# Raportointityökalun kehittäminen Sovelia® PLM-järjestelmään

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Tietotekniikka
Marraskuu 2023
Elias Peltonen

TURUN YLIOPISTO
Tietotekniikan laitos

Elias Peltonen: Raportointityökalun kehittäminen Sovelia® PLM-järjestelmään

TkK-tutkielma, 24 s., 4 liites.
Tietotekniikka
Marraskuu 2023

---

Tiivistelmä tähän. *Tarkoituksena kirjoittaa tämä lopuksi!*

Asiasanat: tähän, lista, avainsanoista

UNIVERSITY OF TURKU
Department of Computing

Elias Peltonen: Raportointityökalun kehittäminen Sovelia® PLM-järjestelmään

Bachelor's Thesis, 24 p., 4 app. p.
Information Technology
November 2023

---

Second abstract in english (in case the document main language is not english)

Keywords: here, a, list, of, keywords

# Sisällys

# Taulukot

# Termistö

**API** ohjelmointirajapinta, engl. Application Programming Interface

**BI** engl. Business Intelligence, liiketoimintatiedon hyödyntäminen

**Big data** suom. massadata, erittäin suuret ja järjestämättömät jatkuvasti lisääntyvät tietomassat

**BOM** Bill of Materials, osaluettelo, tuoterakenne

**CAD** engl. Computer Aided Design, tietokoneen käyttö suunnittelun apuvälineenä

**CSS** engl. *Cascacing Style Sheets*, suom. *porrastetut tyyliarkit*, tyylisivu, jolla voidaan määritellä tyyliohjeita miten esimerkiksi HTML dokumentti voidaan esittää

**HTML** engl. *HyperText Markup Language*, suom. *hypertekstin merkintäkieli*, merkintäkieli, jolla internetsivut ovat kirjoitettu

**JSON** engl. JavaScript Object Notation, yksinkertainen tiedostomuoto tiedon välitykseen ja tallennukseen

**PLM** engl. Product Lifecycle Management, tuotteen elinkaaren hallinta

**SQL** engl. *Structured Query Language*, standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä

**XML** merkintäkielien standardi ja tiedostomuoto, engl. Extensible Markup Language

**XSL** XML-kieliperhe, joka mahdollistaa XML-pohjaisten tiedostojen ulkoasun ja rakennemuutoksen määrittelyn

**ZIP** tiedon pakkaukseen käytetty tiedostomuoto, joka voi sisältää useita tiedostoja ja kansioita pakattuna

# 1 Johdanto

Tietotekniikan avulla voidaan tehostaa ja helpottaa työnteon tuottavuutta, kun samaan tehtävään käytetty aika vähenee. Tietotekniikan hyödyntäminen raportointiin hyvin luonnollista, sillä raportit ovat useimmiten digitaalisesti tuotettuja dokumentteja, joiden kokoaminen vaatii jonkinlaista laskentaa. Raportointidatan kerääminen ja jäsenteleminen manuaalisesti on hyvin vaivalloista ja hidasta, mikä vuoksi tietojärjestelmät voivat tarjota raportointityökaluja, joiden tarkoituksena on koota raportti automoidusti määritellystä lähdedatasta.

Raportoinnin ydinajatuksena on tuottaa tietoa muodossa, joka on helposti ymmärrettävissä ja jaettavissa. Raportointityökalujen avulla olemassa olevasta suuresta määrästä dataa voidaan tuottaa selkeä ja jäsennelty esitys, joka kokoaa lähdedatan tärkeimmät seikat helposti yhdellä silmäyksellä omaksuttavaan muotoon.

Tämän työn tarkoituksena oli toteuttaa raportointityökalu osaksi Sovelia PLM -järjestelmää. Sovelia PLM on kaupallinen tuotteen elinkaaren hallintajärjestelmä(PLM), (engl. *Product Lifecycle Management*). PLM-järjestelmän pääasiallisena tarkoituksena on koota tietoa yrityksen tuotteiden koko elinkaaren vaiheista keskitettyyn tietojärjestelmään. Tämä keskitetty tietojärjestelmää on käytettävissä yrityksen eri työryhmien ja liiketoimintajärjestelmien välillä, minkä tarkoituksena on vähentää virheellisten tuotetietojen aiheuttamia turhia kustannuksia sekä viivästyksiä ja siten nopeuttaa yrityksen prosessia saada kehitetty tuote markkinoille.

Raportointityökalu voidaan nähdä yhtenä PLM järjestelmälle lisäarvoa tuottava-

na ominaisuutena. Luotettavan, tehokkaan ja mukautuvan raportointityökalun avulla PLM-järjestelmä voi tuottaa enemmän lisäarvoa sen käyttäjille tarjoamalla mahdollisuuden jakaa, tallentaa ja analysoida tuotedataa eri tiedostoformaateissa sekä yrityksen sisäisten työryhmien että ulkoisten toimijoiden välillä. PLM-järjestelmiä käyttävillä yrityksillä on tyypillisesti suuria määriä tuotetietoja ja syviä tuoterakenteita, jolloin myös raportoinnin suorituskykyvaatimukset korostuvat.

PLM-järjestelmien tietomallit voidaan jakaa dokumentti- ja relaatiodatapohjaisiin tietorakenteisiin. [1] Koska Sovelia PLM -järjestelmä perustuu relaatiodatapohjaiseen tietomalliin, myös tässä tutkielmassa käsitellään raportointia nimenomaan relaatiodatan pohjalta.

Raportointityökalu integroituu osaksi Sovelian nykyistä lähdekoodia ja sen palvelinkomponentteja. Ohjelmakokonaisuus koostu palvelinkomponentista, joka tuottaa raporttitiedoston raportoinnin kohteena olevasta objektista, sekä konfigurointityökalusta, jonka avulla pääkäyttäjä voi muokata raporttien ulkonäköä ja rakennetta.

### Tutkimuskysymykset

- Mitä tulee ottaa huomioon raportointityökalua kehittäessä osaksi PLM-järjestelmää?
- Millaisia ovat nykyiset raportointityökalut?
- Miksi raportointityökaluja ja PLM-järjestelmiä kehitetään ja miten ne toimivat yhdessä?

### Tutkimusmenetelmät

- Kirjallisuusanalyysi
- Kvalitatiivinen case-analyysi

1

[1]Johdannon sisältöä tulee arvioida uudelleen, kun tutkielman sisältö on valmiimpi. Tutkielman rakenne tulee esittää tiivistetysti ja nykyistä tekstiä tulee karsia.

# 2 Raportointi ja PLM järjestelmät

## 2.1 PLM-strategia ja PLM-järjestelmät lyhyesti

Laajempana käsitteenä tuotteen elinkaaren hallinta eli PLM voidaan nähdä yrityksen strategiana hallita tuotetietoja. PLM strategiana koostuu tuotteista, organisaatioista, työmenetelmistä, prosesseista, ihmisistä ja lopulta usein myös tietoteknisestä elinkaaren hallintajärjestelmästä. Tietoteknisen elinkaaren hallintajärjestelmän, eli PLM-järjestelmän, tarkoituksena on mahdollistaa ja helpottaa PLM-strategian käyttöönottoa ja hyödyntämistä käytännössä. PLM-järjestelmä sisältää siis erilaisia toiminnallisuuksia, joiden avulla sen käyttäjät voivat hallita tuotetietoja keskitetyssä järjestelmässä tuotteen koko elinkaaren ajan.

Tuotteen elinkaari voidaan jakaa alku-, keski- ja loppuvaiheeseen. Tuotteen elinkaaren pääpiirteet on hyvä ymmärtää, jotta PLM-käsitettä voidaan tarkastella syvällisemmin. Bouhaddoun (2012) konferenssiartikkeli *"PLM Model for Supply Chain Optimization"* määrittelee tuotteen elinkaaren vaiheet ja niiden piirteet seuraavasti: [2]

- Alkuvaiheessa tuotteen vaatimuksia määritellään ja tuote on luonnosvaiheessa. Luonnosvaiheessa tuotetta voidaan kutsua prototyypiksi (engl. *prototype*) tai mallinnukseksi (engl. *mockup*).

- Keskivaiheessa tuote siirtyy tuotantoon ja valmistukseen. Tässä vaiheessa toteutetaan laadunvalvontaa ja kasaamista, ja voidaan puhua jo varsinaisesta

tuotteesta. Valmis tuote siirtyy jakeluverkoston kautta itse asiakkaalle. Kun tuote on asiakkaalla, korostuu tuotteen käyttö sekä mahdollinen huolto ja asiakastuki.

- Loppuvaiheessa tuotteen elinkaari päättyy. Tuotteen valmistusta ei koeta enää tarpeelliseksi, joten tässä vaiheessa huomio keskittyy tuotannon lopettamiseen ja tuotteen kierrätykseen.

Alemanni, ym. (2008) esittää artikkelissaan *"Key performance indicators for PLM benefits evaluation: The Alcatel Alenia Space case study"*, että PLM-strategian keskittyessä olennaisesti tuotedatan hallintaan, on PLM-ohjelmisto eli PLM-järjestelmä olennainen osa strategian hyödyntämistä käytännössä. [3] Alemanni korostaa, että PLM-järjestelmän kehittäjän tulee kuitenkin toimia yhteistyössä asiakkaiden kanssa, jotta tuotteiden elinkaaren eri vaiheet ja prosessit voidaan sisällyttää osaksi ohjelmiston toimintoja asiakaskohtaisesti. PLM-käsitteeseen liittyvien määritelmien lisäksi on tärkeää ymmärtää PLM-strategian ja -ohjelmistojen hyötyjä, jotta niiden hyödyntämisen motiivit voidaan ymmärtää. Tarkoituksena on siis vastata siihen, miksi ylipäätään PLM-järjestelmiä käytetään ja kehitetään.

## 2.2 PLM-strategian hyödyt ja merkitys

PLM-strategian hyötyjä on käsitelty laajasti [3] [4]. Strategian hyödyt voidaan jakaa kahteen osa-alueeseen: lyhyen ja pitkän aikavälin hyötyihin. Tietotekniset PLM-järjestelmät mahdollistavat PLM-strategian käyttöön ottamisen ja hyödyntäminen käytännössä koko yrityksen tasolla, mikä mahdollistaa laajojen tuotekantojen johdonmukaisen ja keskitetyn hallinnan yhteistyössä yrityksen eri osastojen ja kumppaneiden välillä. Konkreettisia PLM-strategian hyötyjä voidaan havaita Lee, ym. (2008) toteuttamasta tutkimuksesta PLM-strategian ja -järjestelmien hyödyntämisestä ilmailualalla: IBM-Dassaultin PLM-järjestelmää käytettiin lentokoneiden elin-

kaaren hallinnassa, jolloin ohjelmiston hyödyntäminen vähensi valmistusaikaa 16:sta kuukaudesta seitsemään kuukauteen. Lisäksi Teamcenter PLM -järjestelmä laski tuotantosykleihin käytettyä aikaa 35:llä prosentilla ja valmistusaikaa 66:lla prosentilla. Keskitetyssä järjestelmässä myös lentokoneiden huoltotarve voitiin ottaa paremmin huomioon jo suunnitteluvaiheessa, mikä suoraviivaisti myös tuotteiden huoltoa niiden elinkaaren aikana. [5]

### Lyhyen aikavälin hyödyt

Lyhyellä aikavälillä PLM-strategian ja PLM-järjestelmän käyttöönotto voi vähentää aikaa jota käytetään työntekijöiden jokapäiväisten työtehtävien suorittamiseen. Strategian avulla yrityksen tuotetiedot ovat keskitetysti saatavilla, eikä ajantasaisia tietoa tarvitse kysellä eri osastojen välillä. Tämä johtaa siihen, että työntekijät voivat käyttää enemmän aikaa tehtäviin, jotka tuottavat yritykselle lisäarvoa arvoa. Lisäksi tuotteiden rakenteiden ymmärtäminen ja visualisointi helpottuu PLM-järjestelmän käyttöönoton myötä. Tuoterakenteiden ymmärrystä ja jaettavuutta eri osastojen välillä voidaan parantaa entisestään myös PLM-järjestemän raportointiominaisuuksilla. [3]

### Pitkän aikavälin hyödyt

Pidemmällä aikavälillä hyödyt näkyvät konkreettisemmin PLM-strategiaa hyödyntävien yritysten tunnusluvuissa, erityisesti myyntikatteessa. PLM-järjestelmien keskeinen hyöty on prosessien suoraviivaistaminen, mikä johtaa usein tuotteiden saamiseen nopeammin ja useimmin markkinoille. Kun tuotteet pääsevät nopeammin suunnittelusta markkinoille, niiden suunnitteluun ja kehittämiseen käytetyt kustannukset luonnollisesti laskevat. [2] [3]

### 2.2.1 Osaluettelo PLM-järjestelmän sydämenä

Yksi PLM-järjestelmän tärkeimmistä toiminnallisuuksista on tuotteen osaluettelon (BOM)(engl. *Bill of Materials*) esittäminen organisoidusti. [1] Yksinkertaisuudessaan osaluettelo on lista kaikista osista, joita tarvitaan tuotteet valmistamiseen. Osaluettelossa jokainen yksittäiseen osaan voidaan liittää useita tietokenttiä kuten valmistaja, versio, materiaali ja määrä. Osaluettelo koostuu usein hierarkkisesti osakokoonpanoista, välikokoonpanoista, osakomponenteista ja yksittäisistä osista, eli se kerää dataa siitä, kuinka eri tuotteen komponentit ovat riippuvaisia toisistaan. Osaluetteloa voidaan käyttää viestintään esimerkiksi valmistuskumppanien välillä tai se voidaan rajoittaa yhteen tuotantoyksikköön. [6]

Koska osaluettelot ovat hyvin olennainen osa PLM-järjestelmää, ovat ne tärkeä kohde myös raportoinnille. [7] Osaluetteloista tuotetut raportit voivat analysoida rakennetta pintaa syvemmältä sekä luoda helposti ymmärrettävän yleiskatsauksen massiivisen osaluettelon omaavaan tuotteeseen tarjoamalla samalla esimerkiksi graafeja ja statistiikkaa tuotteesta. Koska raportit voidaan tuottaa erillisinä sähköisinä dokumentteina, voidaan laskentaa jatkaa esimerkiksi Excel-raporttien tapauksessa, tai erityisesti PDF-raportit ovat omiaan arkistoinnille myöhempää käyttöä varten.

### 2.2.2 Raportointi ja Business Intelligence

Liiketoimintatiedon hyödyntämisellä (BI), (engl. *Business Intelligence*) tarkoitetaan yrityksen kykyä hyödyntää dataa merkityksellisellä tavalla. PLM:n kontekstissa BI korostuu etenkin tuotetiedon hyödyntämisessä. Tätä PLM:n ja BI:n yhteyttä on tutkinut Bosch-Mauchand, ym. (2014) artikkelissaan *"Preliminary Requirements and Architecture Definition for Integration of PLM and Business Intelligence Systems"* [8]. Bosch-Mauchand totetaa, että PLM järjestelmä kulkee käsikädessä BI:n kanssa ja PLM-järjestelmän tuotedatan integraatio ja sen jaettavuus eri järjestelmien välillä on hyvin tärkeää tuotetiedon merkityksellisen hyödyntämisen kannalta. Bosch-

Mauchand erittelee, että jotkin PLM-järjestelmät tarjoavat erillisiä moduuleja raporttien tuottamiseen, mutta harva raportointityökalu tai -moduuli hyödyntää BI:n periaatteita. Bosch-Mauchandin mukaan varsinkin kahden tyyppisillä raporteilla voidaan tuottaa lisäarvoa: [8]

- Dokumenttien ja objektien määrällinen analysointi. Esimerkiksi näiden summien tai tyyppien laskenta.
- PLM-järjestelmien ominaisuuksien käyttö. Esimerkiksi tuotteen osien uudelleenkäyttö ja tietokantakyselyt.

Näiden lisäksi raportointia voidaan hyödyntää IT-hallinnon osa-alueilla, kuten esimerkiksi järjestelmän suorituskyvyn monitoroinnin kannalta.

## 2.3 Raportointimoottorit ja -työkalut

Raportointimoottorit ja -työkalut ovat molemmat tietojen käsittelyyn ja analysointiin tarkoitettuja ohjelmistoja. Tämän tutkielman kontekstissa nämä käsitteet määritellään seuraavanlaisesti: Raportointimoottorit ovat ohjelmistoja, jotka automatisoivat raporttien luomisen ohjelmallisesti. Tyypillisesti raportointimoottori saa syötedatan, jonkinlaisen ohjeistuksen tai raporttipohjan (engl. *report template*) ja kaikki logiikka näiden yhdistämiseen sekä lopullisen raportin muodostamiseen jää raportointimoottorin vastuulle. [9] Raportointityökalu on taas laajempi termi, jolla tarkoitetaan ohjelmistoa, joka on suunniteltu auttamaan raporttien luomisessa, hallinnassa ja jakelussa. Yksinkertaistettuna raportointityökalut ovat suunniteltu auttamaan käyttäjiä luomaan raportteja, kun taas raportointimoottorit ovat suunniteltu automatisoimaan itse raporttien luominen jonkin annetun lähdedatan perusteella.

Raportointimoottori on tyypillisesti taustalla toimiva, esimerkiksi palvelinsovellus, joka keskittyy raporttien prosessointiin ja renderöimiseen. Se huolehtii esimerkiksi tietojen hakemisesta, yhdistämisestä ja muotoilusta ennalta määritettyjen ra-

porttipohjien ja ohjeiden perusteella. Se ei sisällä itse käyttöliittymää ja sen toimintoja käytetään ohjelmallisesti. Raportointityökalu sisältää usein jonkinlaisen käyttöliittymän, jonka avulla käyttäjä voi luoda uusia tai muokata olemassa olevia raportteja. Ne voivat olla sisäänrakennettuja johonkin järjestelmään, esimerkiksi PLM-järjestelmään, mutta raportoinityökalut voivat olla myös ulkoisia ohjelmistoja, jotka pääsevät käsiksi analysoitavaan dataan esimerkiksi jonkin rajapinnan kautta. Koska raportointimoottori voi olla konfiguroitavissa raportointityökalun avulla myös loppukäyttäjien toimesta, tulee sitä kehittäessä huomioida loppukäyttäjien yksilölliset ja monipuoliset tarpeet. [10]

Bambang Prasetya Adhi ym. vertailevat tutkimusartikkelissaan *"Performance comparison of reporting engine birt, jasper report, and crystal report on the process business intelligence"* (2019) [10] suosituimpia kaupallisia (SAP Crystal Reports) ja avoimen lähdekoodin (BIRT ja Jasper Report) raportointityökaluja ja niiden alla toimivia raportointimoottoreita. Ahdi ym. käyttävät kokeellisia menetelmiä mitatakseen kolmea raportointityökalun osa-aluetta:

- Soveltuvuus, esimerkiksi kuinka hyvin raportointimoottori tukee erilaisen lähdedatan käyttöä
- Käytettävyys, joka ilmenee oppimisen helppoutena sekä toiminnallisuuden loogisuutena ja tehokkaana käyttönä
- Tehokkuus, joka mittaa itse järjestelmän tehokkuutta, esimerkiksi suoritusaikaa

Näitä osa-alueita arvioimalla voidaan tehdä perusteltuja päätöksiä raportointityökalun ja -moottorin valinnasta, joten nämä ovat tärkeitä seikkoja ottaa huomioon näiden suunnittelussa ja kehityksessä.

### 2.3.1 Raportointimoottorin rakenne ja prosessi

Prosessina raportointimoottori toimii kolmella tasolla: data-, logiikka- ja esitystasolla. [9]

#### Datataso

Datatasolla raportointimoottori voi hakea dataa suoraan tietotokannasta tai esimerkiksi API:n eli ohjelmointirajapinnan (engl. *Application Programming Interface*) kautta. Raportointimoottorin tapauksessa ohjelmointirajapinta voi olla esimerkiksi hakurajapinta, jonka taustalla toimivan hakumoottorin avulla raportointimoottori voi hakea tarvitsemaansa dataa täsmällisemmin. Varsinkin PLM-järjestelmän kontekstissa hakumoottori on hyvin keskeinen osa PLM-järjestelmää. Datataso määrittelee siis mistä ja miten raportointimoottori hankkii lähtödataa sekä millaiset lähtötiedot sillä on koostaa raportti. [9] Näihin lähtötietoihin lukeutuu esimerkiksi mahdolliset raporttipohjat ja muut käyttäjän määrittämät asetukset.

#### Logiikkataso

Logiikkatasolla raportointimoottori jäsentelee lähtödataa ja suorittaa laskentaa. Lähtödatan formaatti on usein historiallisesti ollut XML (engl. *Extensible Markup Language*) sen ollessa yksi internetin yleisimmin käytetyistä dataformaateista, mutta JavaScriptin yleistyttyä myös JSON (engl. *JavaScript Object Notation*) noussut suosituksi tiedostomuodoksi. JSONin etuna on sen suora integraatio JavaScriptin yhteyteen sekä sen nopeus verrattuna vanhempaan XML-standardiin. [11] Logiikkatason toteuttaman laskennan avulla lähtödatasta voidaan luoda esimerkiksi graafeja ja sekä taulukoida dataa, mikä mahdollistaa lasketut sarakkeet ja summat. Tämä voidaan nähdä raportointimoottorin ytimenä, sillä se tuottaa merkityksellistä dataa usein vaikeaselkoisesta lähdedatasta. [9] Lisäksi logiikkatason tulee jäsentää data siten, että se on mahdollista kirjoittaa tiedostoon esitystasolla.

#### Esitystaso

Esitystasolla data kirjoitetaan tiedostoon, jolloin se voidaan tallentaa käyttäjäystävällisessä tiedostoformaatissa. [9] Suosituimpia tiedostoformaatteja ovat HTML-, Excel- ja PDF-tiedostoformaatit. [9] Esitystasolla raportin merkitys ilmenee käyttäjälle: lähtödata on esitetty helposti omaksuttavassa ja ymmärrettävässä muodossa, sekä raportti on luettavissa ja jaettavissa helposti yksittäisenä tiedostona.

### 2.3.2 Raportointi PLM-järjestelmässä

Kuten osioissa 2.2 todettiin, yksi PLM-järjestelmän hyödyistä on yksittäisten työntekijöiden työmäärän vähentäminen ja prosessien suoraviivaistaminen. Tietoteknisten järjestelmien etuna on varsinkin automoitu laskenta, jota hyödyntämällä voidaan vähentää inhimillisiä virheitä. [12] [13] Automoitua laskentaa hyödynnetään erityisesti erilaisten raporttien muodostamisessa.

Koska PLM:n tarkoituksena mahdollistaa koko tuotantoketjun yhteistyö asiakkaiden, kehittäjien, toimittajien ja valmistajien välillä tuotteen eri elinkaaren vaiheissa, [2] on tärkeää että tuotetieto elinkaaren eri vaiheissa on dokumentoitavissa, analysoitavissa ja helposti jaettavissa. Koska PLM-järjestelmien tietomallit ovat usein ohjelmistokohtaisia ja harvemmin standardinomaisia [14] on tärkeää, että tuotedataa on mahdollista viedä myös itse järjestelmän ulkopuolelle tallennettavaksi ja jaettavaksi.

PLM:n kontekstissa raporteilla tarkoitetaan tuotedataa kokoavia ja analysoivia kokonaisuuksia. Raportit voivat olla esimerkiksi digitaalisia PDF- tai Excel-dokumentteja, jotka kokoavat tuotetietoja ja suorittavat laskentaa visualisoimalla dataa esimerkiksi kuvaajin. Toinen PLM-järjestelmille tyypillinen raporttimuoto on interaktiiviset "kojelautoja"(engl. *dashboards*), jotka kokoavat useita kuvaajia ja laskettuja arvoja yksittäiseen käyttäjäystävälliseen näkymään, tarjoamalla esimerkiksi jonkin Web-käyttöliittymän. Tässä tutkielmassa keskitytään enemmän raporttido-

kumenttien tuottamiseen ohjelmallisesti, mutta usein näihin digitaalisiin dokumentteihin on myös mahdollista upottaa kojelautamaisia ominaisuuksia, kuten kuvaajia ja tilastoja.

Raporttidokumenttien tuottamista varten monet PLM-järjestelmät tarjoavat raportointityökalun osana PLM-ohjelmistoa, jonka tarkoituksena on kerätä ja analysoida dataa kokoamalla sitä dokumenttitiedostoformaatteihin ennalta määriteltyjen sääntöjen ja mallien mukaisesti. Lisäksi raportointityökalu voi tarjota jonkinlaisen käyttöliittymän raporttien muokkaamiseen ja konfigurointiin. Alan standardina dokumenttiformaateista raporttien tapauksessa lienee PDF-, Excel- ja HTML-pohjaiset raportit, sillä useimmat raportointityökalut tarjoavat raportteja näissä tiedostomuodoissa ja ne ovat myös tuttuja suurimmalle osalle ohjelmiston käyttäjistä.

### 2.3.3 PLM-järjestelmä toimintaympäristönä

Rohleder ym. (2014) käsittelee tutkimuksessaan *"Requirements Engineering in Business Analytics for Innovation and Product Lifecycle Management"* vaatimusten määrittelyä liiketoimintatiedon hyödyntämisessä PLM-järjestelmissä. Rohleder ym. korostaa, että PLM-järjestelmissä toimitaan usein massadatan (engl. *Big data*) parissa, sillä heidän tutkimuksen mukaan yksittäisen auton tuoterakenne voi koostua noin 120 tuhannesta yksittäisestä osasta, joilla jokaisella on tyypillisesti omia CAD-malleja (tietokoneavusteisen suunnitteluohjelman luomia tiedostoja), piirustuksia ja metadataa. Lisäksi tuotteen useat versiot ja variantit kasvattavan lopullista tietomäärää eksponentiaalisesti. Lisäksi PLM-järjestelmiä käyttää tyypillisesti useita työntekijöitä yrityksen eri osastoissa, jolloin näiden työnkulkujen erilaisuus lisää PLM-järjelmien datan kompleksisuutta entisestään. Kompleksisuus johtaa usein siihen, että valmiit raportointimoottorit, varsinkin esimerkiksi taloudelliseen raportointiin erikoistuneet, eivät välttämättä sovi sellaisenaan käytettäväksi PLM:n kontekstissa. [15]

Kuten kappaleessa 2.2.1 todettiin, osaluettelot ovat keskeiseen osa PLM-järjestelmään tallennetun datan esittämistä. Koska tuoteobjektit koostuvat osaluetteloista, myös PLM-järjestelmässä tuotteista koostettavat raportit perustuvat osaluetteloista kerättyyn lähtödataan. PLM-järjestelmän raportointimoottorit ovat siten erikoistuneita jäsentelemään ja kokoamaan hierarkkista dataa. [15] PLM-järjestelmän tarjoamille raporteille on olennaista tuotteeseen ja sen kehitykseen liittyvät seikat, kuten esimerkiksi tuotteen osien toimittajien jakauma tai tuotteen muokkaushistoria. Lisäksi osaluettelon perusteella voidaan laskea yksittäisten osien summia rakenteessa tai esimerkiksi luoda raportteja tietyistä tuotteen osista, jotka täyttävät annetut kriteerit.

Mahdollisen raportointimoottorin ohella PLM-järjestelmät sisältävät usein hakumoottorin, jonka avulla voidaan etsiä tehokkaasti ja tarkasti tietokannasta annettujen kriteerien mukaisesti. Tiedon haku on yksi PLM-järjestelmän keskeisimmistä ominaisuuksista. [16] Raporttien muodostamisessa ulkoisen hakumoottorin hyödyntäminen vähentää itse raportointimoottorin kuormaa, jolloin raportointimoottorin toiminallisuuden kehittämisessä voidaan keskittyä enemmän itse laskentaan ja lisäarvon tuottamiseen. Täten PLM-järjestelmän tapauksessa lähtödatan hakeminen voi tapahtua esimerkiksi jonkin hakurajapinnan välityksellä, jolloin raportteja voidaan muodostaa tuoterakenteiden lisäksi esimerkiksi jonkin tietyn hakulausekkeen perusteella.

PLM-järjestelmien käyttäjät ovat tyypillisesti suhteellisen suuren mittakaavan teollisuusyrityksiä. Useissa tapauksissa myös tuoterakenteet ovat valtavia [15], joten raportointimoottorin tulee olla tarpeeksi tehokas ja optimoitu, jotta myös suurista tietorakenteista on mahdollista koostaa raportteja siedettävässä suoritusajassa. Raportointimoottorin logiikkatason lisäksi PLM-järjestelmien käyttäjillä on myös tarpeita muokata raporttien ulkoasua raportointimoottorin esitystasolla. Esimerkiksi yrityksen logojen ja raportin visuaalisen ilmeen muokkaaminen on olennainen osa

taas raportointityökalun toiminnallisuutta. Koska raportointityökalun tulee olla sisällytetty saumattomasti muihin PLM-järjestelmän ominaisuuksiin hyvän käyttäjäkokemuksen varmistamiseksi, useat PLM-järjestelmiä tarjoavat yritykset käyttävät PLM-järjestelmissään tätä järjestelmää varta vasten kehitettyjä raportointityökaluja. Tarvetta varta vasten kehitetylle raportointityökalulle lisää myös mainittu PLM-datan kompleksisuus, koska olemassa olevar raportontointityökaluratkaisut eivät välttämättä sovellu sellaiseenaan toimimaan kompleksisen PLM-datan kanssa.

# 3 Case-tapaus: Sovelia PLM:n raportointityökalu

## 3.1 Sovelia PLM

Kehitettävän raportointityökalun toimintaympäristönä toimii kaupallinen PLM-järjestelmä, Sovelia PLM. Sovelia PLM:n kehitys on alkanut yli 30 vuotta sitten ja sen kehitys jatkuu aktiivisesti edelleen. Kuten muutkin PLM-järjestelmät, Sovelia PLM pyrkii ratkaisemaan valmistusalan yritysten haasteita liittyen tuotteen datan hallintaan sen koko elinkaaren ajan. [17] Sovelia PLM:n erityispiirteenä on sisältämät valmiiksi konfiguroidut *"templatet"* eli valmiit mallit objekteille ja prosesseille, jotka ovat muokattavissa asiakkaan tarpeiden mukaan. Lisäksi malleihin kuuluu muita valmiiksi konfiguroituja työkaluja että alalla hyväksi todettuja prosesseja. [18]

Sovelia PLM koostuu objekteista ja objektilinkeistä. Objekteja voidaan määritellä niiden attribuuttien avulla. Objektit voivat olla osia, piirustuksia, dokumenttilinkkejä tai linkkejä muihin toimintoihin. [18] Objektilinkkien avulla voidaan muodostaa osaluetteloita, jotka ovat tärkeitä raportoinnin kohteita. Näiden lisäksi toinen merkittävä konsepti Soveliassa on käyttäjä ja käyttäjäryhmät. Yksinkertaisuudessaan järjestelmällä voi olla luonnollisesti useita käyttäjiä ja käyttäjät voivat kutsua eri käyttäjäryhmiin. Käyttäjäryhmiä voi olla useita, mutta tärkein niistä on ymmärtää *"admin"*-ryhmä (engl. *administrator*), johon kuuluu pääkäyttäjät eli järjestel-

mänvalvojat. Järjestelmänvalvojalla on luonnollisesti oikeuden muuttaa järjestelmän asetuksia. Pääkäyttäjän konsepti on tärkeä ymmärtää, sillä osa kehitettävän raportointityökalun ominaisuuksista on saatavilla vain pääkäyttäjälle. Käyttäjäryhmien lisäksi jokaisella käyttäjällä on lisenssi, joka määrittelee osaltaan käyttäjän oikeuksia rajoittamalla esimerkiksi objektien ja objektilinkkien muokkausoikeuksia. [19]

### Raportointi Sovelia PLM-järjestelmässä

Sovelia PLM ja muut PLM-järjestelmät poikkeavat valmiiden raportointityökaluratkaisujen tyypillisistä toimintaympäristöistä, sillä kuten luvussa 2.3.3 huomattiin, PLM-data on usein kompleksista ja monimuotoista. Siemensin PLM-järjestelmän verkkosivuilla julkaistussa artikkelissa *"The Challenge of Getting High Quality Reports out of PLM"* [7] (2016) esitellään haasteita, joita liittyy korkealaatuisten raporttien tuottamiseen PLM-järjestelmistä. Artikkeli nostaa esille osaluetteloiden (BOM) merkityksen PLM-järjestelmän raportoinnissa: on tärkeää että raporttien saama ja tuottama data on luotettavaa ja laadukasta, jotta päätöksenteko raporttien pohjalta olisi mahdollista. Osaluetteloihin voi tulla jopa satoja muutoksia päivittäin, joten raporttien tapauksessa on tärkeää, että niiden käyttäjät tietävät työskentelevänsä oikean datan kanssa. Täten raporttien ajantasaisuus ja selkeät aikaleimat ovat hyvin tärkeitä PLM-järjestelmän ja myös Sovelian tuottamille raporteille. Sovelia PLM:n raportointimoottorin etuna on hakurajapinnan käyttö, joka palauttaa aina hakupyynnön mukaisesti ajankohtaista dataa, mikä vähentää mahdollisten virheellisten tietojen määrää.

PLM-datan kompleksisuus tekee raportointityökalun kehittämisestä haasteellista, sillä myös Sovelia PLM vaatii erityisesti tarkoitusta varten kehitetyn raportointityökaluratkaisun. Tarkoitusta varten kehitetty ratkaisu on ohjelmistokehittäjälle työläs rakentaa alusta asti itse, mutta sen etuna on sen täysi muovautuvuus tarvetta varten. On kuitenkin todettava, että myös valmiin raportointimoottorin ja -

työkalun implementoiminen Sovelia PLM:n toimintaympäristöön olisi todennäköistä työlästä juuri PLM-datan kompleksisuuden vuoksi.

### Sovelia PLM:n vanha raportointityökalu

Sovelia PLM -järjestelmässä on tuotannossa ja asiakkailla käytössä vanha raportointiyökalu, joka tarjoaa raportteja PDF, Excel ja ZIP-tiedostoformaateissa. Näistä poikkeavimpia ovat ZIP-raportit eli ZIP-arkistot, jotka sisältävät useita PDF- tai Excel-raportteja pakattuna yhteen tiedostoon. Täten ZIP-raporttiin voidaan sisällyttää esimerkiksi raportti tuoterakenteen jokaiselle yksittäiselle objektille, jolloin ZIP-paketin kansiorakenne vastaa itsessään tuotteen rakennetta. Raporttidokumenttein data kerätään *"rakenneagentin"* avulla, joka on nimensä mukaisesti osaa kulkea objekti-linkki -suhteita pitkin ja täten kerätä tarvittavan lähdedatan raportin koostamista varten. Itse raporttidokumentteihin kirjoittaminen tapahtuu Java-ohjelmointikielen Apache FOP-ohjelmakirjaston [20] avulla XSL- (engl. *Extensible Stylesheet Language*, kieliperhe, joka mahdollistaa XML-pohjaisten tiedostojen ulkoasun ja rakennemuutoksen määrittelyn) ja XML-tietorakenteisiin tallennetun datan antaman ohjeistuksen avulla.

Vaikka tämä ratkaisu on toimiva ja edelleen kelvollinen tekniikka dokumenttien muotoiluun, ja siten raporttien generointiin ohjelmallisesti ennalta määritellyn datan mukaisesti, se kohtaa ongelmia datamäärien kasvaessa. Näihin ongelmiin kuuluu esimerkiksi liiallinen muistinkäyttö palvelimilla, prosessin hitaus varsinkin suurempien tuoterakenteiden tapauksissa ja raporttien konfiguroinnin haasteellisuus sen ollessa täysin mahdotonta PLM-järjestelmän loppukäyttäjille. Lisäksi toiminnon ylläpito alati muuttuvassa toimintaympäristössä muuttuu todennäköisesti haasteellisemmaksi tulevaisuudessa, kun uusi teknologia syrjäyttää vanhaa yhä useammilla osa-alueilla, kun vanhalle pohjalle rakennettu raportointimoottori ei voi enää pysyä muutoksen tahdissa.

### Uuden raportointityökalun haasteet

Koska vanhan raportointityökalun modernisointi ja päivittäminen nykypäivään vaatisi perustavanlaatuisia muutoksia koko järjestelmään, päädyttiin valitsemaan kehityskohteeksi kokonaan uuden raportointityökalun kehittämisen. Uuden työkalun tarkoituksena on integroitua paremmin nykyiseen Web-pohjaiseen käyttöliittymään ja tarjota käyttäjäystävällisemmän kokemuksen raporttien luomiseen. Raporttien luominen tuoterakenteista tapahtuisi käyttäjän näkökulmasta paremmin integroituna Web-käyttöliittymään tarjoten mahdollisuuden seurata meneillään olevien raporttien edistymistä sekä perua raporttien koostamisprosesseja käyttöliittymästä käsin. Lisäksi pääkäyttäjällä tulisi olla mahdollisuus muokata ja luoda uusia raporttityyppejä muokkaamalla esimerkiksi raporteissa esitettäviä sarakkeita ja tuoterakenteiden suodattamiseen liittyviä sääntöjä. Myös raporttien ulkonäön tulee olla muokattavissa pääkäyttäjän toimesta, mikä tarkoittaa niiden visuaalisen ilmeen muokkausta lisäämällä asiakasyrityksen logoja, muokkaamalla väriteemaa ja asettelua asiakkaan tarpeiden mukaisesti.

Uuden raportointityökalun tulee ensimmäisessä vaiheessa sisältää samat toiminnallisuudet kuin vanha raportointityökalu, mutta tarkoituksena on tuottaa raportteja huomattavasti nopeammin aiheuttamalla samanaikaisesti vähemmän kuormaa palvelimelle. Raporttien sisällön ja yleisen rakenteen tulee vastata vanhempia raportteja identtisesti, mutta raporttien visuaalinen ilme tulee päivittää nykypäivän standardien mukaiseksi. Lisäksi vaaditaan syvällisiä teknologiatutkimusta ja -analyysiä olemassa olevien raportointityökalujen ja -moottorien ratkaisuista, Sovelia PLM:n ohjelmakannassa jo olemassa olevista komponenteista ja muista mahdollisesti hyödyllisistä teknologioista.

## 3.2 Nykyiset erilliset raportointityökalut

Ennen käytettävien teknologioiden ja tekniikoiden valintaa on hyvä perehtyä jo olemassa oleviin raportointityökaluihin ja -moottoreihin, jotta voidaan muodostaa kokonaiskuva yleisesti käytössä olevista ratkaisuista. Hyvän kokonaiskuvan avulla voidaan tehdä valistuneita päätöksiä siitä, miten raportointityökalua kannattaa lähteä kehittämään sekä ymmärtää käytössä olevien tekniikoiden hyötyjä ja rajoituksia PLM-järjestelmän kontekstissa. Analysoimme kuutta suosittua itsenäistä (engl. *standalone*) raportointityökalua, joista neljä ovat kaupallisia ja kaksi ovat avoimen lähdekoodin ohjelmistoja. Juuri nämä kuusi raportointityökalua valittiin tarkasteltavaksi, sillä ne ovat kaikki paikallisesti asennettavia ohjelmistoja (engl. *On-premise software*) kuten myös Sovelia PLM. Täten tässä tutkimuksessa ei keskitytä esimerkiksi ulkoisten pilvipalveluiden raportointityökalu verkko-ohjelmointirajapintoihin (engl. *Web API*). Yksi esimerkki tällaisesta verkkopohjaisesta API:sta on *Bold Reports* [21], joka tarjoaa monipuolisen verkkorajapinnan kautta toimivan raportointityökalun. Toinen kriteeri tarkasteluun valinnalle oli käytettävä teknologia, sillä Sovelia PLM:n nykyinen arkkitehtuuri nojaa vahvasti Java- ja JavaScript-ohjelmointikieliin, joten täysin uuden ohjelmointikielen lisäämistä järjestelmäarkkitehtuurin pinoon ylläpidettäväksi ei pidetty mielekkäänä vaihtoehtona. Lisäksi esimerkkejä sekä Javaettä JavaScript-pohjaisia raportointityökaluista löytyi runsaasti, joten uuden teknologian valitseminen ei vaikuttanut välttämättömyydeltä. Kolmas tärkeä kriteeri oli ulostulodataformaattien monipuolisuus: koska Sovelia PLM -järjestelmään kehitettävän raportointityökalun tulee tuottaa raportteja eri tiedostoformaateissa, tulee myös tarkasteltavien työkalujenkin.

Puhtaasti kaupallisia ohjelmistoja ovat JavaScript-pohjainen *ActiveReportsJS*, SAP:in, maailman suurimman yritysohjelmistoja tarjoavan yrityksen [22], kaksi raportointityökalua *SAP Business Objects* ja *SAP Crystal Reports* ja Oraclen, talouslehti Forbesin mukaan vuoden 2023 80. suurimman julkisen yrityksen [23], rapor-

tointityökalu *Orancle BI Publisher*. Vahva esimerkki taas avoimen lähdekoodin vaihtoehdosta on *JasperReports*, joka kutsuu itseään suosituimmaksi avoimen lähdekoodin Java-raportointikirjastoksi. [24]. *JasperReports* sisältää myös visuaalisen suunnittelutyökalun, *iReport Designerin*, joten *JasperReports* on kokonainen raportointityökaluohjelmisto. *JasperReports* sisältää myös vaihtoehtoisen maksullisen lisenssin yrityskäyttöön tarjoten yritysasiakkaille parempaa tukea ja ratkaisua suuremman skaalan raportointiin. [25] Toinen avoimen lähdekoodin raportointityökalu on myös Java-pohjainen *BIRT* (lyhenne engl. *"Business Intelligence Reporting Tool"*), joka tarjoaa monipuolisia raportointimahdollisuuksia täysin avoimen lähdekoodin ja Eurooppalaisen *Eclipse Foundationin* ylläpitämä ohjelmisto. [26] *BIRT* on keskittynyt raportoinnin lisäksi myös Business Intelligencen hyödyntämiseen asiakas- ja web-applikaatioissa, joten se sisältää monia erilaisia visualisointimahdollisuuksia ja raportointimoottorin lisäksi myös graafisen raporttikonfigurointityökalun.

### Nykyisten raportointityökalujen yleiskuva

Raportointityökalua kehittääkseen osaksi PLM-järjestelmää varten tulee lisenssien lisäksi ottaa huomioon myös muita ominaisuuksia. Liitteenä olevassa taulukossa A.1 luokitellaan raportointiyökalujen yleisiä ominaisuuksia. Taulukoiden data on kerätty taulukossa mainituista lähteistä, kuten ohjelmiston verkkosivuilta, ohjelmakirjaston dokumentaatiosta tai myyntiesitteistä. Jo mainitut lisenssit ovat eritelty kaupallisiin ja avoimen lähdekoodin ohjelmistoihin. Avoimen lähdekoodin ohjelmistot tarjoavat asiantuntijoille lähdekoodin perehtymisen avulla syvän katsauksen raportointityökalun sisäiseen toiminnallisuuteen, mutta tässä tutkimuksessa keskitymme enemmän pintapuolisempiin ominaisuuksiin ja teknologiavalintoihin tarkemman lähdekoodianalyysin sijasta. Yksi tällaisista ominaisuuksista on lähdedatan formaatti ja mahdolliset datalähteet. Kaikki tutkitut raportointityökalut tukivat jotakin rakenteellisen datan tallentamiseen suunniteltua tiedostoformaattia eli tässä tapaukses-

sa XML:ää tai JSON:ia. Ainoastaan *(*ActiveReportsJS) ei tukenut XML:ää, mutta tukee kuitenkin luonnollisesti JavaScript-pohjaisena JSON:ia. Rakenteellisen datan tukeminen tärkeää varsinkin PLM:n datan tapauksessa, jossa objektien ja siten tuotteiden väliset suhteet ovat avainasemassa. Näiden tiedostoformaattien lisäksi noin puolet tarkasteltavista raportointityökaluista tuki myös SQL-kyselyitä tietolähteenä (engl. *Structured Query Language*). Näiden lisäksi myös tavallisten tekstitiedostojen käyttö tietolähteenä on tyypillistä XML- ja JSON-tiedostojen lisäksi, kuten myös Excel-taulukkolaskentaohjelman tuottamat XLSX-tiedostot. Mielenkiintoista on, että myös Excelin tuottamat XLSX-tiedostot koostuvat itseasiasiassa vain ZIP-pakatuista XML-tiedostoista. [27]

Raportointityökalujen ohjelmointikieliä tarkastellessa suosituimmaksi ilmeni Java, sillä kaikki paitsi *(*ActiveReportsJS) olivat Java-pohjaisia. Yksi syy tähän voi olla Javan suuri ja vakaa suosio 1990-lopussa ja 2000-luvun alussa ohjelmointikielten suosiota mittaavan *TIOBE indexin* mukaan[28], jolloin suurinta osaa näistä raportointityökaluista alettu kehittää. Esimerkiksi *JasperReportsia* on alettu kehittää vuonna 2001 [29], *BIRTiä* vuonna 2004 [30] ja *BusinessObjectsia* jo vuonna 1995 ennen kuin se kuului SAP-tuoteperheeseen [31]. Kuten TIOBE indexistä voidaan havaita, on Java edelleen suhteellisen suosittu kieli käyttää vuonna 2023, mutta sen suosio on kuitenkin jo hieman laskussa. Raporttipohjien konfigurointi tapahtuu raportointityökaluissa pääsääntöisesti jonkin käyttöliittymän avulla. Käyttöliittymän lisäksi *ActiveReportsJS* tarjoaa mahdollisuuden konfiguroida raporttipohjia JSON-datan avulla, *BIRT* ja *JasperReports* XML-datan avulla ja *Oracle BI Publisher* RTF (engl. *Rich Text Format*, eroaa tavallisesta tekstistä tallentamalla tiedostoon myös fontit, fonttikoot ja yleisen muotoilun, mm. kursivoinnin) ja XSL-pohjaisesti. Tästä voidaan havaita, että visuaalinen käyttöliittymä raporttipohjien luomiselle on suosituin tapa toimia, mutta tiedostopohjainen raporttipohjien muokkaus on käypä vaihtoehto.

### Nykyisten raportointityökalujen ulostuloformaatit

Raportointityökalun tärkein tehtävä on tuottaa raportteja, jolloin raporttien tulee olla myös helposti avattavissa ja esitettävissä eri laitteilla. Liitteenä olevassa taulukossa B.1 tarkastellaan samojen kuuden raportointityökalun tarjoamia ulostuloformaatteja, eli sitä millaisissa tiedostomuodoissa työkalut tarjoavat raportteja. Ulostuloformaattien valinta on erityisen tärkeää raportointityökalulle, sillä sen käyttäjät suosivat heille jo entuudestaan tuttuja tiedostoformaatteja, sekä ohjelmistoteknisestä näkökulmasta, sillä raporttien tuottaminen erilaisissa tiedostoformaateissa vaatii usein tiedostoformaattikohtaisia työkaluja.

Ulostuloformaateista erityisen suosittu on Adoben kehittämä PDF (engl. *Portable Document Format*), jota käytetään erityisesti sähköiseen julkaisemiseen. PDF on ollut ISO-standardi vuodesta 2007 alkaen ja ISO (engl. *International Organization for Standardization*, kansainvälinen standardisoimisjärjestö) kuvailee sitä maailman suosituimmaksi dokumenttitiedostoformaatiksi [32]. Suosionsa ja standardoinnin vuoksi PDF onkin luonnollinen valinta raportointityökalun ulostuloformaatille ja siksi myös kaikki kuusi tarkastelevaa raportointityökalua tarjoavat raportteja PDF-muodossa.

Raporttidokumenteille selkeästi suosittuja alustoja ovat erityisesti yritysmaailmassa suositun Microsoft Officen tuottamat tiedostomuodot, erityisesti taulukkolaskentaohjelma Excelin *.xlsx* ja *.xls* ja tekstinkäsittelyohjelma Wordin tuottamat *.docx* ja *.doc* tiedostot. Microsoft Officen tiedostomuotojen tuki ei ole varsinaisesti yllättävää ottaen huomioon, että tilastopalvelu *Statistan* mukaan Microsoftilla on noin 45 prosentin markkinaosuus toimisto-ohjelmistojen alalla Googlella sen ollessa noin 50. [33] On kuitenkin otettava huomioon, että myös Google tarjoaa omissa toimisto-ohjelmistoissaan mahdollisuuden avata ja muokata myös Microsoft Officen tuottamissa tiedostomuodoissa olevia tiedostoja. Varsinkin taulukkomuotoisten raporttien tapauksessa *.xlsx*-tiedostot ovat hyödyllisiä, sillä ne tarjoavat raportointityökalun

käyttäjälle mahdollisuuden tutkia raportin sisältöä vielä pintaa syvemmältä suorittamalla itse laskentaa ja kokoamalla kuvaajia ohjelman avulla. Raportteja Excel- ja Word-tiedostoformaateissa tarjosivat kaikki tarkasteltavat raportointityökalut paitsi ActiveReportsJS, eli nämä formaatit ovat PDF-raporttien ohella erityisen suosittuja.

Kolmas suosittu tiedostomuoto on HTML (engl. *Hypertext Markup Language*, suom. *hypertekstin merkintäkieli*) , joka on erityisesti verkkosivujen kirjoittamisessa käytetty merkintäkieli. HTML ei ole ohjelmointikieli vaan merkintäkieli, joka määrittelee miten sisältö tulisi muotoilla. HTML:ää käytetään usein yhteydessä CSS:n kanssa (engl. *Cascacing Style Sheets*, suom. *porrastetut tyyliarkit*) . CSS on eräänlainen tyylisivu, jolla voidaan määritellä tyyliohjeita siitä, miten esimerkiksi HTML dokumentti voidaan esittää. Kuten HTML myöskään CSS ei ole ohjelmointikieli, mutta se tarjoaa valtavasti mahdollisuuksia muotoilla HTML-dokumentteja. HTML:n ja CSS:n ohella verkkosivut hyödyntävät tyypillisesti myös JavaScriptia. HTML-raportit tarkoitettu esitettäväksi verkkoselaimessa, mikä tekee niistä hyvin alustariippumattomia. Mikäli HTML-raportit sisältävät CSS-tyyliarkin ja JavaScriptia, näitä hyödyntämällä HTML raportteihin voidaan lisätä animaatioita ja muita toiminnallisuuksia, ne voivat olla huomattavasti interaktiivisempia käyttäjälle verrattuna staattisempiin PDF-, Excel ja Word-raporttidokumentteihin. HTML-pohjaisia raportteja tarjosivat poikkeuksetta kaikki tarkasteltavat kuusi raportointityökalua. Tämä ei ole kovinkaan yllättävää, sillä HTML-pohjien (eng. *HTML-template*) luominen käytettäväksi raporttipohjina sekä HTML-raporttien kasaaminen ohjelmallisesti on yksinkertaista toteuttaa nykyaikaisin menetelmin. Yksi esimerkki tällaisesta HTML-pohja-moottorista (engl. *HTML templating engine*) on useille ohjelmointikielille saatavilla oleva *Mustache* [34], joka tarjoaa työkaluja muodostaa esimerkiksi HTML-raportteja määritellyn mallin mukaisesti vaihtelevasta lähtödatasta.

Viimeisenä ulostuloformaattina

## 3.3 Uuden raportointityökalun kehitys

Ensimmäinen haaste raportointityökalun kehittämiselle on käytettävien tekniikoiden ja teknologioiden valinta. Koska tarkoituksena on kehittää täysin uusi toiminnallisuus alusta alkaen, on tärkeää ymmärtää nykyiset Sovelia PLM:ssä käytössä olevat komponentit palvelinarkkitehtuurien näkökulmasta. Itse raportointimoottori täytyy toimia jollakin palvelimella koostaa valmiit raportit palvelinpäässä ennen niiden lataamista käyttäjien saataville. Tällä menettelyllä voidaan vähentää asiakaspään kuormaa siirtämällä raskaampi laskenta palvelimen vastuulle. Raportointimoottorin palvelinympäristöksi valiutui Sovelia PLM:ssä jo käytössä oleva Node.js, joka tarjoaa erilaisia palveluita ja rajapintoja Sovelia PLM:n Web-käyttöliittymälle. Näihin palveluihin kuuluu esimerkiksi 3D-näkymien ja -tiedostojen tarjoaminen Web-käyttöliittymälle, joten myös raportointimoottorin sisällyttäminen samalle palvelimelle oli luonteva ratkaisu. Node.js on avoimen lähdekoodin alustariippumaton ajoympäristö JavaScript-koodin suorittamiseen palvelimella ja se on Stack Overflown teettämän kyselyn mukaan yleisin käytetty Web-teknologioita ammattikehittäjien keskuudessa React.js:n ohella n. 42 prosentin käyttöasteella. [35] Tärkeää on huomata, että Node.js on vuoden 2023 johtava teknologia JavaScriptin käyttöön palvelinpuolella, sillä React.js on tarkoitettu enemmänkin käyttöliittymien kehittämiseen asiakaspäässä. Node.js tarjoaa oletuksena paketinhallintajärjestelmän nimeltään NPM (engl. *Node Package Manager*, tyyliteltynä npm), joka koostuu komentorivityökalusta ja JavaScript -ohjelmapaketteja ylläpitävästä tietokannasta. NPM:n avulla erillisten ohjelmakirjastojen sisällyttäminen ja käyttöönotto on yksinkertaista ja nopeaa. [36]

Node.js on siis hyvin suosittu ja monipuolinen ympäristö. Sen suosion vuoksi Node.js:n tapauksessa käytettävissä on laaja kattaus erilaisia hyödyllisiä ohjelmakirjastoja.

**Mitä voimme ottaa mukaan nykyisistä ratkaisuista ja mikä todettiin toimivaksi?**

**Mitä emme voi hyödyntää ja miksi?**

# Lähdeluettelo


[1] M. David ja F. Rowe, "What does PLMS (product lifecycle management systems) manage: Data or documents? Complementarity and contingency for SMEs", *Computers in Industry*, vol. 75, s. 140–150, tammikuu 2016, ISSN: 0166-3615. DOI: 10.1016/j.compind.2015.05.005. url: https://www.sciencedirect.com/science/article/pii/S0166361515300051 (viitattu 17. 09. 2023).

[2] I. Bouhaddou, A. Benabdelhafid, L. Ouzizi ja Y. Benghabrit, "PLM (Product Lifecycle Management) Model for Supply Chain Optimization", en, teoksessa *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, L. Rivest, A. Bouras ja B. Louhichi, toim., sarja IFIP Advances in Information and Communication Technology, Berlin, Heidelberg: Springer, 2012, s. 134–146, ISBN: 978-3-642-35758-9. DOI: 10.1007/978-3-642-35758-9_12.

[3] M. Alemanni, G. Alessia, S. Tornincasa ja E. Vezzetti, "Key performance indicators for PLM benefits evaluation: The Alcatel Alenia Space case study", *Computers in Industry*, vol. 59, nro 8, s. 833–841, lokakuu 2008, ISSN: 0166-3615. DOI: 10.1016/j.compind.2008.06.003. url: https://www.sciencedirect.com/science/article/pii/S0166361508000663 (viitattu 17. 10. 2023).

[4] L. Rivest, A. Bouras ja B. Louhichi, toim., *Product Lifecycle Management. Towards Knowledge-Rich Enterprises: IFIP WG 5.1 International Conference,*

*PLM 2012, Montreal, QC, Canada, July 9-11, 2012, Revised Selected Papers* (IFIP Advances in Information and Communication Technology), en. Berlin, Heidelberg: Springer, 2012, vol. 388, ISBN: 978-3-642-35757-2 978-3-642-35758-9. DOI: 10.1007/978-3-642-35758-9. url: http://link.springer.com/10.1007/978-3-642-35758-9 (viitattu 17. 10. 2023).

[5] S. G. Lee, Y. .-. Ma, G. L. Thimm ja J. Verstraeten, "Product lifecycle management in aviation maintenance, repair and overhaul", *Computers in Industry*, Product Lifecycle Modelling, Analysis and Management, vol. 59, nro 2, s. 296–303, maaliskuu 2008, ISSN: 0166-3615. DOI: 10.1016/j.compind.2007.06.022. url: https://www.sciencedirect.com/science/article/pii/S0166361507001108 (viitattu 18. 10. 2023).

[6] R. Jones ja L. Tate, *Visualizing Comparisons of Bills of Materials*, arXiv:2309.11620 [cs], syyskuu 2023. DOI: 10.48550/arXiv.2309.11620. url: http://arxiv.org/abs/2309.11620 (viitattu 17. 10. 2023).

[7] K. German, *The Challenge of Getting High Quality Reports out of PLM - Teamcenter*, en-US, Section: News, tammikuu 2016. url: https://blogs.sw.siemens.com/teamcenter/the-challenge-of-getting-high-quality-reports-out-of-plm/ (viitattu 17. 10. 2023).

[8] M. Bosch-Mauchand, M. Bricogne, B. Eynard ja J.-P. Gitto, "Preliminary Requirements and Architecture Definition for Integration of PLM and Business Intelligence Systems", en, teoksessa *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochano ja E. Hancock, toim., vol. 8827, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, s. 265–272, ISBN: 978-3-319-12567-1 978-3-319-12568-8. DOI: 10.1007/978-3-662-44739-0_33. url: http://link.springer.com/10.1007/978-3-662-44739-0_33 (viitattu 22. 10. 2023).

[9] Y. He ja F. Gong, ”Design and Implementation of the Large Enterprise Reporting Engine”, teoksessa *2010 International Conference on Web Information Systems and Mining*, vol. 2, lokakuu 2010, s. 235–238. DOI: 10.1109/WISM.2010.96. url: https://ieeexplore.ieee.org/document/5662268 (viitattu 09. 10. 2023).

[10] B. P. Adhi, D. N. Prasetya ja Widodo, ”Performance comparison of reporting engine birt, jasper report, and crystal report on the process business intelligence”, English, *IOP Conference Series. Materials Science and Engineering*, vol. 508, nro 1, huhtikuu 2019, Place: Bristol, United Kingdom Publisher: IOP Publishing, ISSN: 17578981. DOI: 10.1088/1757-899X/508/1/012129. url: https://www.proquest.com/docview/2560973452/abstract/E835BABAB8FE44D6PQ/1 (viitattu 09. 10. 2023).

[11] N. Nurseitov, M. Paulson, R. Reynolds ja C. Izurieta, ”Comparison of JSON and XML Data Interchange Formats: A Case Study”, en,

[12] Y. Niu, L. Ying, J. Yang, M. Bao ja C. B. Sivaparthipan, ”Organizational business intelligence and decision making using big data analytics”, *Information Processing & Management*, vol. 58, nro 6, s. 102 725, marraskuu 2021, ISSN: 0306-4573. DOI: 10.1016/j.ipm.2021.102725. url: https://www.sciencedirect.com/science/article/pii/S0306457321002090 (viitattu 22. 10. 2023).

[13] L. Raković, M. Sakal ja P. Matković, ”Digital workplace: Advantages and challenges”, en, *Anali Ekonomskog fakulteta u Subotici*, nro 47, s. 65–78, 2022, ISSN: 0350-2120, 2683-4162. DOI: 10.5937/AnEkSub2247065R. url: https://scindeks.ceon.rs/Article.aspx?artid=0350-21202247065R (viitattu 22. 10. 2023).

[14] M.-F. Sriti ja P. Boutinaud, ”PLMXQuery: Towards a Standard PLM Querying Approach”, eng, teoksessa *IFIP Advances in Information and Communication Technology*, vol. AICT-388, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 379–388, ISBN: 9783642357572.

[15] C. Rohleder, J. Lin, I. Kusumah ja G. Özkan, ”Requirements Engineering in Business Analytics for Innovation and Product Lifecycle Management”, en, teoksessa *Advances in Conceptual Modeling*, J. Parsons ja D. Chiu, toim., sarja Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, s. 51–58, ISBN: 978-3-319-14139-8. DOI: 10.1007/978-3-319-14139-8_7.

[16] J. G. Enríquez, J. M. Sánchez-Begines, F. J. Domínguez-Mayo, J. A. García-García ja M. J. Escalona, ”An approach to characterize and evaluate the quality of Product Lifecycle Management Software Systems”, *Computer Standards & Interfaces*, vol. 61, s. 77–88, tammikuu 2019, ISSN: 0920-5489. DOI: 10.1016/j.csi.2018.05.003. url: https://www.sciencedirect.com/science/article/pii/S0920548917303239 (viitattu 17. 11. 2023).

[17] *About Sovelia and one digital platform | sovelia.com — sovelia.com*, https://sovelia.com/about-sovelia. (viitattu 17. 11. 2023).

[18] *Sovelia PLM getting started - How to use Sovelia PLM — help.sovelia.com*, https://help.sovelia.com/docs/getting-started. (viitattu 17. 11. 2023).

[19] *User management — help.sovelia.com*, https://help.sovelia.com/docs/user-management. (viitattu 17. 11. 2023).

[20] *Apache(tm) FOP - a print formatter driven by XSL formatting objects (XSL-FO) and an output independent formatter.* en. url: https://xmlgraphics.apache.org/fop/ (viitattu 19. 11. 2023).

[21] *BI Report Management System & Reporting Tools | Bold Reports*, en-US, syyskuu 2022. url: https://www.boldreports.com (viitattu 03. 12. 2023).

[22] *SAP North America at a Glance: The World's Largest Provider of Enterprise Application Software*, English. url: https://www.sap.com/documents/2014/12/e4198023-0a7c-0010-82c7-eda71af511fa.html (viitattu 28. 11. 2023).

[23] ". M. TUCKER" " HANK, *The Global 2000 2023*, en. url: https://www.forbes.com/lists/global2000/ (viitattu 28. 11. 2023).

[24] *JasperReports Library Datasheet*, en. url: https://www.jaspersoft.com/resources/whitepaper/jasperreports-library-datasheet (viitattu 28. 11. 2023).

[25] *Reporting and embedded business intelligence software*, en. url: https://www.jaspersoft.com/ (viitattu 03. 12. 2023).

[26] *BIRT - Business Intelligence Reporting Tool — eclipse-birt.github.io*, https://eclipse-birt.github.io/birt-website/. (viitattu 23. 11. 2023).

[27] M. Miner, *Under the Hood: Excel Files as Zip Folders - Miner Curiosity*, English, Blog, heinäkuu 2022. url: https://minerupset.com/2022/Excel-Files-As-Zip-Folders/ (viitattu 06. 12. 2023).

[28] *TIOBE Index for November 2023*, en-US. url: https://www.tiobe.com/tiobe-index/ (viitattu 06. 12. 2023).

[29] *Origin date for JasperReports?*, en-US, syyskuu 2006. url: https://community.jaspersoft.com/forums/topic/13224-origin-date-for-jasperreports/ (viitattu 06. 12. 2023).

[30] *Eclipse Foundation and Actuate Announce Approval of Business Intelligence and Reporting Tools Project*, English, kesäkuu 2004. url: https://www.eclipse.org/org/press-release/oct62004birtpr.html (viitattu 06. 12. 2023).

[31] T. M. Mahmoud, *Creating universes with SAP BusinessObjects : create and maintain powerful SAP BusinessObjects Universes with the SAP Information Design Tool* (Professional expertise distilled), eng, 1st edition. Birmingham, [England: Packt Publishing, 2014 - 2014, ISBN: 1-78217-091-X.

[32] C. Naden, *The standard for PDF is revised*, en, tammikuu 2021. url: `https://www.iso.org/cms/render/live/en/sites/isoorg/contents/news/2021/01/Ref2608.html` (viitattu 06. 12. 2023).

[33] *Office productivity software global market share 2022*, en. url: `https://www.statista.com/statistics/983299/worldwide-market-share-of-office-productivity-software/` (viitattu 09. 12. 2023).

[34] *{{ mustache }}*. url: `https://mustache.github.io/` (viitattu 10. 12. 2023).

[35] *Stack Overflow Developer Survey 2023 — survey.stackoverflow.co*, `https://survey.stackoverflow.co/2023/`. (viitattu 23. 11. 2023).

[36] *npm About — npmjs.com*, `https://www.npmjs.com/about`. (viitattu 23. 11. 2023).

[37] *ActiveReportsJS: Introduction — developer.mescius.com*, `https://developer.mescius.com/activereportsjs/docs/`. (viitattu 23. 11. 2023).

[38] "SAP BusinessObjects RESTful Web Service SDK User Guide for Web Intelligence and the BI Semantic Layer", (viitattu 23. 11. 2023).

[39] "SAP Crystal Reports RESTful Web Services Developer Guide", en, (viitattu 23. 11. 2023).

[40] P. Sharma ja K. McDermott, "Oracle Business Intelligence Publisher Overview & Best Practices", en, 2014.

# Liite A Yleistä raportointityökaluista

| **Raportointi-työkalu** | **Lisenssi** | **Datalähde** | **Ohjelmointi-kieli** | **Raporttipohjien konfiguurointi** | **Asennus** |
|---|---|---|---|---|---|
| ActiveReportsJS [37] | kaupallinen | JSON | JavaScript | UI /JSON | paikallinen |
| SAP BusinessObjects [38] | kaupallinen | XML ja JSON | Java | UI | paikallinen tai pilvipalvelu |
| SAP Crystal Reports [39] | kaupallinen | tiedostot mm. XLSX, TXT, XML ja tietokannat (SQL) | Java | UI | paikallinen tai pilvipalvelu |
| Oracle BI Publisher [40] | kaupallinen | SQL, XML, XLSX ja Oracle BI:n omat formaatit | Java | UI, RTF ja XSL | paikallinen |
| JasperReports [24] | avoin lähdekoodi, mahdollisuus kaupalliseen lisenssiin | JavaBeans, Java object, XML ja mukautetut tietolähteet | Java | UI / XML tai .jasper | paikallinen |
| BIRT Project [26] | avoin lähdekoodi | XML, TXT ja tietokannat (SQL) | Java | UI / XML | paikallinen |

Taulukko A.1: Yleistä raportointityökaluista.

# Liite B Raportointityökalujen ulostuloformaatit

| **Raportointityökalu** | **PDF (.pdf)** | **XLSX (.xlsx, .xls)** | **HTML (.html)** | **Word (.docx, .doc)** | **RPT (.rpt)** |
|---|---|---|---|---|---|
| ActiveReportsJS [37] | x | | x | | |
| SAP BusinessObjects [38] | x | x | x | x | x |
| SAP Crystal Reports [39] | x | x | x | x | x |
| Oracle BI Publisher [40] | x | x | x | x | |
| JasperReports [24] | x | x | x | x | |
| BIRT Project [26] | x | x | x | x | |

Taulukko B.1: Raportointityökalujen ulostuloformaatit.