuitenkin suurin osa kirjallisuudesta tarkastelee ongelmaa erillisistä näkökulmista – IFC:n, digitaalisen kaksosen tai ontologioiden kautta – ilman kokonaisintegraatiota. Opinnäytetyössä paikataan tämä aukko yhdistämällä semanttiset ontologiat, tietämysgraafit ja järjestelmätason mallinnuskielet, ja arvioi ratkaisua tapausesimerkkien avulla. Tämä integroiva lähestymistapa muodostaa perustan luvuille 4–5, joissa menetelmät ja tekninen toteutus kuvataan tarkemmin.

## 4 Tietomallien ja nimikkeistöjen semanttinen integraatio

### 4.1 Johdanto prosessin kehittämiseen

Kirjallisuudessa tunnistettiin, että talotekniikan järjestelmien tietosisällöllinen mallinnus on osin hajanaista, erityisesti laitteille, joista ei tyypillisesti tehdä tietomallia tai kuvata ontologisesti. Tätä havaintoa vasten kehitettiin tässä työssä menetelmä, joka yhdistää SysML v2 v2 -pohjaisen suunnittelumallin, ontologiset luokittelut sekä SHACL-validoinnin toistettavaksi prosessiksi.

Menetelmä kehitettiin vastaukseksi käytännön ongelmaan: suunnittelumalleista saatava tieto ei tyypillisesti kata ilmanvaihtokoneen (IV-koneen) laite- ja ohjaustason semantiikkaa sellaisella tarkkuudella, että tieto voitaisiin liittää suoraan rakennusautomaation datapisteisiin ja hyödyntää analytiikassa. IFC-standardin käyttö on vakiintunutta rakennusten suunnittelutiedon siirtämisessä, mutta sen sovelluksissa aktiiviset komponentit, kuten puhaltimet, lämmönsiirtimet ja niiden ohjausketjut, jäävät usein kuvaamatta. IFC sisältää geometrian ja rakennuksen rakenteen, mutta ei tarjoa yhtä täsmällistä tapaa mallintaa datapisteitä ja ohjauslogiikkaa. Tämä aiheuttaa katkoksen suunnittelun ja ylläpidon välille: käyttö- ja ylläpitovaiheessa tarvittava tieto joudutaan kokoamaan käsityönä useista lähteistä (Pauwels & Terkaj, 2016; buildingSMART International, 2023).

IV-kone valittiin tutkimuksen kohteeksi, koska se on rakennuksen energiatalouden ja sisäilmaston kannalta keskeinen yksittäinen järjestelmä. Sen toiminta perustuu useiden osien yhteisvaikutukseen: tulo- ja poistoilmapuhaltimet, lämmitys- ja jäähdytyspatterit, lämmöntalteenotto, suodattimet ja säätöpellit muodostavat verkoston, johon liittyy suuri määrä antureita ja toimilaitteita. Juuri nämä osat määrittävät, kuinka hyvin kone toimii energiatehokkuuden, sisäilman laadun ja huollon kannalta.

Koska IFC:n laajentaminen ei tarjoa riittävää ratkaisua laite- ja datapistetason tietosisältöä. Tässä työssä valittiin SysML v2 (Systems Modeling Language) tietomallikieleksi. SysML v2 tarjoaa formaalin mallinnuskielen, jolla voidaan kuvata

järjestelmän rakenteet, portit, attribuutit ja ohjaussuhteet sekä liittää niihin vaatimuksia. Näin voidaan rakentaa järjestelmällinen malli, joka täydentää IFC:tä laite- ja datapistetason semantiikalla. Menetelmän kehitystä tuki opinnäytetyön tekijän osallistuminen kansainväliseen metamallipohjaisen ohjelmistosuunnittelun GenerIoT-hankkeeseen. Hankkeessa SysML v2:ää käytetään monialaisesti järjestelmien formaaliin määrittelyyn. Tässä työssä soveltaminen rajattiin IV-koneeseen ja sen datapisteisiin, mutta periaate on laajennettavissa muihin taloteknisiin kokonaisuuksiin. Opinnäytetyöntekijä jatkaa SysML v2-mallin ja graafiesitysten tutkimista hankkeessa IoT-kenttälaitteen osalta.

## 4.2 Viitekehykset ja käsitteet

Menetelmä yhdistää SysML v2 v2 (rakenteet, portit, attribuutit, ohjaussuhteet), RDF/OWL (semanttinen data), Brick/BOT/SAREF (ontologiat), SKOS (kansalliset sanastot), SSSOM (mappaukset), SHACL (validointi) ja SPARQL (kyselyt). Yhdistelmä mahdollistaa SysML v2-sisällön linkittämisen kansainvälisiin ja kansallisiin tietomalleihin. Liitteessä 2 on esitetty näiden viitekehysten ja käsitteiden selitykset, lyhenteet ja niiden merkitys kehitetyn kannalta.

Koska ontologiat ja sanastot ovat pääosin englanninkielisiä, SysML v2 v2 -mallinnuksessa noudatetaan muunnosta tukevia periaatteita, joiden avulla suunnittelumalli voidaan muuttaa koneluettavaan ja merkityssisällöltään yhteismitalliseen muotoon.

### Mallinnuksessa

- Erotetaan järjestelmätaso ja laiteinstanssit, jotta järjestelmän kokonaisuus ja yksittäiset laitteet voidaan kuvata selkeästi.
- Käytetään englanninkielisiä (EN) termejä pääasiallisina nimikkeinä ja suomenkielisiä (FI) vastineita tukitietona. Termistö perustuu 223P-standardiin sekä kansainvälisiin rakennusautomaation ontologioihin, kuten Brick Schema ja Project Haystack.
- Määritellään tyypit ja roolit erikseen, esimerkiksi puhallin(Fan) on laitetytologia ja tuloilmapuhallin (SupplyFan) sen rooli järjestelmässä.

- Kuvataan portit ja virrat, jotka ovat SysML v2:n käsitteitä järjestelmän rajapintojen ja tiedonsiirron kuvaamiseen. Portti (port) edustaa komponentin liitäntäpistettä, jonka kautta virtaus (flow) — kuten ilma, vesi tai ohjaussignaali — kulkee sisään tai ulos.
- Määritetään virran tyyppi (Air, Water, Control) ja suunta (sisään–ulos) jokaiselle portille.
- Lisätään parametreille yksiköt, esimerkiksi ilmavirta [m<sup>3</sup>/h] ja teho [W].
- Annetaan osille pysyvät tunnisteet (URI), joita käytetään mallin tietokohteiden yksilöinnissä ja linkittämisessä muihin järjestelmiin.
- Liitetään koodistoviitteet attribuutteina, kuten RAVA- ja RAU/LVI-tunnisteet sekä mahdolliset laitekohtaiset attribuutit, jotka mahdollistavat yhteyden kansallisiin tietokantoihin ja sanastoihin.

SysML v2 v2 -mallin laitteille lisättiin attribuutteja, jotka perustuvat RAU- ja LVI-tietokantojen määritelyihin parametreihin. Jokaiselle laitteelle määriteltiin sen toimintaan ja vastuuseen liittyviä ominaisuuksia, kuten esimerkiksi asentaa tai toimittaa, jotka esiintyvät kansallisissa tietokannoissa.

### 4.3 Menetelmäprosessin yleiskuva

Menetelmä on jäsennelty toistettavaksi ja laajennettavaksi prosessiksi, jonka vaiheet etenevät systemaattisesti suunnittelumallista semanttiseen integraatioon.

Prosessin keskeinen tavoite ei ole ainoastaan muuntaa ilmanvaihtokoneen (IV-koneen) SysML v2-malli koneluettavaan ja tarkistettavaan muotoon, vaan erityisesti muodostaa käsitteellinen yhteyskirjasto, joka yhdistää SysML v2:n käsitteet kansainvälisiin ontologioihin (Brick, BOT, SAREF) sekä kansallisiin sanastoihin (rakennusautomaatio- ja LVI-tietokantojen käsitteet sekä RAVA3Pro-tuotesakoodistot).

SysML v2 toimii tässä tiedon ensisijaisena lähteenä, mikä mahdollistaa suhteiden yksiselitteisen määrittelyn ja tukee tiedon hyödynnettävyyttä suunnittelusta aina käyttö- ja ylläpitovaiheeseen saakka.

Menetelmäprosessin vaiheet, tuotokset ja niiden väliset riippuvuudet on esitetty liitteessä 3, jossa kuvataan yksityiskohtaisesti eteneminen SysML v2-mallista inventaariotiedostoon, käsitteelliseen yhteyskirjastoon ja edelleen RDF-graafiksi.

#### 4.4 Työkalut ja ohjelmistot

Menetelmän toteuttaminen edellyttää ympäristöä, joka tukee SysML v2 v2 -standardin mukaista mallinnusta, RDF/OWL-pohjaista tietomallinnusta sekä graafipohjaisen tiedon käsittelyä. Menetelmän arkkitehtuuri voidaan rakentaa siten, että se koostuu kolmesta toisiaan täydentävästä osasta:

- Mallinnuseditori (esimerkiksi SysIDE tai muu Eclipse-pohjainen editori) tuottaa SysML v2 v2 -mallin, joka kuvaa järjestelmän rakenteet ja suhteet koneluettavassa muodossa.
- Ontologia- ja tietovarastopalvelu (esimerkiksi Oxigraph tai GraphDB) tallentaa mallin RDF-muodossa ja mahdollistaa sen käsittelyn ja validoinnin SPARQL-kyselyiden avulla.
- Validointipalvelu suorittaa SHACL-sääntöihin perustuvat tarkistukset ja varmistaa, että malli täyttää semanttiset ja rakenteelliset vaatimukset.

Lisäksi toteutettiin kaksi ohjelmallista komponenttia, jotka muodostavat menetelmän käytännön ytimen:

- Inventaario-ohjelma lukee SysML v2 v2 -mallin ja tuottaa siitä ahu\_inventory.csv-tiedoston, jossa jokaiselle laitteelle, portille ja parametrille annetaan yksilöllinen tunnistus ja metatiedot.
- Käsitevastaavuusohjelma hyödyntää inventaariotiedostoa ja etsii automaattisesti vastaavuuksia kansainvälisten ontologioiden (Brick, BOT, SAREF) ja kansallisten sanastojen (RAU/LVI, RAVA3Pro) välillä.

- (a) Ehdotetut vastineet voidaan tarkistaa ja hyväksyä manuaalisesti, jolloin muodostuu lopullinen käsitevastaavuuskirjasto (SSSOM).
- (b) Käsitevastaavuuksien luotettavuus voidaan lisäksi varmentaa ja vertailla automaattisesti eri työkaluilla, kuten AML (AgreementMakerLight) ja DeepOnto, sekä asiantuntija-arvion perusteella. Tämä vaihe vahvistaa menetelmän tarkkuutta ja parantaa sen toistettavuutta.
- Ontologiamuunnosohjelma (SysML v2-to-RDF converter) lukee SysML v2-mallista tai inventaaritiedostosta rakenteellisen tiedon ja soveltaa SSSOM-kirjastossa määriteltyjä vastaavuuksia. Ohjelma muodostaa RDF/Turtle -muotoisen tietomallin, jossa SysML v2-entiteetit linkitetään ontologioiden käsitteisiin (Brick, BOT, SAREF) ja kansallisiin SKOS-sanastoihin. Tuloksena syntyy graafipohjainen tietomalli, joka toimii lähtöaineistona päättelylle, validoinnille ja SPARQL -kyselyille.

Ohjelmisto on toteutettu siten, että se tukee mikropalveluarkkitehtuurin periaatteita, joissa mallinnus, tietovarastointi ja validointi toimivat erillisinä mutta yhteen liitettävänä osina.

Tämä rakenne parantaa ratkaisun ylläpidettävyyttä, siirrettävyyttä ja laajennettavuutta eri ympäristöihin, ja se voidaan tarvittaessa toteuttaa myös konttipohjaisesti esimerkiksi Azure Container Apps- tai Kubernetes-alustoilla.

Ontologiavaraston tietoja voidaan hyödyntää avoimien rajapintojen, kuten SPARQL-kyselyiden ja REST API -palvelujen kautta, jolloin mallinnettua tietoa voidaan yhdistää muihin järjestelmiin, kuten IoT-datan tai rakennusautomaation tietoihin.

Näin muodostuu skaalautuva ja yhteentoimiva kokonaisuus, jossa suunnittelumallit ja reaaliaikainen data voidaan yhdistää ja jakaa hallitusti myös pilviympäristössä.

#### 4.5 SysML v2-mallin käsittely ja ontologiapohjaisen käsitemallin muodostaminen.

Menetelmä etenee vaiheittain SysML v2-mallin tuottamisesta sen muuntamiseen semanttisesti yhteensopivaan tietomalliin seuraavasti:

##### SysML v2-mallin luonti

- Ilmanvaihtokone (IV-kone) mallinnetaan SysML v2 v2 -standardin mukaisesti järjestelmänä, jossa jokaisella komponentilla on oma rakenteellinen ja toiminnallinen roolinsa.
- Mallissa esitetään keskeiset osat (puhaltimet, patterit, LTO, suodattimet, säätöpellit) sekä portit ja virrat.
- Attribuutteina käytetään suureita (esim. ilmavirta, paine-ero, lämpötila, kosteus, SFP-luku).
- Anturit kuvataan ennen–jälkeen-periaatteella, jotta mittauspisteiden sijoittuminen on yksiselitteistä.

##### Inventaarion muodostus

- SysML v2-mallista tuotetaan CSV-muotoinen inventaario, jossa jokainen komponentti, portti ja attribuutti saa yksilöllisen tunnisteen.
- Inventaario tekee mallista työkaluriippumattoman ja toimii lähtöaineistona automaattiselle luokittelulle ja validoinnille.

##### Luokkaehdotusten muodostus

- Inventaarion perusteella etsitään vastineita kansainvälisistä ontologioista (Brick, BOT, SAREF) ja kansallisista sanastoista (RAVA3Pro, RAU/LVI).
- Prosessi hyödyntää sekä sääntöpohjaista hakua että tekoälyavusteista analyysiä.
- Ehdotuksille liitetään luotettavuusindikaattori ja perustelu.
- Kotimaiset RAVA- ja RAU/LVI-koodistot muunnetaan SKOS-muotoon, mikä oli opinnäytetyön tekijän oma kontribuutio.

##### Käsitevastaavuuksien vahvistus

- Ehdotetut vastineet tarkistetaan ja hyväksytään manuaalisesti.

- Hyväksytyt yhteydet kirjataan SSSOM-muotoon (lähde, kohde, vastaavuustyyppi, perustelu, lähdeviite).
- Tulokset voidaan lisäksi varmentaa ja vertailla automaattisesti työkaluilla AML (AgreementMakerLight) ja DeepOnto sekä asiantuntija-arvion avulla.

#### RDF-graafin muodostus

- Ontologiamuunnos toteutetaan ohjelmallisesti SysML v2-to-RDF converter -ohjelmalla, joka hyödyntää SSSOM-kirjastoa ja luo RDF/Turtle-muotoisen tiedoston.
- Tuloksena syntyy linkkikerros SysML v2-entiteettien, Brick/BOT/SAREF-luokkien ja SKOS-sanastojen välille.
- Graafi mahdollistaa päättelyn, hakukyselyt ja tiedon yhdistämisen muihin järjestelmiin.

#### Validointi ja testaus

- SHACL-säännöillä varmistetaan rakenteiden oikeellisuus ja kardinaliteetit (esim. "IV-koneella on vähintään yksi tuloilman lämpötila-anturi").
- SPARQL-kyselyillä tehdään regressio- ja ristiriitatestejä (esim. "kaikki supply-lämpötila-anturit ennen lämmityspatteria").
- Validointitulokset ohjaavat korjauksia käsitevastaavuuskirjastoon ja varmistavat mallin johdonmukaisuuden.

Näin muodostuu toistettavissa oleva prosessi, jossa SysML v2-mallista voidaan johdonmukaisesti muodostaa validoitu RDF-tietomalli ja yhdistää se kansainvälisiin ja kansallisiin tietokantoihin.

Ontologiapohjaiset tiedostot (SKOS/TTL, RDF/TTL, SSSOM) voidaan julkaista LODI4DH-periaatteiden (Linked Open Data Infrastructure for Digital Humanities) mukaisesti avoimina, versioituina ja pysyviin URI-osoitteisiin viitattuina resursseina (Hyvönen ym. 2022). LODI4DH tukee linkitetyn datan julkaisemista siten, että tietolähteet ovat pitkäikäisiä, koneluettavia ja yhteismitallisia muiden avoimien tietokantojen kanssa.

Menetelmän tuottamat sanastot (RAU, LVI, RAVA) ja käsitevastaavuuskirjastot (SSSOM) noudattavat tätä mallia, jolloin tulokset voidaan myöhemmin integroida avoimeen semanttiseen tietoinfrastruktuuriin ilman työkaluriippuvuuksia.

Tätä sovelletaan käytännössä liitteissä 7–10, joissa esitetään SKOS/TTL- ja RDF-tiedostojen rakenteet ja pysyvät URI-viittaukset.

## 5 Tapaustutkimus: IV-koneen semanttinen mallinnus

Tässä luvussa havainnollistetaan, miten kehitetty menetelmä soveltuu ilmanvaihtokoneen (IV-koneen) semanttiseen mallinnukseen alusta loppuun. Luku kattaa koko etenemispolun vaiheittain: SysML v2-mallin muodostuksesta inventaarion ja ontologiavastaavuuksien (SSSOM) kautta RDF-graafin tuottamiseen ja sen validointiin (SHACL/SPARQL). Kussakin vaiheessa esitetään perusteltu ja toistettavissa oleva toimintamalli, jota havainnollistetaan liitteissä.

### 5.1 Aineisto ja rajaus

Tapaustutkimuksen lähtöaineistona käytettiin ilmanvaihtokoneen (IV-kone) rakennusautomaation mallikaaviota (liite 4). Kaavio sisältää koneen pääkomponentit, kuten puhaltimet, lämmitys- ja jäähdytyspatterit, lämmön talteenoton, suodattimet sekä anturit ja toimilaitteet. Kohteeksi valittiin yksi Granlundin suunnittelumallina toimiva ilmanvaihtokone, jotta menetelmän soveltuvuutta voitiin arvioida rajatussa mutta riittävän monimutkaisessa kokonaisuudessa.

SysML v2 v2 -malli muodostettiin koodipohjaisesti käyttäen SysIDE-editoria, joka mahdollistaa mallin tuottamisen suoraan tekstimuotoisena lähdekoodina. SysIDE:n parseri suorittaa mallin syntaksin ja semantiikan validoinnin reaaliaikaisesti, jolloin virheelliset määrittelyt, kuten puuttuvat portit, määrittelemättömät osat tai väärä syntaksimuoto, tunnistetaan jo kirjoitusvaiheessa.

Tämä validointiprosessi vastaa käytännössä mallin rakenteellista ja semanttista tarkistusta. Parseri analysoi koodin hierarkkisesti ja varmistaa, että kaikki määrittelyt osat ja yhteydet ovat johdonmukaisia SysML v2 v2 -standardin mukaisesti.

Parserin validointi toimii vaiheittain:

- Syntaksitarkistus varmistaa, että koodi noudattaa SysML v2 v2 -kielen rakennetta (part, port, connect, attribute).
- Rakenneanalyysi tunnistaa lohkojen, porttien ja yhteyksien väliset riippuvuudet.

- Semanttinen eheystarkistusta: tunnistaa mallin sisäisiä ristiriitoja ja raportoi niistä virheilmoituksina.

Validoinnin tuloksena saatiin rakenteellisesti eheä malli, joka toimii lähtöaineistona myöhemmille vaiheille ja tarkasteltiin graafisesti Syson-työkalulla (liite 5). Malli vastaa toiminnallisesti perinteistä säätökaaviota mutta on koneluettava ja soveltuu automaattiseen analyysiin ja ontologiapohjaiseen tietomallinnukseen.

## 5.2 Komponenttiluettelo ja alustava käsitevastaavuuskirjasto

Tämä luku vastaa menetelmäprosessin vaihetta 2 (inventaarion muodostus), jossa SysML v2-mallin rakenteellinen tieto jäsennetään taulukkomuotoon jatkokäsittelyä varten.

Menetelmässä SysML v2-mallin rakenteesta — eli sen osista, liitännöistä ja niiden välisistä yhteyksistä — muodostetaan rakenteellinen inventaario, joka koostaa yhteen ilmanvaihtokoneen (IV-koneen) komponentit, niiden ominaisuudet ja keskinäiset suhteet. Inventaario toimii linkkinä rakenteellisen ja semanttisen tason välillä: sen avulla voidaan myöhemmin yhdistää mallin osat kansallisiin koodistoihin ja kansainvälisiin ontologioihin.

SysML v2-mallista johdettava inventaario esitetään taulukkomuodossa (CSV), jossa kukin rivi kuvaa yhtä komponenttia tai yhteyttä ja sarakkeet sisältävät sen tunnisteet, tekniset ominaisuudet ja viittaukset muihin osiin. Tämä mahdollistaa tiedon käsittelyn eri ohjelmistoissa ja sen muuntamisen myöhemmässä vaiheessa RDF-graafin pohjaksi.

Tässä vaiheessa hyödynnetään RAU-, LVI- ja RAVA-koodistoja, joista saadaan SysML v2-mallin property-attribuutteja täydentävät tekniset tiedot ja suunnitteluarvot. Tämä kielipohjainen vaihe erotetaan ontologisesta linkityksestä, jotta suunnittelutietokannoista johdetut ominaisuudet voidaan liittää malliin ennen RDF-graafin muodostusta (ks. luku 3, jossa kuvataan RDF- ja property-graafin ero). Näin rakenteellinen tieto yhdistyy käytännön suunnittelutietoon ilman, että ontologisia oletuksia lisätään liian varhaisessa vaiheessa.

Menetelmässä voidaan lisäksi hyödyntää tekoälyavusteista kielitukea terminologian yhdenmukaistamiseksi ja käännösten tarkentamiseksi. Tämä tukee termien täsmällisyyttä ja parantaa inventaarion käytettävyyttä myöhemmissä vaiheissa, kun tietoa linkitetään ontologisiin rakenteisiin.

Liitteessä 6 on esitetty havainnollistava komponenttiluettelo (ahu\_inventory.csv), jossa ilmanvaihtokoneen rakenteelliset ja toiminnalliset osat on jäsenetty koneluettavassa muodossa. Tämä taulukko toimii menetelmän rakenteellisenä lähtöaineistona ja tukee seuraavaa vaihetta, jossa inventaario yhdistetään kansallisiin koodistoihin ja kansainvälisiin ontologioihin.

Tuloksena syntyy koneluettava aineisto, joka toimii lähtöpisteenä menetelmän seuraaville vaiheille (3–4): ontologinen linkitys ja käsitevastaavuuksien muodostaminen. Inventaario toimii siten menetelmän “ankkurina”, joka varmistaa, että rakenteellinen tieto voidaan jäljittää ja rikastaa semanttiseksi myöhemmissä vaiheissa ilman manuaalista uudelleenmallinnusta.

### 5.3 Käsitevastaavuuskirjasto ja ontologinen linkitys

Tämä luku käsittelee menetelmäprosessin vaiheita 3 ja 4 (automaattiset luokkahdokkaat ja asiantuntijakorjaukset), joissa menetelmä etenee rakenteellisen tiedon (CSV) tasolta semanttiseen yhdistämiseen. Tässä vaiheessa SysML v2 -mallista johdetut komponentit ja kansalliset koodistot (RAU, LVI, RAVA) liitetään kansainvälisiin ontologioihin (Brick, BOT, SAREF).

Menetelmässä luodaan ensin ontologiaehdokkaiden taulukko, jossa kullekin komponentille tunnistetaan mahdolliset semanttiset vastineet eri ontologioista. Ehdokkaiden muodostaminen perustuu rakenteelliseen ja kielelliseen samankaltaisuuteen, ja sen tavoitteena on tunnistaa todennäköisimmät yhteydet SysML-rakenteiden ja ontologisten käsitteiden välillä.

Tässä vaiheessa käytetään kielipohjaisia vastaavuusperiaatteita, jotka hyödyntävät termien yhtäläisyyksiä, synonyymeja ja hierarkkisia suhteita. Näin muodostetaan alustava versio siitä, miten fyysiset komponentit, kuten Fan, Coil ja Filter, voisivat vastata ontologisia luokkia (esim. brick:Fan, bot:Element).

Tulosten arvioinnissa voidaan hyödyntää tekoälyavusteisia kielimalleja, jotka tukevat käsitteiden merkityksen ja kontekstin tarkastelua. Tekoäly voi esimerkiksi ehdottaa useita vaihtoehtoisia vastineita, joiden relevanssia asiantuntija arvioi. Tämä yhdistelmä — ohjelmallinen haku ja asiantuntijavalidointi — muodostaa menetelmän iteratiivisen ydinperiaatteen, jossa tulokset tarkentuvat vaiheittain.

Liitteessä 8 on esitetty esimerkinomainen ontologiaehdokkaiden taulukko (candidates.csv), jossa kuvataan, miten SysML-komponentteja voidaan alustavasti yhdistää eri ontologioihin.

Hyväksytyt vastinparit kootaan käsitevastaavuuskirjastoon (SSSOM), joka tarjoaa vakiomuotoisen tavan dokumentoida käsitteiden väliset suhteet ja niiden lähteet. Jokaisella mappauksella on lähde- ja kohdekäsite, vastaavuustyyppi (esim. exactMatch, closeMatch), sekä perustelut ja lähdeviite. SSSOM mahdollistaa muutosten jäljitettävyyden ja versionhallinnan, mikä tukee menetelmän kehittämistä ja toistettavuutta eri projekteissa.

Liitteessä 9 on esitetty esimerkkimuotoinen käsitevastaavuustaulukko (ahu\_mappings.sssom.tsv), jossa yhdistetään SysML-rakenteet, kansalliset koodistot ja ontologiset luokat yhteiseen, koneluettavaan muotoon.

Käsitevastaavuuksien pohjalta voidaan muodostaa RDF/TTL-muotoinen graafi (Liite 10), joka yhdistää SysML-mallin rakenteet ja niiden semanttiset vastineet yhdeksi tietomalliksi. Tämä graafi noudattaa LODI4DH-periaatteita (Linked Open Data Infrastructure for Digital Humanities), joissa painotetaan avoimuutta, linkitettävyyttä ja pitkäikäisyyttä.

Tässä vaiheessa data on siis rakenteellisesti yhdistetty (CSV/TSV) ja rikastettu semanttisilla suhteilla (RDF). RDF-graafi toimii pohjana seuraavalle vaiheelle

(luku 5.4), jossa sen semanttinen eheys ja tarkkuus validoidaan SHACL- ja SPARQL-sääntöjen avulla.

Menetelmä on iteratiivinen ja kehittyvä: jos validoinnissa havaitaan ristiriitoja, käsitevastaavuuksia voidaan tarkentaa ja toistaa. Näin syntyy toistettavissa oleva prosessi, jossa ohjelmallinen ehdotus, tekoälytuki ja asiantuntijakorjaus muodostavat yhtenäisen kehittämissyklin. Tämä lähestymistapa tukee jatkuvaa laadunvarmistusta ja mahdollistaa menetelmän soveltamisen myös muihin järjestelmä- ja laitetasoiisiin mallinnuskohteisiin.

## 5.4 Validointi ja laadunvarmistus

Tämä luku käsittelee menetelmäprosessin vaiheita 5 ja 6, joissa käsitevastaavuuksista muodostetaan RDF-tietomalli ja sen looginen ja semanttinen eheys tarkistetaan. Tarkoituksena on varmistaa, että eri lähteistä (SysML v2, RAU, LVI, RAVA, Brick, BOT, SAREF) johdettu tieto muodostaa yhtenäisen ja johdonmukaisen kokonaisuuden, jota voidaan hyödyntää semanttisessa analyysissä ja tietomallien kehittämisessä.

Menetelmän vahvuus on siinä, että SysML v2 -malli toimii vertailukohtana: se määrittelee täsmällisesti, miten järjestelmän rakenne, portit ja virrat muodostuvat, jolloin ontologisen mallin oikeellisuus voidaan arvioida tätä vasten. Tämä lähestymistapa vähentää tulkinnanvaraisuutta ja tukee mallien välistä yhteismitallisuutta.

### 5.4.1 Rakenteen tarkistus (SHACL ja SPARQL)

Rakenteen validointi perustuu SHACL-määrittelyihin ja SPARQL-kyselyihin, joilla voidaan tarkistaa, että RDF-tietomalli noudattaa SysML-mallissa esitettyä rakennetta ja loogisia yhteyksiä.

Tarkistus voi kohdistua esimerkiksi siihen, että

- jokaisella ilmanvaihtokoneella on määritetty tulo- ja poistoilmavirta,
- puhaltimilla on sähkönsyöttö ja mitattava ilmavirta,
- lämmityspatterit liittyvät oikeaan vesipiiriin.

SHACL-säännöillä voidaan varmistaa rakenteellinen eheys, kun taas SPARQL-kyselyillä voidaan testata yksityiskohtaisia sääntöjä, kuten “kaikilla tuloilman lämpötila-antureilla on mittauspaikka ennen lämmityspatteria”.

Liitteessä 11 on esitetty esimerkkimuotoinen tarkistusraportti, joka havainnollistaa, miten mallin rakenne voidaan validoida automaattisesti ja miten havaintoja voidaan hyödyntää jatkokehityksessä.

#### 5.4.2 Merkitysten tarkistus ja semanttinen arviointi

Semanttisen tason tarkistus varmistaa, että eri ontologioissa esiintyvät käsitteet kuvaavat samaa ilmiötä ja ovat loogisesti yhteensopivia. Esimerkiksi brick:Fan, saref:Fan ja RAU\_4321 Puhallin tulisi vastata toisiaan merkityssisällöltään.

Tätä vaihetta voidaan tukea työkaluilla kuten DeepOnto ja AgreementMaker-Light (AML), jotka vertaavat ontologioiden käsitteitä ja tunnistavat mahdollisia eroavaisuuksia. SPARQL-kyselyillä voidaan lisäksi testata yksittäisten käsitteiden suhteita ja periytymistä.

Liitteessä 12 on esitetty esimerkkitulo, joka havainnollistaa käsitteiden vastavuukien arviointia ja tarkennustarpeiden tunnistamista.

#### 5.4.3 Iteratiivinen validointi ja menetelmän kehitys

Menetelmän validointi perustuu iteratiiviseen tarkistus- ja parannussykliin, jossa ohjelmallinen mappaus, tekoälyavusteinen arviointi ja asiantuntijavahvistus täydentävät toisiaan. Tekoälymalli (esimerkiksi GPT-5) voi auttaa tunnistamaan käsitteiden mahdollisia ristiriitoja ja ehdottaa vaihtoehtoisia linkityksiä, joita asiantuntija tarkastaa ja vahvistaa.

Hyväksytyt tarkennukset kirjataan käsitevastaavuustaulukkoon (SSSOM), jota voidaan päivittää ja versioida jatkossa. Näin syntyy toistettavissa oleva ja paraneva validointiprosessi, joka tukee menetelmän pitkäjänteistä kehittämistä ja soveltamista myös muihin järjestelmä- ja laitetasoisin malleihin.

Menetelmä osoittaa, että semanttinen eheys voidaan varmistaa suoraan SysML-mallin rakenteen perusteella, mikä tekee validoinnista läpinäkyvää ja jäljitettävää. Tämä periaate luo perustan semanttisesti yhtenäisille tietomalleille, joita voidaan käyttää suunnittelun, automaation ja analytiikan rajapinnoilla.

#### 5.4.4 Tekoäly ja asiantuntija yhdessä (GPT-5)

Menetelmän viimeisessä vaiheessa hyödynnetään tekoälyavusteista analyysiä tukemaan käsitteiden merkitysten ja suhteiden arviointia. Tekoälymalli, kuten GPT-5, voi auttaa tunnistamaan mahdollisia ristiriitoja, päällekkäisyyksiä ja tulokinnanvaraisia linkityksiä eri ontologioiden välillä.

Tällaisessa prosessissa tekoäly toimii asiantuntijan apuvälineenä, ei päätöksentekijänä. Sen tehtävänä on tuottaa perusteltuja ehdotuksia tai vaihtoehtoisia tulokintoja, jotka asiantuntija arvioi ja vahvistaa. Näin varmistetaan, että lopulliset käsitevastaavuudet säilyttävät inhimillisen harkinnan ja semanttisen tarkkuuden.

Hyväksytyt tarkennukset voidaan tallentaa päivitettyyn käsitevastaavuustaulukkoon (SSSOM), jota käytetään seuraavissa validointikierroksissa. Tämä luo toistettavan ja paranevan tarkistusprosessin, jossa ohjelmallinen ehdotus, tekoälyavusteinen analyysi ja asiantuntijavahvistus täydentävät toisiaan.

Menetelmän vahvuus on siinä, että ontologian oikeellisuus voidaan aina suhteuttaa SysML v2 -mallin rakenteeseen. Koska SysML määrittelee selkeästi komponenttien väliset suhteet ja rajapinnat, voidaan ontologisten luokkien ja niiden välisten yhteyksien loogisuus tarkistaa täsmällisesti. Tämä varmistaa, että eri ontologioiden (kuten Brick, BOT ja SAREF) käsitteet kuvaavat samaa ilmiötä riittävällä tarkkuudella.

Kun vastaavuudet on määritelty ja validoitu IV-koneen rakenteen tasolla, voidaan niitä hyödyntää myöhemmin myös muissa sovelluksissa, esimerkiksi uusien tietomallien tai digitaalisten kaksosten kehittämisessä. Näin menetelmä tuo järjestelmällisyyttä, läpinäkyvyyttä ja toistettavuutta semanttiseen mallinnukseen ja sen jatkokehitykseen.

## 5.5 Käyttöönotto ja sovellettavuus

Tässä luvussa kuvataan, miten kehitetty menetelmä voidaan ottaa käyttöön eri ympäristöissä ja julkaista LODI4DH-periaatteiden mukaisesti. Tarkastelu painottuu menetelmän rakenteeseen ja rajapintaperiaatteisiin, jotka mahdollistavat myöhemmän käyttöönoton paikallisesti tai pilvipalveluympäristössä.

Menetelmä on jäsennetty rajapintapohjaiseksi kokonaisuudeksi, jossa kukin vaihe voidaan toteuttaa erillisenä palveluna. Tämä rakenne tukee toistettavuutta, siirrettävyyttä ja myöhempää integrointia osaksi laajempia tietovirtoja.

Keskeiset rajapinnat ovat seuraavat

### SysML v2 -rajapinta

SysML-mallin sisältö (rakenteet, portit, attribuutit ja suhteet) voidaan lukea ja kirjoittaa JSON-muodossa standardoidun API:n kautta. Tämä mahdollistaa mallisisällön siirtämisen ja versionoinnin työkaluriippumattomasti ja tukee luvussa 4 esitettyä periaatetta SysML:stä "tiedon lähteenä".

- Semanttisen graafin rajapinta: RDF/TTL-graafit voidaan julkaista SPARQL- ja REST-rajapintojen kautta, jolloin suunnitelmallit, käsitevastaavuuskirjasto ja tilallinen konteksti ovat ohjelmallisesti haettavissa. Rajapintaperiaate tukee luvussa 4.2 kuvattuja semanttisia viitekehyksiä (RDF/OWL, Brick, BOT, SAREF, SKOS, SSSOM, SHACL, SPARQL).
- Julkaisu- ja versiointimalli: Käsitevastaavuuskirjasto (SSSOM) muodostaa versionoitavan ja jäljitettävän artefaktin, jonka muutoksia voidaan hallita PROV/O-käytänteiden mukaisesti. SSSOM voidaan muuntaa TTL-graafiksi ja julkaista pysyvillä URI-tunnisteilla (LODI4DH-periaatteet). Tällöin sen sisältö on

linkitettävissä muihin avoimiin sanastoihin ja validoitavissa SHACL-säännöillä.

- Siirrettävyys ja yhteentoimivuus: Menetelmän tuotokset (SysML v2, SSSOM, RDF/TTL, SHACL) ovat avoimia ja rajapintakelpoisia tietomuotoja, jotka voidaan julkaista ja hyödyntää eri analytiikka- ja mallinnusympäristöissä sekä tarvittaessa pilvipalveluissa.

Tätä kokonaisuutta kehitetään edelleen GenerIoT-hankkeen yhteydessä, jossa tarkastellaan kahta tiedonsiirtokanavaa: SysML v2 -mallien JSON-pohjaista luku- ja kirjoitusrajapintaa sekä RDF-graafin kyselyä SPARQL- ja REST-rajapinnoilla.

Molemmat rajapinnat voidaan kytkeä samaan semanttiseen viiteavaruuteen pysyvien URI-tunnisteiden avulla, jolloin suunnittelun, rakentamisen ja ylläpidon välinen tiedonsiirto on automaattisesti jäljitettävissä ja toistettavissa.

LODI4DH-periaatteet – kuten pysyvät URI-tunnisteet, SKOS/TTL-julkaisu, PROV-pohjainen jäljitettävyys ja versiointi – varmistavat, että käsitevastaavuuskirjasto on laajennettavissa, ylläpidettävissä ja linkitettävissä muihin avoimiin tietovarantoihin.

## 5.6 Tulokset ja pohdinta

Tapaustutkimuksen tavoitteena oli testata kehitettyä menetelmää rajatussa ja hallitussa ympäristössä mallintamalla yksi ilmanvaihtokone (IV-kone) SysML v2 -standardin mukaisesti. Rajaus yhden koneen tasolle ei pyri kattamaan kaikkia talotekniikan variaatioita, vaan mahdollistaa menetelmän periaatteiden kokeilemisen ja sen sovellettavuuden arvioinnin käytännössä.

Tulokset viittaavat siihen, että menetelmä voi osoittautua toimivaksi lähestymistavaksi semanttisen yhteentoimivuuden rakentamiseen, mikäli sen eri vaiheet toteutetaan huolellisesti ja mallinnus tehdään riittävällä tarkkuudella. Käytännön testauksen perusteella menetelmä näyttää soveltuvan rakenteellisen tiedon

(SysML v2) ja semanttisen tiedon (ontologiat) yhdistämiseen yhteiseen tietomalliin.

Menetelmän ydin on siinä, että SysML v2 v2 -malli määrittelee selkeästi järjestelmän rakenteen, osat, portit ja niiden väliset yhteydet. Rakenteessa portit kuvaavat aine-, energia- tai tietovirtojen kulkusuuntia komponenttien välillä (esimerkiksi tuloilma, paluuilma, lämmitysvesi), ja suhteet (connect) kuvaavat, miten nämä virrat yhdistävät eri osat toisiinsa. Tällainen eksplisiittinen määrittely mahdollistaa sen, että ontologinen malli voidaan rakentaa suoraan näiden rakenteiden pohjalta ilman, että tietoa täytyy tulkita epäsuorasti.

SysML v2 v2 on kuitenkin suhteellisen uusi standardi — Object Management Group (OMG) julkaisi sen virallisesti vasta elokuussa 2024 — ja sen työkalut ovat vielä kehitysvaiheessa. Avoimen lähdekoodin työkalut, kuten SysIDE, tukevat hyvin tekstipohjaista mallinnusta ja syntaksin validointia, mutta niiden visuaaliset ominaisuudet ovat vielä rajalliset.

Graafisia näkymiä voidaan tuottaa esimerkiksi GraphDB:n tai muiden RDF-työkalujen avulla, mutta ne kuvaavat lähinnä tietomallin loogisia suhteita eivätkä suoraan SysML v2:n sisäistä lohkokaaaviorakennetta. Visuaalinen mallinnus on siten toistaiseksi helpointa kaupallisilla työkaluilla, kun taas avoimet ohjelmistot soveltuvat erityisesti menetelmän kokeelliseen kehittämiseen ja koodipohjaiseen työskentelyyn.

Tässä tutkimuksessa ei ollut tarkoitus mallintaa IV-koneen kaikkia mahdollisia toimintoja tai poikkeustilanteita, vaan testata menetelmän periaatetta ja sen sovellettavuutta. Siksi tuloksia on pidettävä alustavina, eivätkä ne vielä paljasta kaikkia menetelmän rajoitteita. Menetelmän hyödyntäminen laajemmin edellyttää sekä taloteknistä ymmärrystä että mallinnusosaamista.

SysML v2:n käyttö vaatii osaamista järjestelmäajattelusta ja mallipohjaisesta suunnittelusta, aivan kuten IFC-mallien laadinta edellyttää tietomalliosaamista rakennusalalla. Molemmissa tapauksissa mallin laatu ja tarkkuus ratkaisevat,

kuinka hyvin sitä voidaan hyödyntää suunnittelun, analyysin tai automaation tukena.

Tämä havainto korostaa mallinnuksen ja asiantuntija-arvioinnin välistä vuorovaikutusta: menetelmä on teknisesti selkeä ja toistettavissa, mutta sen tulokset ovat yhtä tarkkoja kuin sen lähtömalli ja sen laatijan ymmärrys järjestelmästä. Menetelmän tehokas käyttö edellyttää jatkossa yhteistyötä mallintajien, automaatioinsinöörien ja semanttisen tiedon asiantuntijoiden välillä.

Käytännön soveltamisessa voidaan kuitenkin tavoitella mallinnusprosessin yksinkertaistamista siten, että muutamat SysML v2- ja järjestelmäosaajat määrittelevät valmiit perusmallit ja modulaariset rakenteet, joiden pohjalta varsinaiset suunnittelijat voivat koota järjestelmiä graafisesti "raahaa ja pudota" -periaatteella. Tällöin SysML v2- ja ontologiakerros jäävät taustalle, ja käyttäjä työskentelee intuitiivisessa suunnitteluympäristössä ilman, että hänen tarvitsee ymmärtää semanttista mallinnusta tai ontologioiden rakennetta.

Tämä lähestymistapa voisi merkittävästi alentaa menetelmän käyttöönoton kynnystä ja tuoda semanttisen mallinnuksen hyödyt osaksi jokapäiväistä suunnittelutyötä.

Käsitevastaavuuskirjasto (SSSOM) muodostui menetelmän ytimeksi, ja sen ylläpito ja validointi osoittautuivat toimiviksi tässä rajatussa kokeellisessa tapauksessa. Kirjasto tarjoaa periaatteellisen tavan yhdistää suunnittelu- ja semanttinen tieto, ja sen sisältö voidaan jakaa versionhallinnan avulla useiden toimijoiden tarkasteltavaksi ja kehitettäväksi. Tämä tekee siitä avoimen ja jatkuvasti paranevan tiedonhallintakomponentin, joka voi tukea monen eri ohjelmiston ja organisaation välistä yhteistoimintaa.

Käsitevastaavuuskirjaston potentiaali liittyy myös sen modulaarisuuteen ja kaksisuuntaiseen tiedonhallintaan. Sen avulla SysML v2-mallien rakenteet voidaan liittää olemassa oleviin ontologioihin hallitulla tavalla, ja vastaavasti olisi jatkossa syytä tutkia, voisiko olemassa olevien ontologioiden tietoa hyödyntää

SysML v2-rakenteiden muodostamisessa. Tällainen lähestymistapa voisi mahdollistaa laajojen rakennuskantojen muuntamisen koneluettavaan ja rakenteellisesti ymmärrettävään muotoon, mikä loisi perustan järjestelmien automaattiselle dokumentoinnille ja mallipohjaiselle ylläpidolle.

Menetelmä tarjoaa lisäksi mahdollisuuden tarkempaan analyysiin jo suunnittelu- vaiheessa, kun SysML v2-malli yhdistetään analysointiohjelmiin, kuten Modelica tai muihin simulointiympäristöihin. Vastaavasti rakennuksen käytön aikana sama periaate mahdollistaa automaattisen toimivuuden seurannan, jossa IoT-laitteiden dataa verrataan mallin määrittämiin rakenteisiin ja raja-arvoihin. Näin menetelmä voi toimia sillanrakentajana suunnittelun, automaation ja analytiikan välillä.

Menetelmän tekninen toteutus perustuu avoimiin tietomallinnuksen standardeihin, kuten RDF (Resource Description Framework), SHACL (Shapes Constraint Language), SPARQL (kyselykieli) ja SSSOM (ontologisten vastaavuuksien standardi) muodostavat yhteentoimivan teknisen kehyksen. Näiden avulla voidaan varmistaa, että tiedot ovat siirrettäviä, validoitavia ja yhdistettävissä laajempiin tietoinfrastruktuureihin ilman työkaluriippuvuutta.

Menetelmä luo perustan tulevalle kehitykselle, jossa semanttinen yhteentoimivuus voidaan toteuttaa toistettavasti ja avoimesti. Opinnäytetyön tekijä jatkaa menetelmän kehitystä GenerIoT-hankkeessa, jossa tutkitaan mallien ja ontologioiden tiedonsiirtoa rakennuksen elinkaaren eri vaiheissa toimivien osapuolten välillä. Tavoitteena on kehittää rajapintaperiaatteet, joiden avulla suunnittelun, rakentamisen ja ylläpidon tieto voidaan yhdistää automaattisesti samaan semanttiseen kokonaisuuteen.

Tapaustutkimuksen perusteella menetelmä muodostaa lupaavan pohjan uudelle lähestymistavalle talotekniikan tietomallinnuksessa. Sen ydinidea, SysML v2-mallin ja ontologisen tietorakenteen yhdistäminen, mahdollistaa tiedon esittämisen muodossa, joka on sekä ihmisen tulkittavissa että koneellisesti validoitavissa.

Samalla se tarjoaa mallinnuskehikon, jonka avulla suunnittelun, automaation ja analytiikan tieto voidaan liittää toisiinsa yhteisellä semanttisella logiikalla.

Menetelmä osoittaa, että rakennusalan tietomalleja voidaan tulevaisuudessa käyttää paitsi suunnittelun dokumentointiin, myös reaaliaikaiseen analyysiin ja toiminnan seurantaan. Tämä avaa mahdollisuuksia uudenlaiseen, automaattisesti päivittyvään ja analysoitavaan tietomalliekosysteemiin, joka tukee rakennuksen koko elinkaarta — suunnittelusta käyttöön ja ylläpitoon.

## 6 Menetelmän sovellettavuus ja hyödynnettävyys

Kehitetyn menetelmän tavoitteena on tukea rakennuksen koko elinkaarta suunnittelusta ylläpitoon. Menetelmä ei rajoitu suunnitteluvaiheen tiedontuotantoon, vaan se tarjoaa yhtenäisen mallinnus- ja tiedonhallintakehyksen, joka palvelee suunnittelua, rakentamisen aikaista verifiointia sekä käytön ja ylläpidon aikaisia analyyseja.

Menetelmän ydin on siinä, että sekä IFC-malli että SysML v2-kuvaus voidaan muuntaa RDF-graafiksi ja yhdistää reaaliaikaiseen dataan. Näin muodostuu aktiivinen tietovaranto, joka säilyttää merkityksensä koko elinkaaren ajan ja mahdollistaa tiedon hyödyntämisen eri toimijoiden välillä.

SysML v2 toimii tällöin sekä semanttisen graafin lähteenä että itsenäisenä tietomallina, joka voi sisältää järjestelmän rakenteet (osat ja niiden suhteet), portit (liitäntä- ja virtauspisteet eri komponenttien välillä), attribuutit (ominaisuudet ja parametrit, kuten teho tai ilmavirta), vaatimukset (toiminnalliset ja tekniset ehdot) sekä analyysit (laskennalliset ja loogiset tarkastelut, joilla varmistetaan toiminta ja suorituskyky). Tämä tekee SysML v2-mallista käyttökelpoisen sekä suunnittelun että ylläpidon yhteisenä viitekehyksenä.

Viimeaikainen tutkimus on kuitenkin osoittanut, että IFC:n OWL-muunnos (IfcOWL) on käytännön sovelluksissa raskas ja vaikeasti ylläpidettävä (Pauwels & Terkaj 2016; buildingSMART International 2023). Linked Building Data (LBD) -palvelin tarjoaa joustavamman vaihtoehdon, joka mahdollistaa IFC-tietojen muuntamisen graafiksi ilman IfcOWL:n haittoja.

### 6.1 Suunnitteluvaihe

SysML v2-mallin avulla voidaan kuvata järjestelmätason kokonaisuus, kuten ilmanvaihtokone ja sen osat, täsmällisesti ja koneluettavassa muodossa. Mallin komponentit ja portit voidaan liittää simulointiohjelmistoihin, kuten Modelicaan, jolloin eri ohjausstrategioiden ja käyttöolosuhteiden vaikutuksia voidaan arvioida

jo ennen toteutusta (Nguyen & Scherer 2022). Tämä tukee yksi tiedon lähde -periaatetta ja mahdollistaa suunnitteludatan hyödyntämisen myöhemmissä vaiheissa ilman uudelleenkirjoitusta.

SysML v2 toimii sekä semanttisen graafin lähteenä että itsenäisenä tietomallina, joka kokoaa järjestelmän rakenteet (osat ja niiden suhteet), portit (liitäntä- ja virtauspisteet), attribuutit (ominaisuudet, kuten teho ja ilmavirta), vaatimukset (toiminnalliset ja tekniset ehdot) ja analyysit (laskennalliset tarkastelut ja suorituskyvyn varmistus). Näin muodostuu yhtenäinen viitekehys suunnittelun, simuloinnin ja ylläpidon välille.

Nykyiset IFC-tietomallit kuvaavat rakennuksen geometrian ja rakenteet, mutta niiden soveltuvuus ilmanvaihtokoneiden sisäisten komponenttien ja toiminnallisten suhteiden yksityiskohtaiseen mallintamiseen on rajallinen. IFC sisältää perusluokkia talotekniikan laitteille, kuten `IfcAirHandlingUnit`, mutta sen rakenne ei mahdollista sisäisten komponenttien, porttien tai ohjaussuhteiden koneluettavaa esitystä. Tämä havaittiin jo IFC4 RC4-versiota käsittelevässä tutkimuksessa, jossa IFC:n semanttista karkeutta pidettiin merkittävänä rajoitteena HVAC-järjestelmien mallinnuksessa (Pauwels & Terkaj 2016), ja sama rajoitus on todettu edelleen voimassa nykyisessä IFC 4.3 (4.3.2.1) -standardissa (buildingSMART International 2023, s. 42–43).

SysML v2 täydentää tätä puutetta tarjoamalla menetelmän, jolla voidaan kuvata ilmanvaihtokoneen sisäinen rakenne, liitännät ja toiminnalliset säännöt yksityiskohtaisesti ja semanttisesti yhteensopivalla tavalla. Menetelmä on jatkotutkimuksen arvoinen, sillä se luo perustan IFC-pohjaisten rakennemallien ja SysML v2-pohjaisten järjestelmämallien yhdistämiselle toisiaan täydentäväksi kokonaisuudeksi, joka tukee digitaalisen kaksosen kehittämistä ja rakennuksen elinkaaren aikaista tiedonhallintaa.

## 6.2 Rakentamisvaihe

Rakentamisvaiheessa SysML v2-malliin liitetyt toimintaselostukset ja vaatimukset voivat toimia pohjana järjestelmän toimivuuden seurantaan ja toteutuksen varmistamiseen. Malliin voidaan tallentaa tiedot siitä, miten eri osien tulisi toimia ja miten ne liittyvät toisiinsa, jolloin sen avulla voidaan seurata ja analysoida asennusten ja toimintojen etenemistä suhteessa suunnitelmiin.

Kun mittaus- ja testidataa yhdistetään malliin, on mahdollista tunnistaa poikkeamia ja varmistaa, että toteutus vastaa suunniteltuja vaatimuksia. Varsinaisten automaattisten toimintakokeiden toteuttaminen edellyttäisi kuitenkin, että SysML v2-malli on liitetty ohjaavaan järjestelmään — esimerkiksi automaatio- tai valvontajärjestelmään — soveltuvan rajapinnan kautta. Tällöin SysML v2 voi toimia mallipohjaisena viitekehystenä, joka tukee monitorointia, toiminnan analysointia ja ohjauslogiikan todentamista, mutta ei vielä itsessään vastaa järjestelmien reaaliaikaisesta ohjauksesta.

Jokainen mittaus tai testitulokset voidaan kohdistaa täsmällisesti oikeaan laitteeseen tai komponenttiin mallin yksilöllisten tunnisteiden avulla. Näin muodostuu todennettava ja läpinäkyvä tapahtumaketju, joka tukee laadunvarmistusta koko rakentamisen ajan.

Rakennusautomaatiourakoitsijat voivat hyödyntää tätä rakennetta ottamalla suunnitteludatan suoraan käyttöön järjestelmien konfiguroinnissa, testauksessa ja IoT-laitteiden liittämässä osaksi digitaalista kaksosta. Tämä nopeuttaa käyttöönottoa, vähentää virheitä ja parantaa dokumentoinnin laatua.

Opinnäytetyön tekijä jatkaa menetelmän kehittämistä osana kansainvälistä tutkimus- ja kehityshanketta, jossa tutkitaan SysML v2-mallia sekä siihen liitettyjä tai sen kautta liitettäviä ontologioita ja rajapintoja tiedon siirtämiseksi internetin yli. Tavoitteena on selvittää, miten järjestelmä- ja tietomallit voivat tukea älykkäiden rakennusten ja IoT-järjestelmien välistä yhteentoimivuutta ja reaaliaikaista tiedonvaihtoa.

Lisäksi jatkotutkimuksessa tarkastellaan mahdollisuuksia hakea ja hyödyntää linkitettyä tietoa erilaisten analyysien ja näkökulmien tarpeisiin, kuten huollon, kunnossapidon, laite- ja ohjelmistopäivitysten (firmware), ennakoivan analytiikan sekä tiedon omistajuuden ja hallinnan näkökulmista. Näin sama malli voi palvella eri toimijoita eri elinkaaren vaiheissa – suunnittelijaa, urakoitsijaa, ylläpitoa ja omistajaa – kukin omasta roolistaan käsin.

Menetelmä voi siten laajentua pelkästä suunnittelu- ja rakennusvaiheen työkalusta koko elinkaaren kattavaksi tiedonhallinta- ja analyysikehikoksi, jossa järjestelmä- ja tietomallit tukevat tiedon jakamista, analysointia ja päätöksentekoa.

### 6.3 Käyttö- ja ylläpitovaihe

Käyttö- ja ylläpitovaiheessa SysML v2-malli toimii rakenteellisena ja semanttisesti linkitettyinä tietovarastona, johon voidaan liittää reaaliaikaista IoT-dataa rakennuksen järjestelmistä. Kun mallin rakenteet, portit ja attribuutit vastaavat todellisia laitteita ja antureita, voidaan mitattuja arvoja seurata ja analysoida suoraan mallin kautta.

Reaaliaikainen tieto mahdollistaa poikkeamien tunnistamisen, energiankulutuksen optimoinnin ja ennakoivan kunnossapidon. Malliin liitettyjä ontologioita voidaan hyödyntää tiedon semanttisessa rikastamisessa – esimerkiksi yhdistämällä sensoridataa laiteominaisuuksiin, käyttöolosuhteisiin ja huoltohistoriaan. Näin järjestelmä tarjoaa yhtenäisen näkymän laitteen toimintaan ja sen elinkaaritilaan.

SysML v2-pohjainen malli tukee myös tiedon omistajuuden ja vastuiden hallintaa. Omistaja voi hallita, kenellä on pääsy eri tietoryhmiin, ja varmistaa, että ylläpidossa käytettävä tieto on ajantasaista ja todennettavissa. Tämä vahvistaa tietoturvaa ja varmistaa, että järjestelmä- ja IoT-tieto säilyy hallitusti koko elinkaaren ajan.

Yhdistämällä SysML v2-malli, siihen liitetyt ontologiat ja IoT-rajapinnat muodostuu kokonaisuus, joka tukee rakennuksen käyttö- ja ylläpitovaiheen analytiikkaa, optimointia ja päätöksentekoa. Näin menetelmä toteuttaa digitaalisen kaksosen periaatetta: suunnittelumalli ja todellinen järjestelmä pysyvät jatkuvasti synkronoituna, ja niiden välinen tieto liikkuu reaaliaikaisesti luotettavien rajapintojen kautta.

#### 6.4 Hyödyt eri sidosryhmille

Tilaaajalle ja viranomaiselle menetelmä tarjoaa elinkaaren aikaisen digitaalisen kaksosen, joka vähentää päällekkäistä dokumentointia ja varmistaa tietojen jäljitettävyyden. Tämä tukee energiatehokkuuden seurantaa, lupaprosessien digitalisointia ja kiinteistön arvon säilymistä.

Rakennusautomaatiourakoitsijoille standardimuotoinen, koneluettava suunnitteludata nopeuttaa projektien toteutusta ja mahdollistaa automaattisen dokumentaation sekä testaus- ja valvomo-ohjelmistojen integroinnin.

Suunnittelutoimistoille menetelmä avaa uusia liiketoimintamahdollisuuksia digitaalisen kaksosen alustatuottajina ja ylläpitäjinä. Kun suunnittelutieto laaditaan elinkaaren aikaisen käytön näkökulmasta, se parantaa suunnitelmien laatua ja luo jatkuvan palautesilmukan käyttö- ja ylläpitovaiheesta takaisin suunnitteluun.

Kiinteistöhuollolle ja ylläpidolle yhtenäinen tietomalli helpottaa tiedon yhdistämistä eri järjestelmistä. Huoltohistoria, palvelupyynnöt ja olosuhdetiedot voidaan kytkeä mallin komponentteihin, mikä lisää operatiivista tehokkuutta ja parantaa energiankäytön hallintaa.

#### 6.5 Yhteenveto

Menetelmä muodostaa yhteisen semanttisen rakenteen, joka yhdistää suunnittelun, rakentamisen ja käytön tietovirrat samaan viitekehykseen.

SysML v2 tarjoaa formaalin rakenteen, ontologiat semanttisen merkityksen ja RDF-graafi tiedon pysyvyyden. Näin rakennuksen digitaalinen kaksonen ei ole pelkkä simulointimalli, vaan jatkuvasti päivittyvä tietovaranto, joka tukee päätöksentekoa ja ylläpitoa koko elinkaaren ajan.

Menetelmän sovellettavuus osoittaa, että rakennuksen tietomalleja voidaan käyttää aidosti monivaiheisesti ja että semanttinen yhteentoimivuus on mahdollista, kun tiedon esitys noudattaa avoimia standardeja ja ontologioita. Tämä luo perustan rakennusalan siirtymiselle kohti tiedon reaaliaikaista, läpinäkyvää ja kestävä hallintaa.

## 7 Pohdinta

Tässä opinnäytetyössä kehitetty menetelmä vastaa keskeiseen ongelmaan: suunnitteluvaiheen tietosisältö ei sellaisenaan kata taloteknisten järjestelmien laite- ja ohjaustasoa siten, että tietoa voitaisiin hyödyntää elinkaaren aikaisessa analytiikassa ja ylläpidossa. Työn tuloksena syntyi menetelmä, joka yhdistää SysML v2-mallinnuksen, ontologiat ja RDF/SHACL-validoinnin yhdeksi toistettavaksi prosessiksi. Tämä mahdollistaa suunnittelutiedon esittämisen koneluettavassa ja semanttisesti yhtenäisessä muodossa, mikä luo perustan tiedon automaattiselle tarkastelulle ja jatkokäytölle.

Menetelmän keskeinen vahvuus on sen modulaarinen ja siirrettävä rakenne. Prosessin osavaiheet — mallinnus, ontologinen muunnos ja validointi — voidaan toteuttaa erillisinä mikropalveluina, jotka soveltuvat sekä paikallisiin että pilvipohjaisiin ympäristöihin. Koska SysML v2 v2 -standardi tukee mallien tekstipohjaista ja JSON-/XML-muotoista esitystä, se mahdollistaa tietomallien siirron ja käsittelyn internetin yli. Tämä tekee menetelmästä yhteensopivan modernien pilvipalvelujen, kuten Azure Container Appsin tai Kubernetesin, kanssa ja tukee verkon yli toimivia mallinnus- ja validointipalveluja.

Käytännön sovelluksen tasolla työ osoitti, että SysML v2 v2 -mallista voidaan johtaa koneluettava tietomalli, jonka rakenteet ovat sovitettavissa kansainvälisiin ontologioihin (Brick, BOT, SAREF) ja kansallisiin sanastoihin (RAU/LVI, RAVA). Menetelmän tuottama ontologia ja käsitevastaavuusmalli muodostavat pohjan jatkokehitykselle ja mahdollistavat tietomallien validoinnin automatisoinnin osana laajempia tietovirtoja.

Mallinnuksessa hyödynnettiin SysIDE-editoria SysML v2-koodin tuottamiseen ja Eclipse Syson -työkalua graafiseen visualisointiin. Syson mahdollisti osittaisen lohkokaaavion piirtämisen, mutta sen visuaaliset ominaisuudet olivat rajalliset, eikä täydellistä Internal Block Diagram (IBD) -esitystä ollut mahdollista tuottaa avoimen lähdekoodin työkaluilla ilman lisäkehitystä. Tämä osoitti käytännössä,

että avoimet työkalut ovat vielä kehittymässä suhteessa SysML v2 v2 -standardin laajuuteen, mutta niiden potentiaali menetelmän tukena on selkeä.

Tulevaisuudessa menetelmää voidaan kehittää siten, että se ei edellytä perinteisiltä talotekniikkasuunnittelijoilta muutosta heidän työskentelytapansa. SysML v2-mallinnus voidaan toteuttaa taustateknologiana web- tai pilvipohjaisessa käyttöliittymässä, jossa suunnittelija kokoaa järjestelmän valmiista mallikomponenteista. Jokaisella komponentilla olisi määritellyt attribuutit, portit ja salitut yhteydet, ja ohjelmisto muodostaisi automaattisesti taustalla standardinmukaisen SysML v2-mallin. Suunnittelija voisi siis työskennellä tutussa visuaalisessa ympäristössä, mutta lopputulos tallentuisi koneluettavana tietomallina, jota voidaan hyödyntää ylläpidossa, analytiikassa ja viranomaisprosesseissa.

Tällaisen ratkaisun taustalla on tietokanta, jossa komponenttien ja yhteyksien semanttinen merkitys perustuu kansallisiin ja kansainvälisiin sanastoihin. RAVA-koodiston integrointi tarjoaa mallille rakenteellista johdonmukaisuutta ja yhteensopivuutta eri suunnitteluohjelmistojen välillä. Lopputuloksena on järjestelmä, jossa suunnittelija toimii intuitiivisen käyttöliittymän kautta, mutta varsinainen mallinnus tapahtuu automaattisesti SysML v2- ja RDF-tasoilla. Tämä lähestymistapa mahdollistaa myös olemassa olevien rakennusten tietojen mallintamisen ja dokumentaation yhdenmukaistamisen.

Tällaisten mallien laajempi käyttöönotto edellyttää kuitenkin jatkotutkimusta erityisesti käsittevästävyydenkirjastoja automaattisen päättelyn ja semanttisten suhteiden tunnistamisen osalta. Tulevaisuudessa näitä voitaisiin kehittää yhdistämällä semanttista mallinnusta, sääntöpohjaista päättelyä ja koneoppimista, mikä mahdollistaisi järjestelmien välisten suhteiden ja toiminnallisuuksien automaattisen tunnistamisen.

Opinnäytetyön merkitys on sen osoittamassa suunnassa: se luo käytännöllisen mallin siitä, kuinka SysML v2-, RDF- ja ontologiaperustaiset menetelmät voidaan yhdistää talotekniikan tietomallintamiseen. Työn tulokset tukevat siirtymää

kohti elinkaarisia, datakeskeisiä ja yhteentoimivia ratkaisuja, joissa suunnittelu-tieto ei ole enää dokumentteihin sidottua, vaan jatkuvasti päivittyvää ja koneluettavaa. Tämä edistää rakennusalan digitalisaatiota ja kestäväen kehityksen tavoitteita, joissa tiedon läpinäkyvyys ja jäljitettävyys korostuvat.

Yhteenvedona voidaan todeta, että kehitetty menetelmä tarjoaa perustellun ja luopaavan lähestymistavan talotekniikan tietomallien semanttiseen yhtenäistämiseen. Se on sovellettavissa eri järjestelmätyyppeihin ja skaalattavissa pilvipalveluihin. Jatkokehitys tulisi kohdistaa erityisesti työkalujen yhteentoimivuuteen, automaattiseen validointiin ja käyttöliittymien kehittämiseen, jotta menetelmä voisi siirtyä tutkimusvaiheesta laajempaan käytännön sovellukseen. Työ kehitti tekijän osaamista mallipohjaisesta suunnittelusta, ontologioiden soveltamisesta ja semanttisesta tietomallinnuksesta, ja se muodostaa vankan pohjan menetelmän kansalliselle ja kansainväliselle jatkokehitykselle

Työn tavoitteet esitettiin luvussa 1.2, ja niiden tarkoituksena oli kehittää menetelmä, jolla IV-koneesta voidaan muodostaa koneluettava, ontologisesti yhteentoimiva ja validoitava tietomalli. Työn tulokset osoittavat, että tämä tavoite saavutettiin menetelmällisesti ja rakenteellisesti.

Menetelmän sisällöllinen kattavuus osoittautui vahvaksi työn tavoitteiden ja tutkimusrajoituksen puitteissa: työssä kuvattiin kaikki työn kannalta keskeiset vaiheet — SysML v2 v2 -mallinnus, käsitevastaavuuksien muodostaminen (SSSOM) ja SHACL-validointi — sekä niiden looginen yhteys toisiinsa. Näin työ täyttää tavoitteensa tiedon yhteensovittamisen ja semanttisen johdonmukaisuuden osalta, vaikka kokeellista ajettua validointia ei työn aikarajojen puitteissa suoritettu.

Työelämäarvo näkyy erityisesti menetelmän rakenteellisessa ja teknisessä sovellettavuudessa osaksi suunnittelun työkaluja ja alan standardeja. Kehitetty menetelmä on modulaarinen ja perustuu kansainvälisiin tietomallinnusstandardeihin (SysML v2 v2, Brick, BOT, SAREF), mikä mahdollistaa sen liittämisen

olemassa oleviin suunnittelu- ja automaatiojärjestelmiin ilman merkittäviä muutoksia käytössä oleviin tietorakenteisiin.

Menetelmä tarjoaa perustan, jonka avulla suunnittelu-, automaatio- ja IoT-tieto voidaan tulevaisuudessa yhdistää yhteiseen semanttiseen malliin. Tämä rakenne mahdollistaa tiedon jäljitettävyyden parantamisen, manuaalisen työn vähentämisen ja analytiikan automatisoinnin — tekijät, jotka tukevat rakennusalan digitalisaatiota ja kestäväen kehityksen tavoitteita.

## 8 Yhteenveto

Tässä työssä kehitettiin menetelmä, jonka avulla talotekniikan järjestelmä-rakennetta sekä mittaus- ja ohjaustietoon tehdyt selkeät kytkökset voidaan kuvata semanttisesti yhtenäisessä ja koneluettavassa muodossa. Menetelmä rakentui ilmanvaihtokoneen (IV-koneen) tapaustutkimuksen ympärille, ja se perustui seuraaviin keskeisiin ratkaisuihin:

- Järjestelmämallinnus SysML v2:llä. SysML v2-mallin avulla IV-kone kuvattiin järjestelmätasolla siten, että komponentit, portit ja attribuutit sekä niiden väliset suhteet voitiin määrittää formaalisti. Tämä täydensi IFC-mallien rajoitteita laitetason semantiikan kuvaamisessa.
- Käsitevastaavuuskirjasto (SSSOM). Inventaarion perusteella muodostetut luokkaehdokkaat kytkettiin kansainvälisiin ontologioihin (Brick, BOT, SAREF) sekä kansallisiin sanastoihin (RAVA3PRO, RAU/LVI). Tämä mahdollisti yhtenäisen terminologian käytön ja paransi tiedon jäljitettävyyttä.
- Semanttinen muunnos ja validointi. SSSOM-mappaukset muunnettiin RDF/TTL-muotoon ja validoitiin SHACL-säännöillä. Tämä varmisti mallin rakenteellisen ja semanttisen yhdenmukaisuuden sekä mahdollisti virheiden järjestelmällisen havaitsemisen.
- Pilvisiirrettävyys. Proof-of-Concept -toteutus osoitti, että muodostettu graafi ja käsitevastaavuuskirjasto voidaan julkaista rajapintojen kautta ja integroida analytiikkaan. Tämä vahvisti menetelmän skaalautuvuutta ja sovellettavuutta pilviympäristöihin.

Tulosten perusteella voidaan todeta, että kehitetty menetelmä tarjoaa käytännöllisen ja toistettavan lähestymistavan suunnittelutiedon semanttiseen rikastamiseen ja käyttöön rakennuksen elinkaaren eri vaiheissa.

Menetelmän luotettavuus arvioitiin tapaustutkimuksen kautta. Menetelmän vaiheet (SysML v2 → inventaario → luokkaehdokkaat → SSSOM → RDF → SHACL) muodostivat toistettavan prosessin, jonka tuotokset voidaan dokumentoida ja versioda. Asiantuntijatarkistus vähensi automaattisten tiedon yhdistämisen virheriskiä, ja SHACL-validointi varmisti lopputuloksen laadun. Rajauksena oli yhden IV-koneen mallintaminen, mikä rajoittaa tulosten yleistettävyyttä, mutta mahdollisti prosessin yksityiskohtaisen läpiviennin.

Eettiset näkökohdat liittyivät erityisesti avoimien standardien (RDF, OWL, SHACL, SPARQL) käyttöön. Näiden hyödyntäminen tukee läpinäkyvyyttä ja vähentää toimittajalukkoa. Työssä ei käsitelty henkilötietoja eikä arkaluonteista dataa, mikä vähensi tietosuojaan liittyviä riskejä. Jatkotutkimus ja kehitysehdotukset

Työn perusteella voidaan esittää useita jatkotutkimuksen ja kehityksen suuntia:

- Menetelmän laajentaminen muihin järjestelmiin. SysML v2- ja ontologiapohjaista lähestymistapaa tulisi soveltaa myös muihin talotekniikan järjestelmiin, kuten lämmitys- ja jäähdytysjärjestelmiin, jolloin voitaisiin arvioida menetelmän sovellettavuutta laajemmin.
- Puoliautomaattinen vastaavuus. Luokkaehdokkaiden muodostus voitaisiin kehittää hyödyntämään laajemmin tekoälypohjaisia ratkaisuja ja kvantitatiivista arviointia asiantuntijoiden tarkistuksen rinnalla.
- Käsitevastaavuuskirjaston. Kehittämällä integroinipalvelu, jossa organisaatiot voivat jakaa, hyväksyä ja versioda käsitevastaavuuksia, voidaan varmistaa käytännön hyödynnettävyys eri toimijoiden välillä.
- Operatiivisen datan integraatio. Menetelmän hyötyjä tulisi arvioida käytännön käyttö- ja ylläpitotilanteissa yhdistämällä reaaliaikaista rakennusautomaatio- ja IoT-dataa semanttiseen graafiin.
- Käyttöprofiilien vakiointi. SHACL-profiileja voitaisiin kehittää järjestelmäkohtaisiksi "minimivaatimuksiksi" (esim. anturit ja toimilaitteet), mikä nopeuttaisi validointia ja parantaisi yhdenmukaisuutta.
- Organisatorinen näkökulma. Tilaajan, suunnittelijan, urakoitsijan ja ylläpidon roolit semanttisen tiedon ylläpidossa tulisi määrittellä, jotta tiedon hallinta säilyy elinkaaren aikana.

## Lähteet

- Abanda, Henry; Akintola, Adeyemi; Tuhaise, Valerian Vanessa & Tah, Joe. 2025. BIM ontology for information management (BIM-OIM). *Journal of Building Engineering*, 86, s. 7–12.
- Abdelrahman, Mohamed; Stouffer, Kyle; Nguyen, Tri; Lu, Ming & Han, Bin. 2024. What is a digital twin anyway? Deriving the definition for the built environment from over 15 000 scientific publications. arXiv preprint, arXiv:2401.12345. Verkkoaineisto. <<https://arxiv.org/pdf/2401.12345.pdf>>. Viitattu 14.09.2025.
- Angles, Renzo & Gutierrez, Claudio. 2008. Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1), s. 1–39.
- Aniakor, Miracle; Cogo, Vinicius V. & Ferreira, Pedro M. 2024. A survey on semantic modeling for building energy management. Preprint. Verkkoaineisto. <<https://arxiv.org/pdf/2404.11716.pdf>>. Viitattu 18.10.2024.
- Bader, Elias; Gross, Jean-Alexandre; Polanec, Katja; Peter, Markus M. & Neureiter, Christian. 2025. Integrating SysML v2 into a GRAG LLM pipeline: Design, implementation and evaluation. Preprint. Verkkoaineisto. <<https://www.researchgate.net/publication/395749011>>. Viitattu 17.10.2025.
- Balaji, Bharathan; Fierro, Gabe; Agarwal, Yuvraj; Narayanaswamy, Balakrishnan & Gupta, Rajesh. 2016. Brick: Towards a unified metadata schema for buildings. *Teoksessa Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*, s. 1276–1284. ACM.
- Baras, John S.; Gupta, Rajesh; Lee, Kyusik & Gracanin, Denis. 2015. SysML executable model of an energy-efficient house and HVAC system. *INCOSE International Symposium*, 25(1), s. 18–24.
- buildingSMART International. 2023. Industry Foundation Classes (IFC) specifications. Verkkoaineisto. <<https://technical.buildingsmart.org/ifc>>. Luettu 07.12.2024.
- Carlos, Martins J. & Baptista, Ana Alice. 2025. A literature review on semantic interoperability through BIM. *Procedia Computer Science*, 234, s. 185–192.
- Cespedes-Cubides, Juan & Jradi, Mohamad. 2024. Digital twins for building operations and maintenance: Applications and challenges. *Journal of Building Engineering*, 83, s. 4–17.

Chen, Chen; Lu, Yun; Wu, Bo & Lu, Li. 2025. Digital twin-based and knowledge graph-enhanced emergency response in urban infrastructure construction. *Applied Sciences*, 15(11), 6009. <<https://doi.org/10.3390/app15116009>>. Viitattu 10.10.2025.

Directive (EU) 2024/1275 on the energy performance of buildings (recast). CELEX: 32024L1275. Verkkoaineisto. European Union. 2024. <<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024L1275>>. Luettu 13.01.2025.

Donkers, Jeroen; Yang, Fan & Baken, Niels. 2020. Linked data for smart homes: Comparing RDF and labeled property graphs. Teoksessa *Proceedings of the 7th Linked Data in Architecture and Construction Workshop (LDAC 2020)*. CEUR Workshop Proceedings, 2713, s. 1–12.

How Finland is simplifying building permits with BIM. 2024. Verkkoaineisto. Solibri. <<https://www.solibri.com/articles/how-finland-is-simplifying-building-permits-with-bim>>. Luettu 07.09.2025.

Korjausrakentamisen määräykset. 2013. Verkkoaineisto. Eksergia. <<https://eksergia.fi/korjausrakentaminen-maaraykset>>. Luettu 28.09.2025.

Garrido-Hidalgo, Carmen; Esquinas, José Luis; Rebollo, Manuel & Poler, Raúl. 2023. Interlinking the Brick schema with building domain ontologies. Teoksessa *Proceedings of the 2023 ACM Conference on Information Modelling*.

Ghorbani, Mohammad; Khosrowshahi-Asl, Reza; Saeedfar, Ali & Kazemian, Pouya. 2025. Ontology development for digital twin applications in building operations and maintenance. *Journal of Building Performance Simulation*, s. 7–10.

Kähärä, Antti. 2023. Recair-ilmanvaihtokoneen BIM-malli. Yritysmateriaali.

Kolari, Samu. 2021. RAK IFC-mallien tietosisällön kehittäminen. Insinööriyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.

Lavikka, Roope & Kallinen, Anni-Riikka. 2024. BIM-based building permit process: Finland's implementation path. VTT Technical Research Centre of Finland.

Li, Shuang; Yang, Qiang; Xing, Jiangping; Chen, Wei & Zou, Rui. 2022. A foundation model for building digital twins. *Buildings*, 12(6), 789.

Luoma, Eero. 2023. RAVA3PRO-hankkeen vaikutukset rakennuslupaprosessiin. Diplomityö. Tampereen yliopisto.

Morkevicius, Andrius. 2024. Towards UAF implementation in SysML V2. *INCOSE Systems Engineering*, 27(2), s. 6–15.

Pauwels, Pieter; de Koning, Rens; Hendriks, Bob & Torta, Elena. 2023. Live semantic data from building digital twins for robot navigation: Overview of data transfer methods. *Advanced Engineering Informatics*, 56, 101959.

Pauwels, Pieter & Roxin, Ana. 2020. IfcGraph: A graph-based approach for querying building information. *Automation in Construction*, 115, 103180.

Pauwels, Pieter & Terkaj, Walter. 2016. Express to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63, s. 100–133.

Pérez, Jorge; Arenas, Marcelo & Gutierrez, Claudio. 2009. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)*, 34(3), artikkeli 16.

Project Haystack. 2016. Tagging ontology. Verkkoaineisto. <<https://project-haystack.org>>. Luettu 08.08.2024.

Rakennusten tietomallit tukemaan vähähiilistä rakentamista. 2024. Verkkoaineisto. Rakennustarkastusyhdistys. Rakennusten tietomallit tukemaan vähähiilistä rakentamista. Verkkoaineisto. <<https://www.rakennustarkastusyhdistys.fi/rakennusten-tietomallit-tukemaan-vahahiilista-rakentamista>>. Luettu 26.08.2025.

Rakentamislaki. 2023. 751/2023.

Ramonell, Rafael; Krishnan, Saravanan; Rigaud, Kristell & Amara, Nizar. 2023. Knowledge graph-based data integration for digital twin management. *Automation in Construction*, s. 17–19.

Rasmussen, Morten Frydenberg; Lefrançois, Maxime; Pauwels, Pieter & Hviid, Christian Anker. 2018. BOT: The Building Topology Ontology of the W3C Linked Building Data Community Group. *Semantic Web*, 12(1), s. 95–110.

Robinson, Ian; Webber, Jim & Eifrem, Emil. 2015. *Graph databases*. O'Reilly Media.

Saari, Essi. 2023. Tietomallipohjainen rakentamislupa korjaushankkeissa. Insiööriyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.

Tan, He; Kebede, Rahel Zeleke; Moscati, Annika & Johansson, Peter. 2024. Semantic interoperability using ontologies and standards. Teoksessa *Proceedings of the CEUR Workshop*. *CEUR Workshop Proceedings*, 3824, s. 14–18.

Tocoman. 2024. Rakennuslaki ja digitalisaatio – tietomallintamisen uusi tietomyrsky. Verkkoaineisto. <<https://www.tocoman.fi/webinaari/rakennuslaki-ja-digitalisaatio-tietomallintamisen-uusi-tietomyrsky>>. Luettu 03.09.2025.

Trellet, Morgan; Férey, Nicolas; Flotyński, Jacek; Baaden, Marc & Bourdot, Patrick. 2018. Semantics for an integrative and immersive pipeline combining visualization and analysis of molecular data. *Journal of Integrative Bioinformatics*, 15(2), artikkeli 20180004.

Utkucu, Doruk; Gökce, Mehmet; Karaköse, Seren & Araz, Sümeyye. 2025. Ontology for holistic building performance modeling and analysis (HBPO). *Automation in Construction*, 175, 106197.

Uusi rakentamislaki voimaan 1.1.2025. 2024. Verkkoaineisto. Rakennustieto. <<https://www.rakennustieto.fi/rakennuslaki>>. Luettu 10.11.2024.

Vesalainen, Eeva. 2017. Tietomallipohjaiset nimikkeistöt taloteknisen mallin vaikeudessa. Diplomityö. Aalto-yliopisto. Aaltodoc-tietokanta.

Vittori, Davide & Gualandi, Stefano. 2023. Integrating BIM and Brick schemas for building information interoperability. Teoksessa *Proceedings of the European Conference on Computing in Construction (EC3 2023)*, s. 137–144.

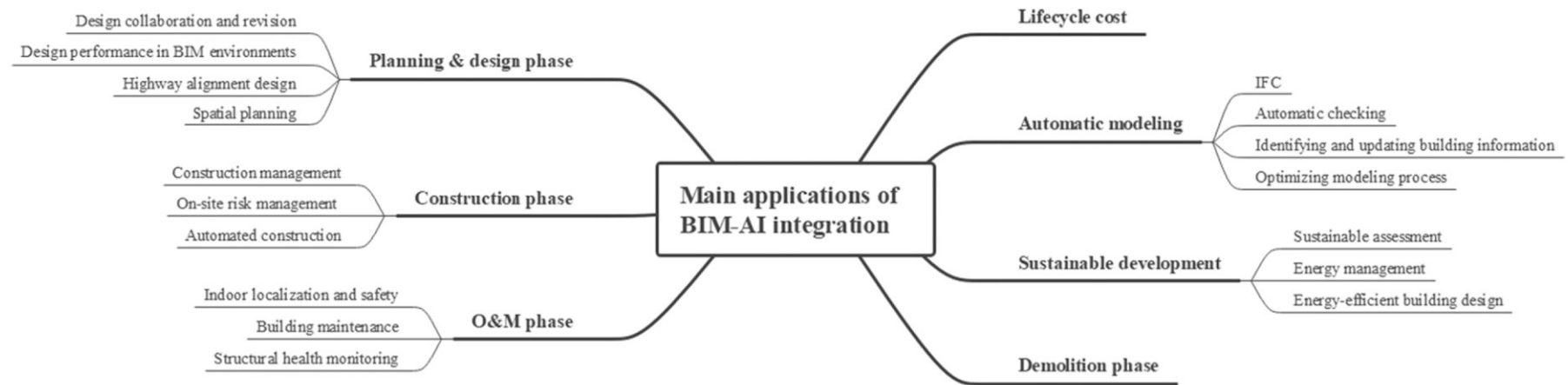
Yang, Xiaojun; Zhu, Xudong; Hou, Weizhuo & Chen, Shiyi. 2022. BEKG: A built environment knowledge graph. arXiv preprint, arXiv:2211.02864.

YTV2020 — Talotekniikan ja rakennesuunnittelun nimistöjen koekäyttö alkaa. Verkkoaineisto. 2021. KIRAHub. <<https://kirahub.org/ytv2020-talotekniikan-ja-rakennesuunnittelun-nimistojen-koekaytto-alkaa>>. Luettu 03.06.2025.

Zhang, Fan; Chan, Albert P. C.; Darko, Amos; Chen, Zhengxian & Li, Dezhi. 2023. Integrated applications of BIM and AI techniques in construction projects. *Journal of Building Engineering*, 68, 105456.

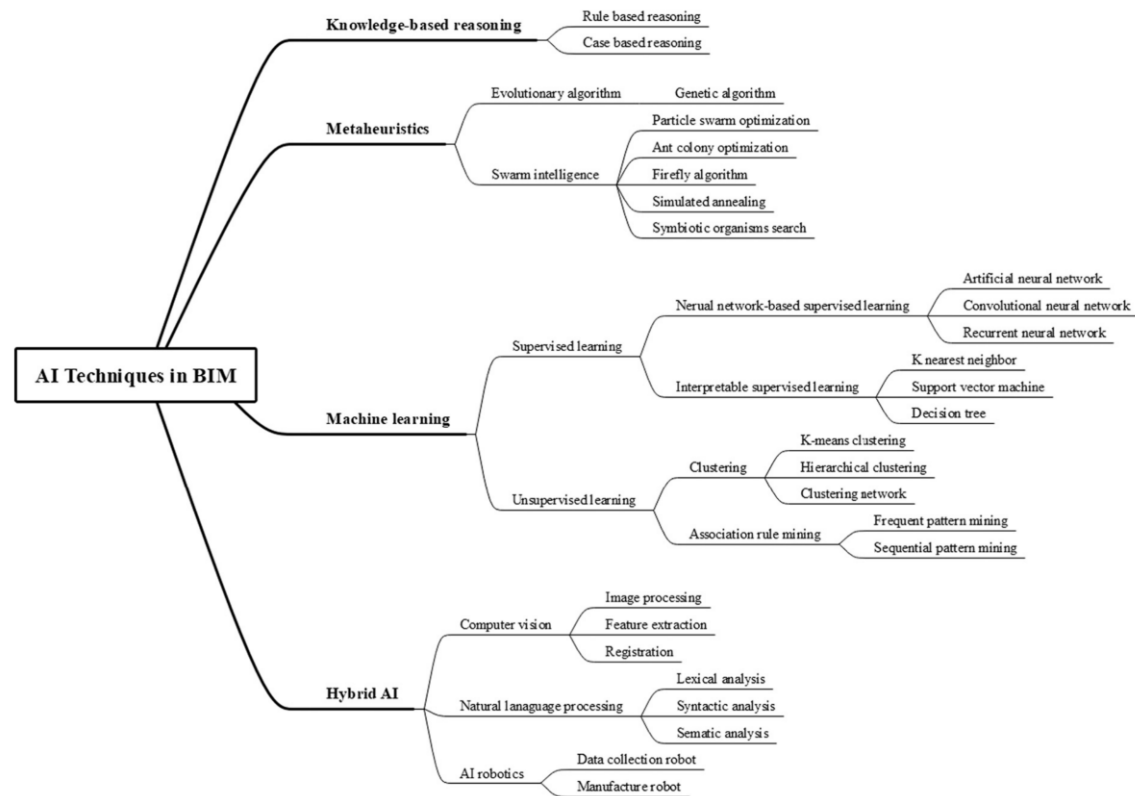
Zhu, Hongyi; Wang, Ning; Hou, Zhenhuan & Tang, Meilin. 2021. Releasing the power of graph for building information discovery. *Automation in Construction*.

## Liite 1. BIM–AI-integraation pääsovellukset ja tekoälyteknologiat



Kuva L1. Pääsovellukset BIM–AI-integraatioissa (Zhang ym. 2022, s. 7).

## Liite 1. BIM–AI-integraation pääsovellukset ja tekoälyteknologiat



Kuva L2. AI-teknologioita BIM-mallien yhteydessä (Zhang ym. 2022, s. 10).

## Liite2. Menetelmässä käytetyt keskeiset käsitteet ja viitekehukset

Taulukko 1. Menetelmässä käytetyt pääkäsitteet ja niiden merkitys.

Käsite / lyhenne	Selitys	Merkitys tässä menetelmässä
SysML v2	Systems Modeling Language, järjestelmien rakenteiden ja toiminnallisuuden mallintamiskieli.	IV-koneen rakenteiden, porttien ja attribuuttien kuvaamisen lähtökohta.
RDF	Resource Description Framework, semanttisen webin perusmalli.	Perusta linkitetyn datan muodostamiseen.
OWL	Web Ontology Language, RDF:n laajennus luokitteluun ja päättelyyn.	Mahdollistaa semanttisen päättelyn ja luokittelun graafeissa.
Brick	Ontologia, joka kuvaa rakennusautomaation laitteet ja anturit.	Tarjoaa kansainväliset käsitteet datapisteille.
BOT	Building Topology Ontology.	Kuvaa rakennuksen hierarkian: kerrokset, huoneet, vyöhykkeet.
SAREF	Smart Applications REFerence ontology.	Mahdollistaa IoT-laitteiden mallintamisen.
SKOS	Simple Knowledge Organization System.	Mahdollistaa kansallisten sanastojen (RAVA, RAU/LVI) julkaisemisen semanttisessa muodossa.
SSSOM	Standardi sanastojen ja ontologioiden välisiin taulukkomuotoisiin kuvauksiin.	Mahdollistaa käsitevastavuuksien tallentamisen ja jakamisen.
SHACL	Shapes Constraint Language.	Käytetään RDF-graafien validointiin.
SPARQL	Kyselykieli RDF:lle.	Käytetään tietojen hakemiseen ja analysointiin semanttisessa graafissa.

**Liite3. RDF- ja property-graafien vertailu.**

Ominaisuus	RDF-graafit	Property-graafit
Standardointi ja semantiikka	Perustuvat W3C-standardeihin (RDF, RDFS, OWL) ja tukevat formaalia päättelyä sekä yhteentoimivuutta kansainvälisissä tietojärjestelmissä (Pérez ym. 2009).	Vähemmän standardoituja, tietomalli riippuu käytetystä tietokannasta (esim. Neo4j, JanusGraph). Semantiikka on joustava, mutta ei yhtä formaali (Robinson ym. 2015: 75–85)
Tietorakenne	Tieto esitetään triploina (subjekti–predikaatti–objekti), mikä mahdollistaa tiedon linkittämisen avoimiin ontologioihin ja linkitettyyn dataan (Pauwels & Terkaj 2016: 100–133).	Solmuilla ja kaarilla voi olla suoria ominaisuuksia (properties), mikä tekee mallista intuitiivisen ja tehokkaan käytännön sovelluksissa (Angles & Gutierrez 2008: 1–39).
Ontologioiden hyödyntäminen	Ontologiat voidaan liittää suoraan tiedon semanttiseen määrittelyyn (esim. Brick, BOT), mikä tukee tiedon laajaa yhdistämistä ja analytiikkaa (Rasmussen ym. 2018: 95–101)	Ontologioiden liittäminen ei ole luontaista, vaan vaatii erillisiä integrointikerroksia. Sopii kuitenkin hyvin, kun tarvitaan sovelluskohtaisia ratkaisuja.
Kyselykielet	SPARQL mahdollistaa monimutkaisia semanttisia kyselyitä ja päättelyä (Pérez ym. 2009).	Käytetään graafikyselykieliä kuten Cypher tai Gremlin, jotka ovat tehokkaita iteratiiviseen analyysiin, mutta ilman semanttista päättelyä (Robinson ym. 2015: 75–85).
Käyttötapaukset rakennetussa ympäristössä	Sopii erityisesti tilanteisiin, joissa tarvitaan tietojen semanttista integraatiota (esim. IFC→RDF, Brick, BOT) ja avoimia rajapintoja (Pauwels & Roxin 2020).	Soveltuu dynaamiseen analyysiin, reaaliaikaiseen käyttöön ja sovelluskehitykseen. Esim. energiadata Neo4j:ssä. (Donkers ym. 2020).

**Liite 4. Menetelmäprosessin vaiheet, syötteet, työkalut, tuotokset ja hyödyt**

Vaihe	Tehtävä (selitys)	Syöte / Aineisto	Työkalut / standardit	Tuotos	Hyöty
1 SysML v2-malli IV-koneelle	Kuvataan komponentit, portit, attribuutit ja ohjaussuhteet järjestelmätasolla.	IV-koneen suunnittelu-tieto	SysML v2 v2	Järjestelmätason malli	Formaali rakenne, jota IFC ei kata laitetasolla.
2 Inventaario (ahu_inventory.csv)	Viedään mallin entiteetit taulukoksi jatkokäsittelyä varten ja yhdistetään (RAVA/RAU/LVI) termeihin.	SysML v2-malli	CSV/XMLI export	ahu_inventory.csv	Työkaluriippumaton, läpinäkyvä lähtö jatkovaiheisiin.
3 Automaattiset luokkaehdokkaat	Ehdotetaan ontologialuokkia (Brick/BOT/SAREF) ja termejä .	Inventaario	AML/DeepOnto tms., heuristiikat	candidates.csv	Vähentää käsityötä, tuo yhdenmukaisuutta.
4 Asiantuntijatarkistus ja käsitevastaavuuskirjasto	Vahvistetaan/korjataan ehdotukset, kirjataan perustelut.	Ehdokkaat	SSSOM	ahu_mappings.sssom.tsv	Jäljitettävä, versioitava käsitevastaavuuskirjasto.
5 Muunnos RDF/TTL	Tehdään linkitetty data käsitevastaavuuksista.	SSSOM-taulukko	sssom-py, RDF/OWL	ahu_mappings.ttl	Yhteentoimiva graafi, päätely ja kyselyt mahdollisia.
6 Validointi ja testaus	Rakenteellinen ja semanttinen tarkastus, regressiotestit.	TTL-graafi	SHACL, SPARQL	Validointiraportit	Laatu, toistettavuus, virheiden nopea havaitseminen.



## Liite 6. SysML v2 v2 ja IBD-osakaavio (komponentit + portit 1/3)

Luonne ja tarkoitus:

Tämä liite havainnollistaa, miten ilmanvaihtokoneen (AHU) rakenne ja toiminnalliset komponentit voidaan kuvata SysML v2 v2 -mallina menetelmän mukaisesti. Liite osoittaa, miten komponentit, portit ja yhteydet muodostavat rakenteellisen pohjan myöhemmälle ontologiselle esitykselle ja RDF-graafille.

Kuvaus:

Koodissa määritellään AHU-kokoonpano ulkoisilla rajapinnoilla (OutdoorAir, SupplyAir, ReturnAir, ExhaustAir) ja instansoidaan kuvan mukaiset osat (Damper, Filter, RotaryHeatExchanger, Coil, Fan). Yhdistykset (connect) kuvaavat ilmavirran kulun: ulkoilmasta vaimennus- ja suodatusasteiden kautta lämmön talteenottoyksikölle, siitä lämmitys- ja jäähdytyspatterien kautta puhaltimelle ja edelleen tuloilmaksi. Paluuilma johdetaan lämmön talteenottoyksikölle ja siitä poistoilmaksi. Vesipiirin portit on määritelty, mutta jätetty esimerkkimallissa käyttämättömiksi.

Portit on tyypitetty AirPort- ja WaterPort-liityntöinä, mikä määrittää komponenttien fyysisen rajapinnan ja tiedon kulkusuunnan. Kaikkia mallin osia ei voitu esittää graafisesti, sillä käytetyt avoimen lähdekoodin työkalut eivät vielä tue SysML v2 v2 -standardin visuaalisia ominaisuuksia täysimääräisesti. Internal Block Diagram (IBD) -esitystapa näyttää lohkon sisäiset rakenteet ja virtaussuunnat ja vastaa rakenteeltaan perinteistä säätökaaviota, mutta sen etuna on koneluettavuus ja suora yhteys mallin semanttisiin määrittämiin.

**Liite 6. SysML v2 v2 ja IBD-osakaavio (komponentit + portit 2/3)**

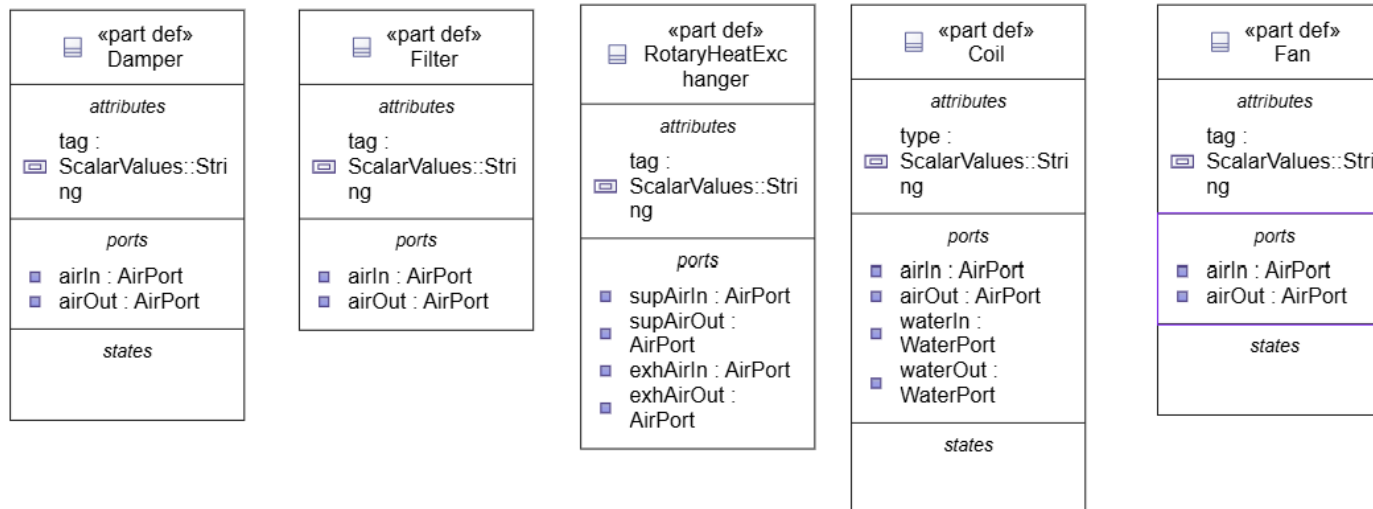
## Koodi L5.1. Tiivistetty SysML v2 v2 -kokoontulo (ote)

```
ppackage 'VentilationUnitModel' {  
  
    part def Damper {  
        attribute tag : String;  
        port airIn, airOut : AirPort;  
    }  
  
    part def Filter {  
        attribute tag : String;  
        port airIn, airOut : AirPort;  
    }  
  
    part def RotaryHeatExchanger {  
        attribute tag : String;  
        port supAirIn, supAirOut, exhAirIn, exhAirOut : AirPort;  
    }  
  
    part def Coil {  
        attribute type : String; // heating | cooling  
        port airIn, airOut : AirPort;  
        port waterIn, waterOut : WaterPort;  
    }  
  
    part def Fan {  
        attribute tag : String;  
        port airIn, airOut : AirPort;  
    }  
  
    part def VentilationUnit_Physical {  
        inout port OutdoorAir, SupplyAir : AirPort;  
        part damper : Damper;  
        part filter : Filter;  
        part heatExchanger : RotaryHeatExchanger;  
        part coil : Coil { attribute type = "heating"; }  
        part fan : Fan;  
  
        connect OutdoorAir -> damper.airIn;  
        connect damper.airOut -> filter.airIn;  
        connect filter.airOut -> heatExchanger.supAirIn;  
        connect heatExchanger.supAirOut -> coil.airIn;  
        connect coil.airOut -> fan.airIn;  
        connect fan.airOut -> SupplyAir;  
    }  
}
```

## Liite 6. SysML v2 v2 ja IBD-osakaavio (komponentit + portit 3/3)

Kuvassa esitetään ilmanvaihtokoneen pääosien määrittelyt osana järjestelmän rakenteellista mallia.

Kaavio toimii periaatteellisena esityksenä SysML v2 v2 -mallin rakenteesta ja havainnollistaa, miten komponenttien väliset portit ja ominaisuudet tukevat järjestelmän sisäisen toiminnallisuuden mallinnusta.



## **Liite 7. Inventaario (ahu\_inventory.csv – ote 1/2)**

Luonne ja tarkoitus:

Liite esittää esimerkin menetelmässä kuvattavasta inventaariosta, joka voidaan muodostaa SysML v2-mallin perusteella jatkokäsittelyä varten. Se havainnollistaa, miten komponentit ja niiden suhteet voidaan jäsentää CSV-muodossa jatkoanalyysien lähtötiedoksi.

Kuvaus:

Inventaario toimii menetelmän rakenteellisena välivaiheena, jonka avulla mallin osat voidaan myöhemmin yhdistää käsitevastaavuuksiin ja RDF-graafeihin. Esi-tetty taulukko on havainnollistava ote, joka kuvaa menetelmän mukaista tulosmuotoa.

**Liite 7. Inventaario (ahu\_inventory.csv – ote 2/2)**

Taulukko L6.1. AHU-järjestelmän inventaario (ote)

<b>id</b>	<b>name_fi</b>	<b>name_en</b>	<b>type</b>	<b>parent</b>	<b>description</b>
AHU_01	Ilmanvaihtokone	Air Handling Unit	System	–	Konekokonaisuus
FAN_SUP_01	Tuloilmapuhallin	Supply Fan	Component	AHU_01	Sähkökäyttöinen puhallin
COIL_HEAT_01	Lämmityspatteri	Heating Coil	Component	AHU_01	Kuumavesipatteri
FILTER_01	Suodatin	Filter	Component	AHU_01	F7-luokan suodatin

**Liite 8. Ehdokkaat (candidates.csv – ote)**

Luonne ja tarkoitus:

Liite havainnollistaa, miten menetelmässä voidaan tunnistaa käsite-ehdokkaita vertaamalla SysML v2-mallin käsitteitä eri ontologioihin (Brick, BOT, SAREF). Kuvaus:

Kuvaus:

Ehdokkaiden vertailu perustuu menetelmän periaatteeseen, jossa automaattinen tunnistus voidaan toteuttaa erilaisilla sanastollisilla ja semanttisilla vertailumenetelmillä. Taulukko toimii esimerkkinä tällaisen vaiheen tuotoksesta.

Taulukko L7.1. Automaattisesti tunnistetut käsite-ehdokkaat (ote)

<b>SysML v2_concept</b>	<b>candidate_uri</b>	<b>candidate_label</b>	<b>similarity</b>	<b>ontology</b>
Fan	brick:Fan	Fan	0.93	Brick
Coil	saref:HeatingCoil	Heating Coil	0.89	SAREF
Filter	brick:Filter	Filter	0.92	Brick

## Liite 9. Käsitevastaavuuskirjasto (ahu\_mappings.sssom.tsv – ote)

Luonne ja tarkoitus:

Liite esittää havainnollistavan esimerkin siitä, miltä SysML v2-ontologia -käsitevastaavuudet voivat näyttää, kun ne tallennetaan SSSOM-standardin mukaisesti.

Kuvaus:

Käsitevastaavuuskirjasto toimii menetelmän ytimessä, jossa eri ontologioista saadut käsitteet harmonisoidaan yhtenäiseksi tietorakenteeksi. Taulukko osoittaa käytännön esimerkin harmonisoinnin lopputuloksesta.

Taulukko L8.1. Käsitevastaavuuskirjasto (ote)

<b>subject_id</b>	<b>subject_label</b>	<b>predicate_id</b>	<b>object_id</b>	<b>object_label</b>	<b>mapping_justification</b>
SysML v2:Fan	Fan	skos:exact-Match	brick:Fan	Fan	Manual alignment
SysML v2:HeatingCoil	Heating Coil	skos:exact-Match	saref:HeatingCoil	Heating Coil	DeepOnto alignment
SysML v2:Filter	Filter	skos:close-Match	brick:Filter	Filter	AML candidate

Taulukko osoittaa, miten eri lähteistä saadut vastaavuudet on yhtenäistetty standardoiduksi kirjastoksi.

## Liite 10. AML ja DeepOnto – verifiointitulokset

Luonne ja tarkoitus:

Liite havainnollistaa menetelmän testausvaiheen suunniteltua vertailua, jossa arvioidaan automaattisen käsiteyhdistämisen tarkkuutta ja kattavuutta (Precision, Recall).

Kuvaus:

Tulokset osoittavat, millaisia eroja voidaan odottaa eri menetelmien välillä. Esi-tetyt luvut toimivat mallinnuksellisinä tulosesimerkkeinä, jotka perustuvat kirjalli-suudessa raportoituun suorituskyykytasoon vastaavista algoritmeista.

Taulukko L9.1. AML ja DeepOnto -menetelmien vertailu

Algorithm	Candidate count	Accepted matches	Precision	Recall
AML	128	102	0.83	0.79
DeepOnto	142	119	0.84	0.85

## Liite 11. RDF/TTL-graafi (ahu\_graph.ttl – ote)

Luonne ja tarkoitus:

Liite kuvaa, miten SysML v2-mallin rakenne voidaan esittää ontologisessa muodossa RDF/TTL-graafina.

Kuvaus:

Graafi havainnollistaa menetelmän lopputulosta, jossa AHU:n komponentit ja suhteet on kuvattu Brick-ontologian mukaisesti. Tämä osoittaa, että menetelmä mahdollistaa tietojen koneluettavan ja laajennettavan esityksen.

### Koodi L10.1. AHU-järjestelmän RDF/TTL-graafi (ote)

```
@prefix ex: <https://example.org/ahu/> .
@prefix brick: <https://brickschema.org/schema/Brick#> .

ex:AHU_01 a brick:Air_Handler ;
    brick:hasPart ex:AHU_01_FAN_SUP ,
                  ex:AHU_01_FILTER_01 ,
                  ex:AHU_01_HEAT_COIL .

ex:AHU_01_FAN_SUP a brick:Fan ;
    brick:hasPoint ex:AHU_01_FAN_SPEED_SENSOR .

ex:AHU_01_FAN_SPEED_SENSOR a brick:Speed_Sensor .
```

Koodilohkossa nähdään, miten SysML v2-komponentit on muunnettu Brick-luokkiin ja liitetty AHU-kokoonpanoon.

## Liite 12. SHACL-säännöt ja SPARQL-kyselyt (+ tulokset)

Luonne ja tarkoitus:

Liite esittää esimerkin siitä, miten menetelmässä voidaan määritellä rakennetta ja semantiikkaa validoivia sääntöjä sekä kyselyitä.

Kuvaus:

SHACL-säännöt ja SPARQL-kyselyt havainnollistavat validoinnin ja tietohaun periaatteita, joita voidaan soveltaa menetelmän tuloksiin. Ne osoittavat, että menetelmä tukee tietojen automaattista tarkistettavuutta.

Koodi L11.1. SHACL-validointisääntö (ote)

```
sh:FanShape a sh:NodeShape ;
  sh:targetClass brick:Fan ;
  sh:property [
    sh:path brick:hasPoint ;
    sh:class brick:Sensor ;
    sh:minCount 1
  ] .
```

Koodi L11.2. SPARQL-kysely (ote)

```
SELECT ?fan ?sensor
WHERE {
  ?fan a brick:Fan .
  ?fan brick:hasPoint ?sensor .
}
```

SHACL-sääntö varmistaa, että jokaisella Fan-instanssilla on vähintään yksi Sensor, ja SPARQL-kysely palauttaa nämä parit graafista.

## Liite 13. YAML-konfiguraatio ja rajapinta-asetukset (GraphDB)

Luonne ja tarkoitus:

Liite havainnollistaa, miten RDF/TTL-muunnosohjelman asetukset voidaan määrittellä YAML-muodossa ja liittää tietokantarajapintaan.

Kuvaus:

Konfiguraatio on esimerkinomainen, ja sen avulla osoitetaan, miten menetelmä voidaan liittää GraphDB:n tai vastaavan RDF4J-yhteensopivan palvelun kanssa. Tämä tukee menetelmän yhteensopivuutta pilviympäristöihin.

Koodi L12.1. YAML-konfiguraatio RDF-muunnosohjelmalle (GraphDB, ote)

```
converter:
  input_file: ahu_inventory.csv
  mapping_file: ahu_mappings.sssom.tsv
  output_file: ahu_graph.ttl
  namespaces:
    ex: https://example.org/ahu/
    brick: https://brickschema.org/schema/Brick#

graphdb:
  # GraphDB (RDF4J) Workbench base URL
  base_url: http://localhost:7200
  # Repository ID (luo GraphDB:ssä, esim. ahu_repo)
  repository: ahu_repo
  endpoints:
    sparql_query: http://localhost:7200/repositories/ahu_repo
    sparql_update: http://localhost:7200/repositories/ahu_repo/statements
  import:
    # TTL-tiedoston lataus suoraan statements-rajapintaan
    statements_endpoint: http://localhost:7200/repositories/ahu_repo/statements
    content_type: text/turtle
```

Konfiguraatio ohjaa ohjelman tuottamaan TTL-tiedoston ja lataamaan sen GraphDB-repositorioonsa RDF4J-yhteensopivalla rajapinnalla.

## Liite 14. Suorituskomennot ja käyttöohje (GraphDB, toistettavuus)

Luonne ja tarkoitus:

Liite kuvaa periaatteellisesti, miten menetelmän vaiheet voidaan suorittaa ohjelmallisesti RDF- ja SPARQL-rajapintojen kautta.

Kuvaus:

Komennot on esitetty havainnollistavana esimerkkinä paikallisen kehitysympäristön suorituksesta. Ne osoittavat, että menetelmä on suunniteltu toistettavaksi ja laajennettavaksi tulevilla toteutuksissa.

### Koodi L13.1. Ohjelman suorituskomennot (GraphDB)

```
# 1) Aktivoi ympäristö ja generoi TTL
conda activate SysML v22rdf
python SysML v2_sssom_to_ttl.py \
  --inventory ahu_inventory.csv \
  --mapping ahu_mappings.sssom.tsv \
  --output ahu_graph.ttl

# 2) Lataa TTL GraphDB:hen (RDF4J statements -rajapinta)
#   Huom: -d lähettää TTL:n suoraan; GraphDB tunnistaa Content-Type:
text/turtle
curl -X POST \
  -H "Content-Type: text/turtle" \
  --data-binary @ahu_graph.ttl \
  "http://localhost:7200/repositories/ahu_repo/statements"

# 3) Aja esimerkkikysely (SPARQL SELECT) GraphDB:n query-endpointiin
curl -G \
  -H "Accept: application/sparql-results+json" \
  --data-urlencode 'query=SELECT ?fan ?sensor WHERE { ?fan a
<https://brickschema.org/schema/Brick#Fan> . ?fan
<https://brickschema.org/schema/Brick#hasPoint> ?sensor . } LIMIT 50' \
  "http://localhost:7200/repositories/ahu_repo"

# 5) Avaa GraphDB Workbench selaimessa (UI-kyselyt ja visualisointi)
#   - SPARQL-kyselyt: Workbench -> SPARQL
#   - Visualisointi: Workbench -> Explore -> Visual graphs
# (Avaa selaimella) http://localhost:7200
```

## **Liite 15. Tekoälyn käyttö opinnäytetyössä**

Tässä opinnäytetyössä on hyödynnetty tekoälytyökaluja työn suunnittelun, jäsentelyn, kieliasun ja sisällön tuottamisen tukena. Tekoälyä on käytetty erityisesti kirjallisuuskatsauksen kokoamisessa, käsitteiden ja terminologian yhtenäistämässä sekä lähteiden hakemisen ja rajaamisen tukena.

Tekoälyä on hyödynnetty myös opinnäytetyössä kehitetyssä menetelmässä, jossa sen käyttö on osa prosessia erityisesti kielellisessä yhteensovituksessa (lexical alignment). Tekoälyä on käytetty sanakirjatyypillisesti suomi–englanti-kielisten käsitevastaavuuksien muodostamisessa ja tarkistamisessa sekä terminologisen yhdenmukaisuuden arvioinnissa.

Lisäksi tekoäly on tukenut ohjelmointivaiheessa koodirakenteiden ja validointilogiikan hahmottamista sekä työn kirjoitusvaiheessa tekstin selkeyttämistä. Kaikki tekoälyn avulla tuotetut tai muokatut sisällöt, ehdotukset ja käännökset on hyväksytty opinnäytetyön tekijän toimesta.

Tekoäly ei ole toiminut työn tekijänä eikä analyysin suorittajana, vaan sen rooli on ollut avustava ja oppimista tukeva. Opinnäytetyön tekijänä olen vastuussa kaikesta opinnäytteeni sisällöstä.