



Import-työkalun toiminnallisuuksien laajentaminen

Tommi Seppä

OPINNÄYTETYÖ
Joulukuu 2021

Tieto- ja viestintäteknikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintäteknikka
Ohjelmistotekniikka

SEPPÄ, TOMMI
Import-työkalun toiminnallisuuksien laajentaminen

Opinnäytetyö 26 sivua, joista liitteitä 0 sivua
Joulukuu 2021

Opinnäytetyön tarkoituksena oli laajentaa olemassa olevan Import-työkalun toiminnallisuuksia vastaamaan käyttäjien nykyistä tarvetta. Työkalulla tuodaan PLM XML- ja 3D CAD -muotoista dataa tuotteen elinkaaren hallintaohjelmisto Teamcenteriin. Työkalu toi työn aloitushetkellä kaikki nimikkeet vain yhdellä julkaisutilalla järjestelmään. Tämä oli ongelmallista, sillä vain yhden julkaisutilan käyttö aiheutti datassa puutteita. Jatkossa käyttäjien tuli voida valita julkaisutila, millä nimikkeet järjestelmään vietäisiin. Työn aikana päivitettiin myös työkalun dokumentaatiota.

Opinnäytetyössä tutustuttiin olemassa olevan ratkaisun lähdekoodiin sekä Teamcenterin partnereilleen tarjoamiin kustomointimenetelmiin ja -vaihtoehtoihin. Työkalun eri versioiden testaamisessa käytettiin tarkoitukseen muokattua olemassa olevaa virtuaaliympäristöä. Virtuaaliympäristön käyttö osoittautui hyväksi tavaksi testata työkalun toimivuutta ja sen tarvitsemia konfiguraatiomuutoksia.

Import-työkalun toimintaan tutustumisen perusteella tunnistettiin tarvittavat muutokset web-portaalin ja Teamcenterin dispatcher-moduulin translaattorissa. Java ja JavaScript-pohjaiseen web-portaaliin lisättiin tarvittavat toiminnallisuudet julkaisutilan käsittelemiseksi. Lisäksi Siemensin C/C++ -integraatorajapinnan avulla ohjelmoitiin kaksi komentokehotetyökalua, jotka mahdollistivat julkaisutilojen tarkistamisen ja rakenteen tilavedoksen luomisen.

Projektin lopputuloksena saatiin työkalu, jolla rakenteet voidaan viedä valitulla julkaisutilalla järjestelmään ja tuodun rakenteen tilavedos on helposti saatavilla. Lisäksi työkalun dokumentaatio ja lähdekoodit saatettiin ajan tasalle, mikä helpottaa työskentelyä jatkossa. Työkaluun jäi vielä parannettavaa. Tällä hetkellä tietoa joudutaan esimerkiksi jalostamaan useassa vaiheessa. Nämä voisi mahdollisesti yhdistää yhden toiminnallisuuden alle, jolloin ylimääräisistä vaiheista päästäisiin eroon.

Asiasanat: tuotteen elinkaaren hallinta, tuotetiedon hallinta, plm, pdm, tuote-tieto, teamcenter, integraatio

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
ICT Engineering
Software Engineering

SEPPÄ TOMMI
Expanding the capabilities of Import-tool

Bachelor's thesis 26 pages, appendices 0 pages
December 2021

The purpose of this thesis work was to expand the capabilities of existing import tool to match the current needs of the users. The tool is used to import product data in PLM XML and 3D CAD formats to Product Lifecycle Management software Teamcenter. At the current state of the tool, only one release status is supported for the imported products. This is problematic since it can cause defects to the data already in the system. In the future users should be able to select the release status to use for the products imported to the system. During this thesis work the documentation was reviewed and updated when necessary.

Development of the new solution started with examining the current source code for the solutions and the coding APIs and customization options Siemens provides for the Teamcenter system. For testing the current and new solution, a virtual machine environment was modified to fit this purpose. Using virtual environment proved to be a very good way of testing how the tool was working and the configurations, that it needed.

Based on the investigation conducted, changes needed for the solution were found in the web portal and Teamcenter dispatcher translator module. The required functionalities to process the selected release status were added to the Java/JavaScript based web portal. For the dispatcher translator, two command line tools were created using the Siemen integration toolkit C/C++ API. These tools made it possible to check and set the release statuses from Teamcenter and take a snapshot of the imported structure.

As an outcome of the project, the import tool can now import product structures with different releases statuses to Teamcenter and create easily accessible snapshot of the product structure. There is still room for improvement in the import tool. Currently, for example the input data is enriched in multiple stages, and it could be possible to do these in one step, making the process leaner. In addition, the up-to-date documentation and source code makes it easier to familiarize yourself to this tool in the future and thus easier for others to make changes.

Key words: product lifecycle management, product data, product data management, pdm, plm, teamcenter, customization, integration

SISÄLLYS

1	JOHDANTO	6
2	TEORIA	7
	2.1 Tuotteen elinkaaren hallinta	7
	2.2 Tuotteen elinkaaren hallintaohjelmat.....	8
	2.3 Nimiketieto	10
3	TEKNISET RATKAISUT	13
	3.1 Teamcenterin rakenne	13
	3.2 Teamcenterin kustomointi	14
	3.3 PLM XML	16
4	TOTEUTUS	18
	4.1 Lähdekoodiin perehtyminen	18
	4.2 Lähtötilanne	18
	4.3 Testijärjestelmän pystyttäminen.....	19
	4.4 Työkalun uusi käyttäytyminen	20
	4.5 Web-portaalin muutokset	21
	4.6 Dispatcher muutokset	22
	4.7 Käännösprosessin muutokset	22
5	POHDINTA	24
	LÄHTEET.....	25

LYHENTEET JA TERMIT

Active Workspace	Teamcenterin web-asiakasohjelma.
ALM	Application Lifecycle Management
APS	Advanced Planning and Scheduling
CAD	Computer-Aided Design and Drafting
Dispatcher	Teamcenterin komponentti, jolla voidaan tausta-ajossa tehdä mm. tiedostomuunnoksia.
ERP	Enterprise Resource Planning
ITK	Integrated Development Kit
JT	Jupiter Tessellation. 3D data file format.
MES	Manufacturing Execution System
MRO	Maintenance Repair and Operations
NX	Siemens CAD program.
PDM	Product Data Management
PLM	Product Lifecycle Management
PLM XML	Siemensin tuotetiedon hallintaformaatti. Extensible Markup Language
Rich Client	Teamcenter tietokoneelle asennettava asiakasohjelma.
SCM	Supply Chain Management

1 JOHDANTO

Tuotteen elinkaaren hallintaohjelmien (eng. Product Lifecycle Management, PLM) rooli yrityksissä on korostunut tuotteiden ja yritysten prosessien tuottaessa yhä enemmän tietoa, jota voidaan hyödyntää toiminnan parantamisessa. Tuotteiden elinkaaren hallintajärjestelmillä ei välttämättä tuoteta suoraan kaikkea tuotteen elinkaaren aikana syntynyttä tietoa, vaan tietoa voidaan tuoda niihin useista lähteistä. Opinnäytetyössä laajennettiin ominaisuuksia työkaluun, jolla PLM järjestelmään voidaan tuoda nimikkeiden rakenteita 3D CAD -malleineen.

Opinnäytetyön aihe sai alkunsa IDEAL GRP -yrityksen tarpeesta parantaa käytössä olevan Import-työkalun toiminnallisuuksia vastaamaan paremmin käyttötarpeita. Käyttäjiltä tulleen palautteen perusteella, työkaluun tarvittiin mahdollisuus viedä nimikkeitä järjestelmään käyttäjän valitsemassa julkaisutilassa. Lisäksi sen tulisi tehdä tuodusta rakenteesta helposti saatavilla oleva tilavedos. Tällä hetkellä työkalu korvaa olemassa olevat julkaisutilat, joka aiheuttaa ongelmia datan eheyteen.

Opinnäytetyön alkuosassa tutustutaan tuotteen elinkaaren hallintaan ja tuotetiedon perusteisiin, joiden rooli yrityksissä on korostunut tuotteiden ja yritysten prosessien tuottaessa yhä enemmän tietoa, jota voidaan hyödyntää toiminnan parantamisessa. Opinnäytetyön loppuosassa perehdytään Teamcenter PLM -järjestelmän rakenteeseen ja kustomointi vaihtoehtoihin. Tämän perusteella tutustaan Import-työkalun toimintaan yleisellä tasolla ja kuvaillaan siihen tarvittavat muutokset.

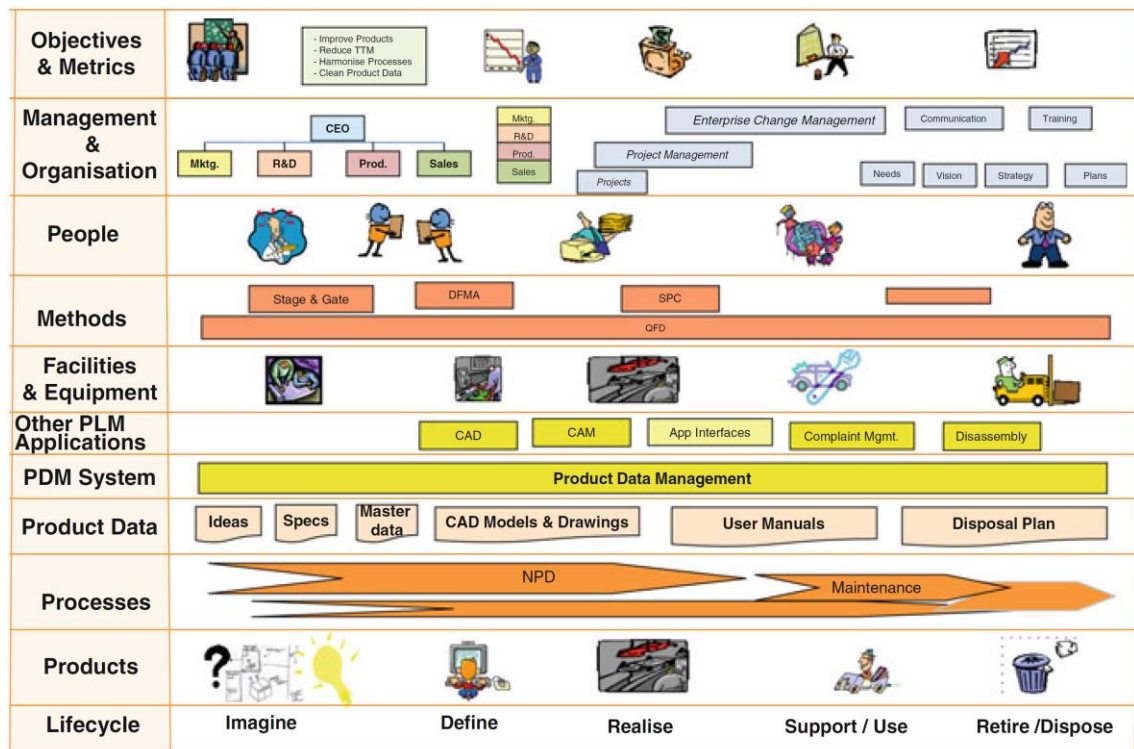
2 TEORIA

2.1 Tuotteen elinkaaren hallinta

Tuotetieto on kaikki tuotteeseen ja sen luomiseksi tarvittaviin prosesseihin, kuten suunnitteluun, valmistukseen, käyttöön ja kierrättämiseen, liittyvä informaatio. Informaatio ei ole selkeästi tietyn tyyppistä vaan se voi: kuvailla tuotteen ominaisuuksia, olla tuotteen komponentti, 3D -malli, ohjelmisto, käyttöohje, tunniste numero tai jotakin muuta tuotteen elinkaareen liittyvää tietoa. (Stark 2015, 134)

Historiallisesti tuotteiden tietoa on aina hallittu tavalla taikka toisella. Tuotteen elinkaaren hallintaparadigma on kehittynyt tarpeesta hallita tuotteista syntyvää suunnittelu- ja rakennedatata. Aluksi painopiste oli suunnittelun ja tuotetiedon hallinnassa. Tiedon määrän kasvaessa ja kytkösten syntyessä muihin toiminnan osa-alueille yrityksen sisällä ja sen ulkopuolella, syntyi tarve laaja-alaisemmalle johtamistavalle tuotteen elinkaaren hallintaparadigmalle. (Saaksvuori & Immonen 2008, 1–2)

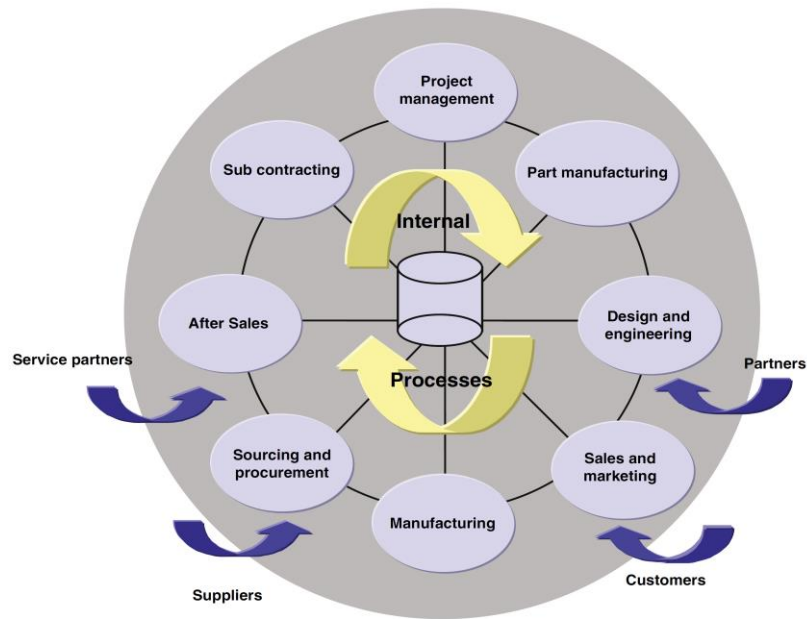
Tuotteen elinkaaren hallinta on yrityksen aktiivista tuotetiedon johtamista kokosen elinkaaren ajan ensimmäisestä tuote ideasta tuotteeksi ja siihen asti, kunnes tuote poistuu käytöstä ja se hävitetään (Stark 2016, 2). Geneeriseen tuotteen elinkaareen määritellään viisi vaihetta: kuvittelu, määrittely, toteutus, tuki/käyttö, käytöstä poisto/hävittäminen. (Stark 2016, 4) Tuotteen elinkaaren hallinta on kokonaisvaltainen lähestymistapa, jossa tuote ja siihen liittyvä data nostetaan yrityksen toiminnan keskiöön. Yrityksen tulisi luoda PLM -visio ja -strategia sekä selkeä toimintapa niiden jatkuvaksi kehittämiseksi. (Stark 2016, 1–15) Toimintaa arvioidessa voidaan hyödyntää työkaluja kuten John Starkin (2015, 9) PLM -matriisia, joka on esitettyinä kuviossa 1. Se pyrkii kuvaamaan laaja-alaisesti tuotteen hallintaan liittyä asioita sen elinkaaren aikana.



KUVIO 1. Starkin (2015, 9) matriisi, jossa esitetään asioita, joita tulisi ottaa huomioon tuotteen elinkaaren hallinnassa.

2.2 Tuotteen elinkaaren hallintaohjelmat

Tuotteen elinkaaren eri vaiheiden tietoa on perinteisesti hallittu erilaisilla ohjelmissa. Tämä johtuu ilmeisesti siitä, että ohjelmia on alun perin kehitetty ratkaisemaan yksittäisiä tarpeita tietyssä elinkaaren vaiheessa olevaan ongelmaan tai tarpeeseen. Tarve ja teknologia näiden yhdistämiselle on tullut vasta myöhemmin. Tuotteen tietoa on perinteisesti ollut ainakin: myynnin (CRM), oston (CSM), suunnittelun (CADit, PDM, Office), laadun (PDM, Office), hallinnon (ERP, Office), tuotannon (MES, MOM, APS, Office), alihankinnan (SCM, Office) ja palveluiden (MRO, Office) järjestelmissä. Tuotteen elinkaaren hallintaohjelmistojen tuleekin pystyä palvelemaan hyvin erilaisia sidosryhmiä kuten kuvioista 2 voidaan havaita.



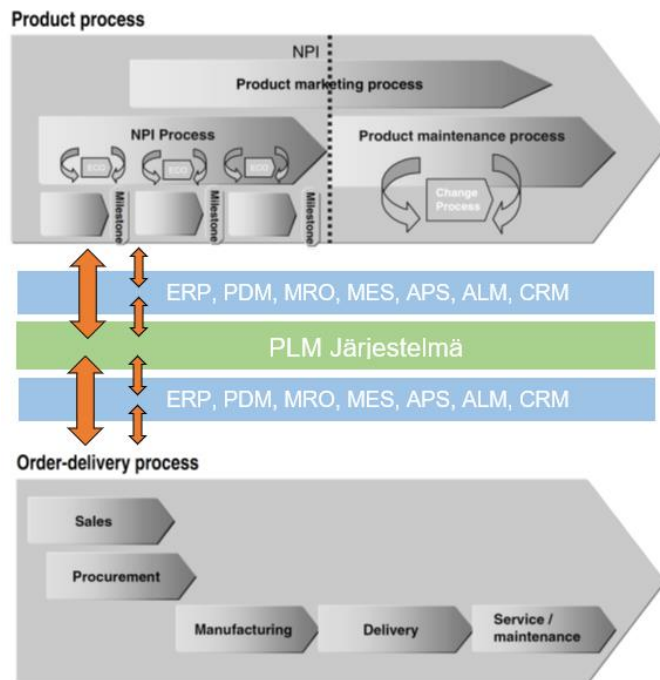
KUVIO 2. Tuotteen elinkaaren hallinta järjestelmien tulee pystyä palvelemaan hyvin erilaisia tarpeita ja sidosryhmiä. (Saaksvuori & Immonen 2008)

Saaksvuoren ja Immosen mukaan ideaali tapauksessa tuotteen elinkaaren hallintajärjestelmä on tiedon prosessointijärjestelmä tai kokoelma IT -järjestelmiä, jotka yhdistävät yrityksen kaikki toiminnot kokonaisuudeksi. Integrointi tapahtuu yhdistämällä sekä kontrolloimalla yrityksen prosesseja ja tuotteita tuotetiedon avulla. Järjestelmä ei saisi olla yksinäinen saareke eikä keskittyä vain esim. suunnittelun tarpeisiin vaan sen tulisi yhdistää ja palvella koko organisaatiota ja sen sidosryhmiä. (Sääksjärvi 2008, 13–14)

PLM -järjestelmät, kuten Siemensin Teamcenter, tarjoavat perinteisen tuotetiedon (CAD, nimike, rakenne) lisäksi omia komponenttejaan tuotteen elinkaaren eri vaiheiden hallintaan. Elinkaaren hallinnan ohjelmistoissa voi olla ainakin seuraavia toiminnallisuuksia: nimikkeen-, rakenteiden-, julkaisun-, muutoksen-, portfolion-, projektin- ja ohjelmien-, laadun-, huollon- sekä vaatimustenhallinta. PLM -järjestelmät myös kuvaavat, miten näiden komponenttien tiedot liittyvät toisiinsa ja tarjoavat tiedon saataville kaikille sitä tarvitseville.

PLM -järjestelmät ovat myös integroitavissa muihin järjestelmiin, koska niiden avulla tulee voida hallinta tietoa myös muista lähteistä. Muiden toimijoiden ratkaisuista saadaan siis data virtaamaan kaksisuuntaisesti PLM -järjestelmän välillä. Tiedon virtaaminen elinkaaren aikana on esitetty kuviossa 3. Tietoa kertyy eri

järjestelmiin elinkaaren aikana ja riippuen siitä, miten PLM -järjestelmät on rakennettu, voidaan tätä tietoa hyödyntää eri tavalla elinkaaren aikana.



KUVIO 3. Tuote data voi liikkua kaksisuuntaisesti eri järjestelmien välillä. (muokailen Saaksvuori & Immonen 2008, 4)

2.3 Nimiketieto

Nimike (eng. item) on systemaattinen ja standardoitu tapa tunnistaa, koodata ja nimetä tuote, tuotteen elementti tai moduuli, komponentti, materiaali tai palvelu. Se, mikä määrittellään nimikkeeksi, riippuu yrityksen omista tarpeista ja tuotteista. Nimikkeitä voivat olla myös esimerkiksi ohjeet, NC -ohjelmat, työkalut, alihankkijat tai kiinnikkeet. Nimikeluokat, hierarkia ja niiden väliset relaatiot tulee tarkkaan suunnitella ja dokumentoida. Suunnitelmaa, joka sisältää mm. nämä nimikkeiden suhteet kutsutaan tietomalliksi ja sitä tulisi noudattaa koko organisaation tasolla. (Saaksvuori & Immonen 2008, 11–13)

Nimikkeeseen liittyvää tietoa voi olla tuotetietoa tai metadataa eli tietoa tiedosta. Tuotetieto on yleensä jokin tuotos, kuten CAD -malli tai Word -tiedoston sisältämä informaatio. Metadata on tuotetiedon elinkaaren kuvaamiseen tarvittavaa tietoa. Nimikkeeseen liittyy aina yksilöivä tunnus, kuvaava nimi sekä lyhyt kuvaus nimikkeestä. Nimikkeeseen liittyy myös paljon muita perustietoja, joista osa vaihtelee

yriyöksittäin. (Stark 2016, 171) Kuvat 1 ja 2 ovat kuvakaappauksia Teamcenteristä. Kuvissa näkyy metadataa, kuten ID ja julkaisutila. Näkyvissä on myös erilisiä tuotetietoja, kuten pdf- ja NX 3D -tiedostonimikkeet (eng. dataset), jotka on kytketty omilla relaatioillaan jigsaw-nimikkeeseen.

The screenshot displays the Siemens Teamcenter Active Workspaces interface for a 3D model named '004688/E;1-JigSaw'. The interface is divided into several sections:

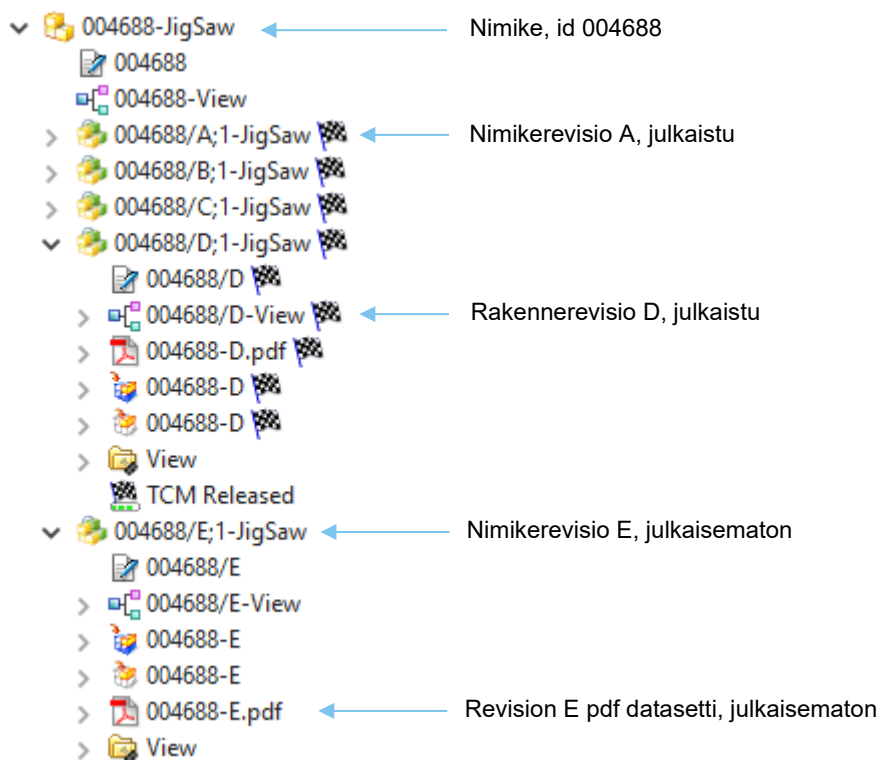
- Left Panel:** A navigation sidebar with icons for Home, Folders, Active Folders, Favorites, Inbox, Changes, Schedules, Schedule Tasks, Reports, Alerts, Help, and No Active.
- Tree View:** A hierarchical tree showing the structure of the model. The root is '004688/E;1-JigSaw' (ID: 004688). Below it are various components:

Element	ID	Seq
004688/E;1-JigSaw	004688	
004679/B;1-trigger	004679	10
004678/B;1-vis 6x7	004678	20
004681/B;1-baseplate	004681	30
004684/A;1-specialnut	004684	40
004672/A;1-battery compartment	004672	50
004693/A;1-jigsawbody	004693	60
004686/A;1-front_jig saw body	004686	10
004690/A;1-back_jig saw body	004690	20
004671/A;1-led-receptacle	004671	70
004670/D;1-motor	004670	80
004717/A;1-Blade x 2	004717	10
004673/A;1-frontfan x 2	004673	20
036325/A;1-Connector2A x 2	036325	30
004675/B;1-vis 3x7 x 6	004675	40
004691/A;1-verticalbearing x 2	004691	50
004665/A;1-armature x 2	004665	60
- Properties Panel:** Displays metadata for the selected element:
 - ID: 004688
 - Revision: E €
 - Revision Name: JigSaw
 - Description: 004688
 - Occurrence Name:
 - Reference Designator:
 - Sequence:
 - Quantity:
 - Unit Of Measure:
 - Release Status: Released
 - Date Released: 06-Apr-2020 15:31
 - Release Effectivity: Released 23-Apr-2020 00:00 to UP (NONE)
 - Element Effectivity ID:
 - Element Effectivities:
 - Owner: Ed The Engineer (ed)
 - Group ID: Engineering
 - Last Modifying User: Ed The Engineer (ed)
 - Practica: False
- Preview Panel:** Shows a 3D model of the Jigsaw and a list of associated datasets:
 - 004688-E UGMASTER
 - Show Markups
 - Markup Panel
 - Full
 - Screen
- Right Panel:** A vertical toolbar with icons for Information, Discuss, Open, Cut, Copy, Paste, New, Edit, Manage, Share, and View.

KUVA 1. Jigsaw nimike Teamcenterin Active Workspacesissa. Vasemmalla nimikkeeseen rakenne näkymä ja oikealla nimikkeeseen metadata tietoja kuten julkaisutila ja omistaja.

Nimikkeistä voidaan muodostaa tuoterakenteita. Tuoterakenteissa nimikkeet, olivat ne sitten osia, kokoonpanoja, dokumentteja tai palveluita, kytketään hierarkiseksi rakenteeksi, joka kuvaa tuotteen rakenteen. Tuoterakenne ja sen hallinta muodostavat perustan useille PLM -järjestelmien toiminnoille. (Saaksvuori & Immonen 2008, 45) Tuoterakenne kuvaa tuoterakenteen jostakin tietystä näkökulmasta. Yleisin on tuotannon näkökulmasta määritetty osaluettelo (eng. Bill Of Materials), joka kertoo mistä komponenteista tuote koostuu. Se voi olla monitasoinen kuvaten, mistä alikokoonpanoista ja komponenteista tuote koostuu. Samasta tuotteesta voidaan muodostaa myös muita rakenteita, koska eri toimintojen perspektiivistä osaluettelo ei ole mielekäs. Sama rakenne voi olla hyvinkin erilainen oston, myynnin, suunnittelun tai palveluiden näkökulmasta. PLM järjestelmät tukevat tällaista nk. monirakenne lähestymistapaa.

Kun nimikkeen oleelliset tiedot muuttuvat, muodostetaan siitä uusi revisio. Näiden revisioiden käyttöä seurataan julkaisutilan avulla, johon voidaan myös liittää revision voimassaoloaika tai -yksiköt. Rakenteilla nimikkeestä esiintyy aina jokin tietty revisio. Revisioiden määrittäminen voidaan tehdä kiinteästi, jolloin jokin revisio lukitaan käytettäväksi rakenteessa, tai ne voidaan määrittää dynaamisesti revisiosäännön perusteella. Revisiosäännöt käyttävät esimerkiksi julkaisutilaa ja nimikkeen päivämääriä määrittääkseen mikä revisio rakenteelle näytetään. Revisiosääntöjä ovat esimerkiksi ”Uusin luontipäivän mukaan” tai ”Julkaistu julkaisutilan mukaan”, jolloin nähdään sillä hetkellä käytössä oleva rakenne. Kuvassa 2 on näkyvissä Jigsaw-nimike, jolla on useampi TCM Released-julkaisutilassa oleva revisio, joita revisiosääntö rakenteessa tulkitsee. (Teamcenter Basics 2021 4.36–4.48, Structure Manager 2021, 11.1–11.10)

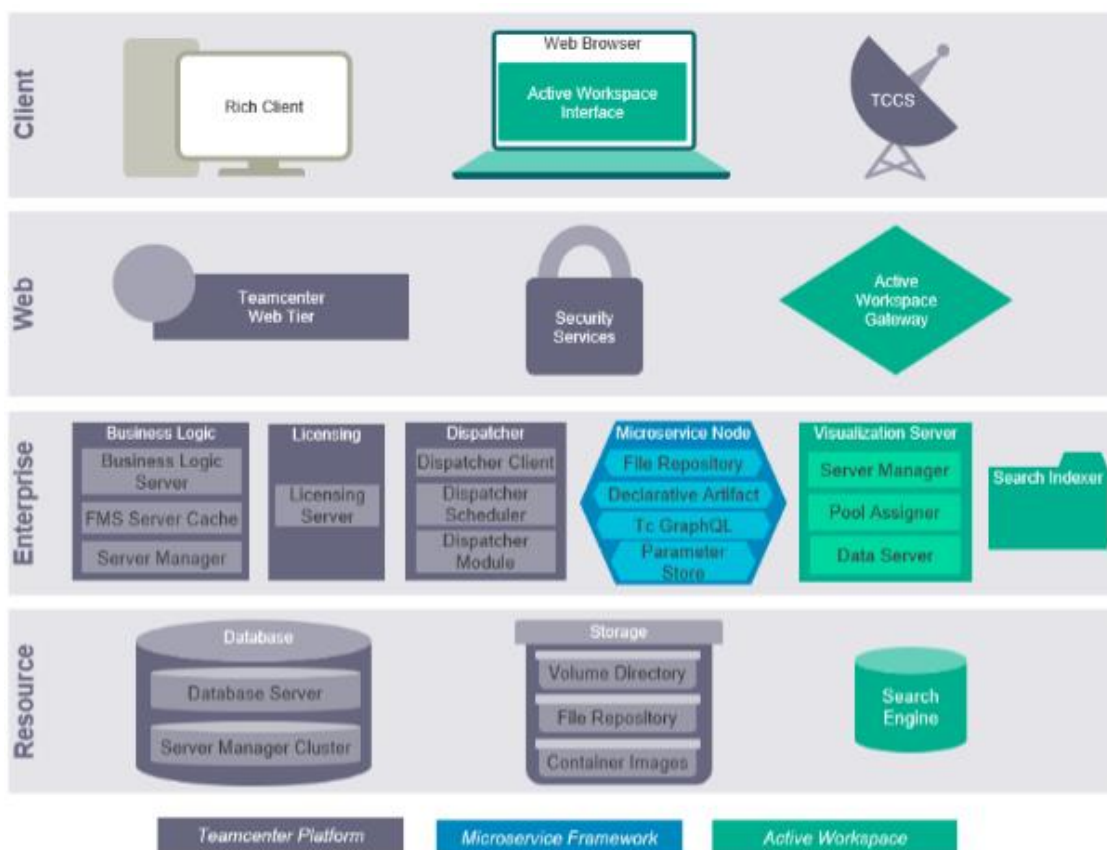


KUVA 2. Jigsaw nimike Teamcenterin rich clientissa. Revisiot liittyvät nimikkeen päätasoon relaatiolla. Vastaavasti PDF- ja CAD -tiedostot liittyvät revisioihin.

3 TEKNISET RATKAISUT

3.1 Teamcenterin rakenne

Teamcenter PLM -järjestelmän rakenne kuvataan yleensä koostuvan seuraavista osista: asiakas-, web-, yritys- ja resurssikerroksesta. Tätä rakennetta kutsutaan nelikerrosarkkitehtuuriksi. Rakenne on esitettyä kuviossa 4. (Teamcenter Server Installation 2021, 3.1–3.3)



KUVIO 4. Teamcenterin (13.0) rakenne. (Teamcenter Server Installation 2021, 3.2)

Asiakaskerroksella sijaitsevat itse ohjelmat, joilla Teamcenteriin yhdistetään. Tähän voidaan käyttää joko selaimessa toimivaa modernia Active Workspace- tai asennettavaa Rich Client-ohjelmaa. Lisäksi asiakaskerroksella sijaitsee jokaisen käyttäjän oma paikallinen välimuisti. Asiakaskerros yhdistyy Teamcenterin palveluihin web-kerroksen kautta. (Teamcenter Server Installation 2021, 3.1–3.3)

Web-kerroksella sijaitsevat palvelimet ohjaavat kaikkea Teamcenterin asiakas- ja yrityskerroksen välistä liikennettä. Kerroksen toteuttamiseen Teamcenterissä käytetään Java EE -ohjelmointikehystä, jota voidaan käyttää yhteensopivilla palvelimilla esim. Oracle WebLogic. Myös .NET web-kerroksen käyttö on mahdollista. (Teamcenter Server Installation 2021, 3.1–3.3)

Yrityskerroksella sijaitsevat varsinaiset Teamcenterin toimintalogiikan sisältävät palvelimet ja muut komponentit. Vain tämä kerros kommunikoi suoraan resurssikerroksen kanssa. (Teamcenter Server Installation 2021, 3.1–3.3)

Tietokanta ja loogiset taltioid sijaitsevat resurssikerroksella. Tietokanta sisältää kaiken tiedon Teamcenterissä olevista nimikkeistä, kuten metadata, sekä muuta ohjelman toimintaan liittyvää tietoa. Tiedostot on tallennettu tiedostopalvelimelle, mutta niihin liittyvä metadata on tietokannassa. (Teamcenter Server Installation 2021, 3.1–3.3)

Dispatcher on Teamcenterin palvelu, joka voidaan laskea kuuluvaksi nelikerrosarkkitehtuurin yrityskerrokselle. Dispatcher on erillinen palvelu, jota käytetään yleensä tekemään erilaisia tiedostokäännöksiä (CAD) tai muita toimenpiteitä, joiden tekeminen voi kestää kauan tai vaatii erillisiä resursseja. Se on kuorman tunteva (muisti, cpu), modulaarinen ja skaalautuva jonotusjärjestelmä pyyntöjen käsittelemiseen. Sen voidaan katsoa toimivan asiakasohjelmien kanssa samalla kerroksella ja kommunikoivan Teamcenterin palvelimien kanssa samalla tavalla. Dispatcher koostuu asiakas-, aikatauluttaja- ja moduuli komponenteista. Asiakas komponentti käsittelee aikatauluttajalle tulleita käännöspyyntöjä kutsumalla moduulissa määritettyjä käännösohjelmia. (Installing and Configuring Dispatcher 2021, 1.1–1.4, 7.1)

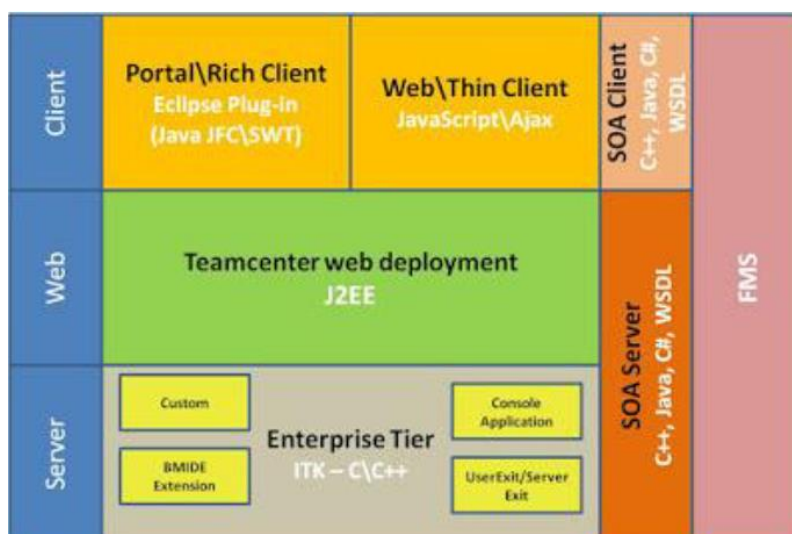
3.2 Teamcenterin kustomointi

Teamcenter tarjoaa käyttäjilleen kattavan toiminnallisuuden listan ja se soveltuu hyvin käytettäväksi sellaisenaankin moneen käyttötapaukseen. Yritysten toimintatavat ja tarpeet poikkeavat toisistaan kuitenkin paljon ja tämän vuoksi sen rakenne on kehitetty joustavaksi. Teamcenter tarjoaakin useita eri tapoja, joilla sen

käyttäytymistä voidaan konfiguroida, muokata tai laajentaa koodaamalla tarvittavia ratkaisuja Siemensin tarjoamien APIen avulla. Kustomointi ja APIen hyödyntäminen vaatii yritykseltä Siemensin Partner-statusta, eivätkä rajapinnat tai materiaalit näihin liittyen ole yleisesti saatavilla.

Teamcenterin toiminnallisuuksia on mahdollista muokata usealla eri tavalla. Muutoksilla voidaan muokata käyttöliittymää sekä laajentaa ohjelman toiminnallisuuksia tai sovittaa niitä paremmin yrityksen prosesseihin. Käyttöliittymämuokkauksia voidaan tehdä sekä rich client- että Active Workspace-käyttöliittymiin. Näiden molempien asiakasohjelmien ulkoasua ja toimintaa voidaan muokata mm. tyylisivujen avulla. Lisäksi rich clienttia voidaan laajentaa Javalla ja AW:tä kustomoidulla Node.js:llä. (Teamcenter Client Customization 2021, 1.1–1.3, 4.1, AW Configuration and Extensibility 2021, 6.1–6.10)

Toimintalogiikkaan vaikuttavien muutosten tekemiseen järjestelmä tarjoaa monia eri tapoja kuten kuvioista 5 voidaan havaita. Nämä muutokset tehdään pääasiassa yrityskerroksella. Tietomallia muuttamalla voidaan muokata mm. Teamcenterin käytössä olevia nimiketyyppejä (objekteja), niillä olevia attribuutteja (~metadataa) sekä vaikuttaa erilaisilla säännöillä nimiketyyppien perustoiminnallisuuksiin, kuten revisiointi- tai tallennuskäyttäytymiseen. Järjestelmään on myös sisäänrakennettu kustomointipisteitä, joihin voidaan kytkeä ohjelmakoodia haluttujen toimintojen aikaan saamiseksi. (Teamcenter Customization 2021, 2.1–2.3, 2.21–2.22)



KUVIO 5. Teamcenterin muokkaukseen käytettäviä tekniikoita ja muokkauspiisteitä. (Teamcenter Infrastructure Basics 2021)

Teamcenter tarjoaa palvelurajapinnan, jota voidaan kuluttaa muiden ohjelmien toimesta. Tämän lisäksi se tarjoaa myös mahdollisuuden lisätä omia palveluja arkkitehtuuriin. Opinnäytetyössä web-aplikaatio kommunikoi tämän rajapinnan avulla Teamcenterin kanssa. (Teamcenter Customization 2021, 3.1–3.3, Teamcenter Services 2021)

Teamcenterin alimman tason toimintojen ohjelmointi tapahtuu Integration Toolkit -rajapinnan avulla. Sen avulla voidaan ohjelmoida itsenäisiä palvelimella ajettavia ohjelmia tai Teamcenterin ajon aikana kutsumia toiminnallisuuksia. Opinnäytetyössä ITK -rajapinnalla toteutettiin kaksi komentokehotetyökalua, jotka toteuttivat nimikkeiden julkaisutilojen käsittelyn ja tilannevedosten luomisen päänimikkeelle. (Teamcenter Customization 2021, 4.1–4.3)

Teamcenterin dispatcher palvelua ei ole esitettyä kuvassa 5, mutta se tarjoaa toiminnallisen muokkauspisteen järjestelmään. Dispatcherille voidaan tehdä uusia kääntäjiä ja niiden käyttäytymistä voidaan muuttaa. Opinnäytetyössä luodut kaksi komentokehotetyökalua lisättiin dispatcherin kääntäjän ajamaan skriptiin. (Installing and Configuring Dispatcher, 1–2)

3.3 PLM XML

PLM XML on Siemensin luoma xml -skeema, jolla voidaan esittää tuotteen elinkaaren hallinnassa tarvittavaa tietoa. Sillä voidaan esittää muun muassa nimikkeitä, niiden tuotetietoa ja metadataa sekä rakenteita. Kuvassa 3 on esitetty katkelma yksinkertaisen rakenteen PLM XML exportista. PLM XML -skeema on luotu erilaisten järjestelmien yhteensovittamiseksi ja se on vapaasti saatavilla. PLM XML noudattaa W3C XML -skeema suosituksia. Siemensin PLM XML SDK tarjoaa kehittäjille C++ API:n, jonka avulla sovelluskehittäjät voivat hyödyntää Siemensin tuotteista tulevaa tietoa sekä esittää skeeman avulla tietoa Siemensin PLM -tuotteiden ymmärtämässä muodossa. Opinnäytetyössä käytettävä PLM XML -data ei noudattanut täysin Siemensin standardia, mikä hankaloitti sen tulkitsemista. (PLM XML 2021, PLM XML Product DS 2021)

```

<ProductRevision id="id36" name="1 LVL ASSY" subType="ItemRevision" accessRefs="#id14" masterRef="#id38" revision="0">
  <ApplicationRef application="Teamcenter" label="kiLAAkzf5ELEDD" version="kiPAAkzf5ELEDD"></ApplicationRef>
  <UserData id="id37">
    <UserValue title="last_mod_date" value="2021-11-24T15:34:34Z"></UserValue>
    <UserValue title="object_string" value="MM1005153/0-1 LVL ASSY"></UserValue>
  </UserData>
</ProductRevision>
<ProductRevision id="id87" name="TOP LEVEL" subType="ItemRevision" accessRefs="#id14" masterRef="#id89" revision="0">
  <ApplicationRef application="Teamcenter" label="C3MAAkzf5ELEDD" version="C7BAAkzf5ELEDD"></ApplicationRef>
  <UserData id="id88">
    <UserValue title="last_mod_date" value="2021-11-24T15:24:12Z"></UserValue>
    <UserValue title="object_string" value="MM1005152/0-TOP LEVEL"></UserValue>
  </UserData>
</ProductRevision>
<RevisionRule id="id2" name="Latest by Creation Date">
  <ApplicationRef application="Teamcenter" label="QsKAAAA45ELEDD" version="QsKAAAA45ELEDD"></ApplicationRef>
</RevisionRule>
<ProductView id="id4" ruleRefs="#id2" primaryOccurrenceRef="id6" rootRefs="id6">
  <ApplicationRef application="Teamcenter" label="iPMAAkzf5ELEDD/QsKAAAA45ELEDD/AAAAAAAAAAAAA/BOM/"></ApplicationRef>
  <UserData id="id3" type="TC Specific Properties">
    <UserValue title="BOM_precision_type" value="imprecise"></UserValue>
  </UserData>
  <Occurrence id="id6" instancedRef="#id87" occurrenceRefs="id9 id43 id79">
    <ApplicationRef application="Teamcenter" label="iPMAAkzf5ELEDD/"></ApplicationRef>
    <UserData id="id7" type="AttributesInContext">
      <UserValue title="AO_ID" value=""></UserValue>
      <UserValue title="SequenceNumber" value=""></UserValue>
      <UserValue title="OccurrenceName" value=""></UserValue>
      <UserValue title="Quantity" value=""></UserValue>
    </UserData>
    <Transform id="id5">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0</Transform>
  </Occurrence>
  <Occurrence id="id9" instancedRef="#id36" instanceRef="#id6">
    <ApplicationRef application="Teamcenter" label="iPMAAkzf5ELEDD/"></ApplicationRef>
    <UserData id="id10" type="AttributesInContext">
      <UserValue title="AO_ID" value=""></UserValue>
      <UserValue title="SequenceNumber" value="1"></UserValue>
      <UserValue title="OccurrenceName" value=""></UserValue>
      <UserValue title="Quantity" value=""></UserValue>
    </UserData>
    <Transform id="id8">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0</Transform>
  </Occurrence>

```

bl_indented_title	b...	bl_rev_release_statu...
MM1005152/0-TOP LEVEL	0	
MM1005153/0-1 LVL ASSY	0	Released - Y0
MM1005156/0-COMPO 2	0	Released - Y0
MM1005157/0-COMPO 3	0	Released - Y0
MM1005154/1-1 LVL ASSY 2	1	
MM1005160/0-ASSY 3	0	Approved
MM1005158/0-COMPO 4	0	Approved
MM1005159/0-COMPO 5	0	Approved
MM1005155/0-1 LVL COMPO	0	Released in TC

KUVA 3. Katkelma testinimikkeen yksinkertaisesta PLM XML exportista, jossa voidaan nähdä rakenteiden linkitys sekä nimikkeen muutamia metadata kenttiä.

4 TOTEUTUS

4.1 Lähdekoodiin perehtyminen

Haluttujen toiminnallisuuksien toteuttamiseksi tarvittavien muutosten selvittäminen aloitettiin perehtymällä nykyisen työkalun toteutukseen. Ratkaisun ensimmäinen versio oli tehty yli 5 vuotta sitten ja siihen oli ajan mittaan tehty muutoksia käyttäjiltä tulleen palautteen perusteella. Dokumentaation ajantasaisuudesta ei voitu olla varmoja, joten koodi oli käytävä lävitse ja dokumentaatiota päivitettävä tarvittavilta osin. Ratkaisun toteuttajat olivat kuitenkin edelleen yrityksen palveluksessa, joten he pystyivät antamaan perehdytyksen työkalun perustoiminnasta. He pystyivät myös auttamaan ohjelman toimintalogiikan ja komponenttien toiminnan selvittämisessä.

Ratkaisuun perehtymisen ensimmäinen vaihe oli kaikkien uusimpien lähdekoodien etsiminen ja niiden läpikäyminen. Yrityksessä käytetään mm. SVN -versiohallintaohjelmaa, jossa kaikkien uusimpien lähdekoodien tulisi olla. Tästä huolimatta lähdekoodiin tutustuessa tuli vastaan epäjohdonmukaisuuksia ja uusimpia versioita jouduttiin etsimään myös muista lähteistä. Import-työkalussa oli käytetty useampaa eri ohjelmointikieltä, mikä hidasti hieman työn alkuun pääsemistä.

4.2 Lähtötilanne

Import-työkalu tarjoaa käyttäjille web-portaalin, jonka avulla 3D CAD -mallit ja rakenteet metadatoineen voidaan siirtää Teamcenteriin. Portaalin kautta ladattava tuotepaketti tulee olla pakattu zip -formaattiin ja sen tulee sisältää tuotteen 3D CAD -tiedostoja JT ja NX yhteensopivissa formaateissa sekä tuotteiden muut tiedot PLM XML -formaatissa. Web-portaalin kautta käyttäjä kirjautuu sisään Teamcenteriin ja kirjautumisen yhteydessä järjestelmästä noudetaan käyttäjätietojen lisäksi projektit, joihin käyttäjä kuuluu. Käyttäjän tulee valita projekti ennen paketin lataamista järjestelmään, sillä järjestelmään vietävä tuote laitetaan tämän projektin omistukseen. Projektit ovat osa Teamcenterin pääsynhallintaa, joka rajoittaa tuotteeseen pääsyä käyttäjiltä.

Web-portaali lähettää ladatun paketin palvelimelle, jossa siitä generoidaan pyyntö dispatcherille. Dispatcher käynnistää käänösprosessin, jonka aikana PLM XML - ja 3D -tiedostojen sisältämä tieto muutetaan NX -työkalun ja parserin avulla Teamcenterille kelpaaviksi siirtotiedostoiksi.

Lopputuloksena Teamcenterissä on koko tuotteen rakenne 3D -malleineen, täydennettynä jo Teamcenteristä löytyneillä osilla. Kaikille tuoduille tuotteille on annettu Approved-julkaisutila, joka estää niiden muokkaamisen järjestelmässä. Lisäksi päätason nimike on kiinnitetty käyttäjän valitsemaan projektikansioon.

4.3 Testijärjestelmän pystyttäminen

Vanhan ja uuden ratkaisun toiminnan testaamista ei pystytty tekemään ilman toimivaa Teamcenter-järjestelmää. Tähän tarkoitukseen muokattiin olemassa oleva virtuaaliympäristö vastaamaan asiakkaan käyttöympäristöä tarvittavilta osin. Virtuaaliympäristöön asennettiin tarvittavia komponentteja asiakkaan tietomallista sekä käännettiin uusimmat versiot lähdekoodeista. Ympäristön Tomcat-palvelimelle asennettiin työkalun web-aplikaatio sekä dispatcherille konfiguroitiin tarvittava käänöskomponentti, jolle web-aplikaatio lähettää purettavan zip -paketin. Virtuaaliympäristössä oli käytössä Teamcenterin versio 12.3, kun taas asiakkaalla käytössä oli versio 11.2. Nämä käyttivät eri Java-versiota, mutta työkalun toiminnan kannalta tällä ei ollut merkitystä, kunhan käytettiin vanhemmasta versiosta löytyviä API kutsuja. Ympäristöön oli myös asennettuna uudempi versio NX -mallinusuohjelmasta, mutta työkalun hyödyntämässä Import-toiminnossa ei ollut eroavaisuuksia tuotannossa olevaan. Tässä mielessä versiot olivat yhteneväiset.

Virtuaaliympäristön rakentaminen on PLM -järjestelmän kehittäjälle erittäin hyödyllinen, sillä se helpottaa monikomponenttisen järjestelmän toiminnan hahmotamista, testaamista sekä uuden ratkaisun käyttöönottoa. Virtuaaliympäristöllä testatessa ei tarvita mm. pääsyä ja oikeuksia asiakkaan ympäristöihin. Muokattulla virtuaaliympäristöllä olemassa oleva ratkaisu saatiin rakennettua toimivaksi

kokonaisuudeksi ja sen toiminta varmistettua ennen asiakasympäristöön siirtymistä.

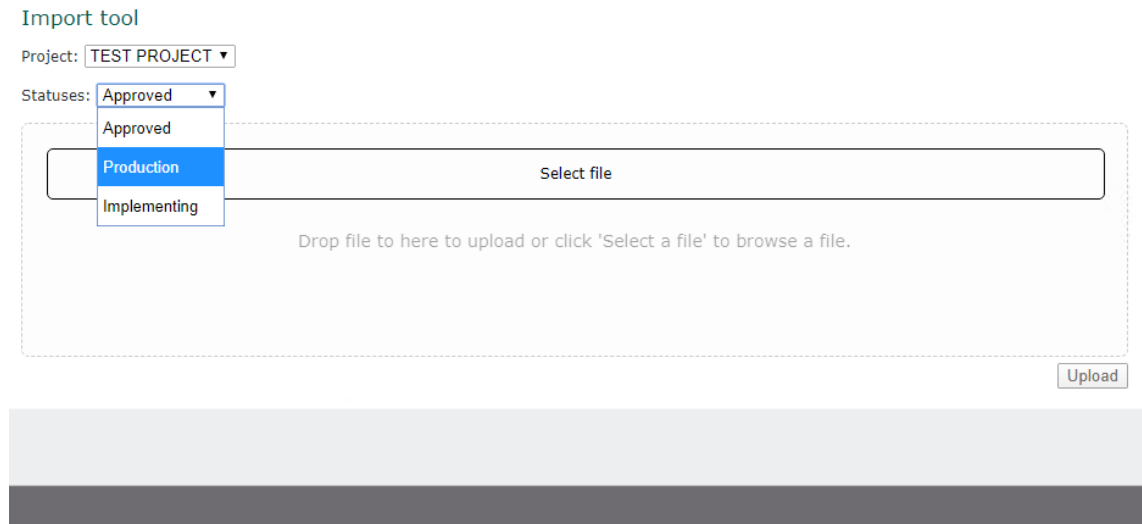
Hieman haasteelliseksi testaamisen kannalta osoittautui lähtödatan laatu ja saatavuus. Työkalulle lähetettävät mallit ja PLM XML -tieto tulivat autoteollisuuden toimijan itse kehittämästään tuotetiedonhallintajärjestelmästä, mikä ei täysin vastannut PLM XML -standardia. Käytössä oli kaksi eri kokoista testipakettia, joista suurempi yli 4 Gb -paketti aiheutti virtuaalikoneen resurssien loppumisen eikä sitä voitu testaamisessa käyttää. Samasta paketista oli myös pilkottu versio, josta osa datasta oli karsittu pois. Tämä aiheutti ongelmia NX:n työkalulle. PLM XML:sta olisi mahdollisesti voitu eriyttää jokin alikokoonpano testaamista varten, mutta se olisi ollut operaationa niin työläs, ettei siihen ryhdytty. Testaamista varten olisikin tarvittu riittävän pieni paketti, jonka sisältö olisi tunnettu tarkkaan. Rakennetulla testiympäristöllä pystyttiin kuitenkin testaamaan uuden ratkaisun kriittisiä kohtia, vaikka end-to-end -prosessia ei täysin onnistuneesti voitukaan tehdä.

4.4 Työkalun uusi käyttäytyminen

Työkalua käytettäessä oli huomattu tarvittavan usein vaihtoehtoa säätää tuotteille asetettua julkaisutilaa. Ennen opinnäytetyössä tehtyjä muutoksia tuotteille asetettiin aina vain Approved-julkaisutila. Tuotteille tuli voida asettaa ensisijaiseksi julkaisutilaksi joko Approved, Implementing tai Production. Julkaisutiloja ei kuitenkaan voitu asettaa samalla tavalla, kuin aikaisemmin asetettiin Approved-tila. Työkalun tuli pystyä tarkistamaan nimikkeen julkaisutila Teamcenteristä ja päättämään muutetaanko tila vai lisätäänkö sille vain uusi julkaisutila olemassa julkaisutilan lisäksi. Jos nimikkeellä oli vain Approved-tila järjestelmässä, tuli sille lisätä ainoastaan käyttäjän asettama julkaisutila. Jos asetettava julkaisutila oli Production korvattiin sillä olemassa oleva Implementing-tila poistamatta nimikkeellä olevia muita julkaisutiloja. Jos asetettava ja järjestelmässä oleva julkaisutilat olivat samat, mitään toimenpiteitä ei tehty esim. julkaisupäivän muutos julkaisutilalle. Muita kuin mainittuja julkaisutiloja ei käsitelty. Työkalun tuli myös luoda sisään tuotavasta rakenteesta tilannevedos, joka liitettiin tuodun päätason nimikkeen revisiolle. Tilannevedos on Teamcenterin toiminnallisuus, joka tallentaa yksinkertaisen kuvauksen rakenteesta nimike tietoineen ilman 3D -dataa.

4.5 Web-portaalin muutokset

Uuden käyttäytymisen mahdollistamiseksi web-käyttöliittymään lisättiin uusi alavetovalikko, johon halutut statukset lisättiin. Web-portaali lisätyn julkaisutilavaliikon kanssa on esitetty kuvassa 4.



KUVA 4. Web-portaali lisättyllä julkaisutila valikolla.

Tieto valitusta julkaisutilasta välitettiin back-endille SubmitEventistä lisäämällä se uploader JavaScript-objektin parametriksi. Tämä on esitetty kuvassa 5. Uploader käsittelee tiedoston lataamisen osissa palvelimelle ja välittää tiedon latauksesta ja julkaisutilasta back-endin ImportServlet luokalle.

```

* Submits files user has added to the server.
*/
function submitFiles(event) {
    if (validateImportParameters()) {
        uploader.setParams(
            {
                "status": document.getElementById("statuses").value,
                "project": document.getElementById("projects").value
            }
        );
        uploader.uploadStoredFiles();
    }
}

```

KUVA 5. JavaScript funktio, jolla valittu julkaisutila viedään back-endille.

Servletin avulla status välitetään edelleen Teamcenterin dispatcherille, jotta sitä voidaan hyödyntää ajettaessa lopullisen siirtotiedoston muodostavaa koodia. Julkaisutila-tieto lähetetään kutsumalla Teamcenterin palveluapin rajapintoja, joilla

luodaan uusi pyyntö dispatcherille. Tähän pyyntöön voidaan sisällyttää myös julkaisutila parametrinä kuvan 6 osoittamalla tavalla.

```
keyValue = new DispatcherManagement.KeyValueArguments();  
keyValue.key = "status";  
keyValue.value = info.status;  
keyValues.add(keyValue);
```

KUVA 6. Leike back-endin koodista, jossa määritellään dispatcherille tilamuutoksen muuttuja.

4.6 Dispatcher muutokset

Dispatcherin kääntäjälle tarvittiin myös pieni muutos, koska sille tulevaan kutsuun oli lisätty yksi parametri. Tämä kyseinen parametri lisättiin pyynnön esikäsittelijän TaskPrep Java-luokkaan ja päivitetty kirjasto tuli olla dispatcherin asiakasmoduulissa käytössä. Tällöin dispatcherille tuleva julkaisutila kirjoitetaan muiden parametrien mukana tekstitiedostoon, josta se voidaan käänösprosessin aikana lukea.

4.7 Käännösprosessin muutokset

Dispatcher käynnistää erillisen käänösprosessin, jonka aikana nimikkeiden tiedot kerätään Teamcenteriin viemistä varten. Käänösprosessin aikana puretun zip -paketin sisältämä PLM XML -tiedosto käännetään xsl -muunnoksella siirtotiedostoiksi. Syntynyttä siirtotiedostoa rikastetaan generoimalla NX:llä täydentävää nimiketietoa paketissa olleista CAD -malleista. NX tunnistaa nimikkeitä, jotka ovat CAD -rakenteille merkittäviä, mutta eivät esiinny PLM XML -tiedostossa. Tätä tuotosta jouduttiin vielä jatkojalostamaan parserilla. Parserin ajamisen jälkeen siirtotiedostoissa oli kaikki Teamcenteriin siirrettävät nimikkeet.

Aluksi julkaisutilan muutosta suunniteltiin parseriin. Koska sillä ei voitu suoraan kutsua ITK rajapintaa, oli helpompaa luoda komentokehotetyökalu, joka käy tarkistamassa ja asettamassa nimikkeiden julkaisutilat yhden session aikana. Nimikkeiden julkaisutilojen tarkistamiseksi ja asettamiseksi tehtiin uusi Teamcenter

ITK -komentokehotetyökalu. Työkalulle annettiin syötteenä käyttäjän asettama julkaisutila sekä tiedoston polku, josta nimikkeet parsitaan. Ohjelma toteutettiin C++ -ohjelmointikielellä. Kuvassa 7 on esimerkki siitä, kuinka ITK -rajapintaa käytettiin tässä tapauksessa nimikkeen revisioiden hakemiseen. Rajapintaa käytettäessä piti tuntea käytössä olevien objektien käyttäytyminen ohjelmassa, jotta oikeat toimenpiteet osattiin tehdä esimerkiksi julkaisutilojen korvauksissa ja poistoissa. Samalla ohjelmointirajapinnalla toteutettiin myös tilavedosten luominen. Kun julkaisutilojen ja tilavedosten käsittely lisättiin erillisinä komponentteina käännökseen, ne eivät vaikuttaneet käännöksen muuhun toimintaan. Näin niitä voitiin testata erillään.

```
// gets all ITEM subtypes as well.
fail = ERROR_CHECK(ITEM_find(itemID.c_str(), &n_items, &items_t));

// Must match to 1 item or item is not unambiguous
if (n_items == 1) {

    // get revision list
    ifail = ERROR_CHECK(AOM_ask_value_tags(*items_t,
                                          "revision_list",
                                          &rev_count,
                                          &item_revs_t));

    // iterating all revisions and printing object names
    for (int i = 0; i < rev_count; i++) {

        Teamcenter::scoped_smptr<char> revID_tmp;
        ifail = ERROR_CHECK(AOM_ask_value_string(item_revs_t[i],
                                                "object_string",
                                                &desc_val));

        std::cout << desc_val << std::endl;
    }
}
```

KUVA 7. Esimerkki ITK -koodista, joka hakee nimikkeen revisiot ja tulostaa jokaiselle objektinimen.

5 POHDINTA

Opinnäytetyössä toteutettiin Import-työkalun toiminnallisuuksien laajentaminen. Import-työkaluun tarvittiin mekanismi Teamcenter PLM -järjestelmään vietävien nimikkeiden julkaisutilan asettamiseksi sekä tuodun rakenteen lisäämiseksi tilavedoksena päätason nimikkeen revisiolle. Lisäksi käytiin lävitse työkalulle tehty dokumentaatio sekä varmistettiin versiohallinnassa olevien komponenttien ajan tasaisuus.

Olemassa olevaan ratkaisuun ja Teamcenterin muokkausrajapintoihin tutustumisen perusteella havaittiin, että muutoksia tarvitaan Web-portaaliin front- ja back-endiin sekä dispatcherin kääntäjään, jotta halutut toiminnallisuudet saataisiin toteutettua. Web-portaalin kohdalla muutokset voitiin tehdä olemassa olevaan ratkaisuun. Dispatcherin laukaisemaan käännösprosessiin tarvittiin kuitenkin ITK -rajapinnalla toteutetut komentokehotetyökalut, joilla julkaisutilojen ja tilavedosten vaatimat toiminnallisuudet toteutettiin. Näihin ratkaisuihin päädyttiin ohjelman olemassa olevan rakenteen sekä projektiin käytettävissä olevan ajan perusteella. Lisäksi tällä tavalla olemassa olevan testatun toiminnallisuuden päälle voitiin nyt lisätä ylimääräinen komponentti, joka ei vaikuttanut käännösprosessin muiden vaiheiden toimintaan. Tällöin voitiin luottaa käännöksen tuottavan vastaavia tuloksia kuin aikaisemminkin ja testattavaa löytyi vain uusista toiminnallisuuksista.

Lopputuloksena saatiin toimiva ratkaisu, jolla voidaan siirtää rakenteita järjestelmään halutulla julkaisutilalla. Järjestelmä luo lisäksi sisäänviennin yhteydessä rakenteesta tilavedoksen päätason nimikkeen revisiolle. Jatkossa nimikkeiden julkaisutilat noudattavat hierarkiaa eikä vanhoja julkaisutiloja automaattisesti ylikirjoiteta. Lisäksi ajan tasalla oleva dokumentaatio ja versiohallinta helpottavat työskentelyä jatkossa. Jatkokehityksenä voitaisiin tutkia, voitaisiinko yhdistää käytössä olevan siirtolistaparserin sekä opinnäytetyönä luodut komentokehotetyökalut yhdistää yhdeksi kokonaisuudeksi.

LÄHTEET

- Stark, John. 2015. Product Lifecycle Management (Volume 1): 21:st Century Paradigm for Product Realisation. 3rd edition. Springer International Publishing.
- Saaksvuori, A. & Immonen, A. 2008. Product Lifecycle Management. 3rd edition. Springer International Publishing.
- Stark, John. 2016. Product Lifecycle Management (Volume 2): The Devil is in the Details. 3rd edition. Springer International Publishing.
- Teamcenter Basics. 2021. Siemens Partner Portal. Luettu: 6.12.2021. https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20210421143201885.my_teamcenter/pdf?dest=M8.newlink.id1197118
- Structure Manager. 2021. Siemens Partner Portal. Luettu: 6.12.2021. https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20210421143201885.pse/html/getting_started_get_getting_started
- Teamcenter Server Installation. 2021. Siemens Partner Portal. Luettu: 6.12.2021. https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20210421143201885.server_win/pdf?dest=M8.newlink.partdbinstall
- Installing and Configuring Dispatcher. 2021. Siemens Partner Portal. Luettu: 6.12.2021. <https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20201019171429688.plm00565/pdf?dest=M8.newlink.xid1674628>
- Teamcenter Client Customization. 2021. Siemens Partner Portal. Luettu: 6.12.2021. https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20201019171429688.programmers_guide/html/id1455272
- AW Configuration and Extensibility. 2021. Siemens Partner Portal. Luettu: 6.12.2021. <https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20200109154055345.Configuration/html/xid1348837>
- Teamcenter Infrastructure Basics 2019. IDEAL GRP. Sisäinen koulutusmateriaali. Luettu 6.12.2021.
- Server Customization. 2021. Siemens Partner Portal. Luettu: 6.12.2021. <https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20201019171429688.itk/pdf?dest=M8.newlink.id724482>
- Teamcenter Services. 2021. Siemens Partner Portal. Luettu: 6.12.2021. https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20201019171429688.plm00076/pdf?dest=M8.newlink.before_you_begin_services

PLM XML. 2021. Siemens web page. Luettu 7.12.2021.
<https://www.plm.automation.siemens.com/global/en/products/plm-components/plm-xml.html>

PLM XML Product DS. Old Siemens Product Datasheet. Luettu 7.12.2021.
https://www.plm.automation.siemens.com/legacy/products/open/plmxml/docs/wp_plm_xml_14.pdf